Cryptanalysis of Selected Block Ciphers

Alkhzaimi, Hoda A.

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

Cryptanalysis of Selected Block Ciphers

Dissertation

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at the

Department of Mathematics and Computer Science-COMPUTE

in

The Technical University of Denmark

by

Hoda A.Alkhzaimi

December 2014

To my Sanctuary in life

Bladi, Baba Zayed and My family

with love

Title of Thesis

Cryptanalysis of Selected Block Ciphers

PhD Project Supervisor

Professor Lars R. Knudsen(DTU Compute, Denmark)

PhD Student

Hoda A.Alkhzaimi

Assessment Committee

Associate Professor Christian Rechberger (DTU Compute, Denmark)

Professor Thomas Johansson (Lund University, Sweden)

Professor Bart Preneel (Katholieke Universiteit Leuven, Belgium)

Abstract

The focus of this dissertation is to present cryptanalytic results on selected block ciphers. Block ciphers are the mathematical structure that will take a plaintext message and convert it into a ciphertext one block at a time using a secret key. They play an essential role in many cryptographic architectures and frameworks. For a long time they were known as the main building block that will provide *confidentiality* in an information system. They would also be able to represent a full spectrum of cryptographic services as many block ciphers can be used to construct stream ciphers, hash functions, pseudorandom number generators, and authenticated encryption designs.

For this reason a multitude of initiatives over the years has been established to provide a secure and sound designs for block ciphers as in the calls for Data Encryption Standard (DES) and Advanced Encryption Standard (AES), lightweight ciphers initiatives, and the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR).

In this thesis, we first present cryptanalytic results on different ciphers. We propose attack named the Invariant Subspace Attack. It is utilized to break the full block cipher PRINTcipher for a significant fraction of its keys. This new attack also gives us new insights into other, more well-established attacks. In addition, we also show that for weak keys, strongly biased linear approximations exists for any number of rounds.

Furthermore, we provide variety of attacks on the family of lightweight block cipher SIMON that was published by the U.S National Security Agency (NSA). The ciphers are developed with optimization towards both hardware and software in mind. While the specification paper discusses design requirements and performance of the presented lightweight ciphers thoroughly, no security assessment is given. We present a series of observations on the presented construction that, in some cases, yield attacks, while in other cases may provide basis of further analysis by the cryptographic community. Specifically, The attacks obtained are using classical- as well as truncated differentials. In addition to that, we also investigate the security of SIMON against different linear

cryptanalysis methods, i.e., classic linear,and linear hull attacks. we present a connection between linear characteristic and differential characteristic, multiple linear and differential and linear hull and differential, and employ it to adapt the current known results on differential cryptanalysis of SIMON to linear cryptanalysis results.

Finally, we investigate links between different methods of cryptanalysis and how they can be utilized for block cipher cryptanalysis. We consider the known results on the links among integral, impossible differential and zero-correlation linear hulls in order to prove that constructing a zero-correlation linear hull always implies the existence of an integral distinguisher. Moreover, we show that constructing zero-correlation linear hull on a Feistel structure with SP-type round functions, where P is a binary matrix, is equivalent to constructing impossible differential on the same structure except that P is substituted by the transposed matrix P^T. We present an integral distinguishers of 5-round Feistel structure with bijective round functions and 3-round Feistel structure with round functions not necessarily being bijective. In addition to an integral distinguishers of Camellia so far, i.e., 7-round integral distinguishers of Camellia with FL/FL^{-1} layer and 8-round integral distinguishers of Camellia without FL/FL^{-1} layer.

Abstrakt (in Danish)

Fokus for denne afhandling er at præsentere kryptoanalytiske resultater på udvalgte blokciphers. Block ciphers er den matematiske struktur, der tager en klartekstmeddelelse og konverterer den til en ciphertext én blok ad gangen ved hjælp af en hemmelig nøgle. De spiller en væsentlig rolle i mange kryptografiske arkitekturer og systemer. I lang tid var de kendt som den vigtigste byggesten, der giver fortrolighed i et informationssystem. De er også i stand til at repræsentere et fuldt spektrum af kryptografiske tjenester da blok ciphers kan bruges til at konstruere stream ciphers, hash funktioner, pseudo tilfældige tal generatorer, message authentication codes (MLA), og autentificerede krypteringsdesign. Af denne grund er der etableret en lang række initiativer i årenes løb at skabe et sikkert og sundt design til block ciphers som i datakrypteringsstandard DES og Advanced Encryption Standard (AES), flere såkaldte letvægtsciphers, og i konkurrencen om "Authenticated Encryption": Security, Applicability, and Robustness (CAESAR). Det første fokus for denne afhandling er at præsentere kryptoanalytiske resultater på forskellige ciphers. Vi foreslår et nyt angreb navngivet "Invariant Subspace Attack". Det anvendes til at bryde blockcipheret "PRINTcipher" for en betydelig del af nøglerummet. Dette nye angreb giver os også ny indsigter i andre, mere veletablerede angreb. Vi udleder en afkortet differential karakteristisk med en runde-uafhængig men yderst nøgle-afhængig sandsynlighed. Derudover viser vi også, at for svage nøgler, eksisterer der stærkt ikke-tilfældige lineære tilnærmelser for et vilkårligt antal runder. I denne forstand opfører PRINTcipher sig meget forskelligt fra hvad der sædvanligvis antages. Derudover tilbyder vi mange forskellige angreb på letvægtsblockcipher-familien SIMON, som blev offentliggjort af det amerikanske National Security Agency (NSA). De ciphers er udviklet med optimering for både hardware og software i tankerne. Medens specifikationen af familien diskuterer krav design og ydeevne af de præsenterede letvægts ciphers grundigt, er der ingen sikkerhedsvurdering angivet. Vi præsenterer en serie af bemærkninger af konstruktionen, der, i nogle tilfælde, muliggør angreb, mens de i andre tilfælde kan tilvejebringe grundlaget for yderligere analyse af det kryptografiske samfund. Specifikt er angrebene opnået ved anvendelse af klassiske- samt trunkerede

differentialer. Hertil kommer, at vi også undersøger sikkerheden af SIMON mod forskellige lineære kryptoanalyse metoder, dvs., klassisk lineær, multi-lineær og lineære "hull" angreb. Vi præsenterer en forbindelse mellem en lineær karakteristik og en differential karakteristisk, multilineære og lineære "hull" angreb, og anvender dette til at tilpasse sig de nuværende kendte resultater på differential kryptoanalyse af SIMON til lineære kryptoanalyse resultater.

Det andet fokus i denne afhandling er at undersøge nærmere forbindelser mellem de forskellige metoder af kryptoanalyse og hvordan de kan udnyttes til blockcipher kryptoanalyse. Vi betragter de kendte resultater på de links mellem integraler, umulige differentialer og nul- korrelation lineære "hull" med henblik på at bevise, at konstruktion af en nul-korrelation for lineære "hull" altid indebærer eksistensen af en "integral distinguisher". Endvidere viser vi, at konstruktion af en nul-korrelation lineær "hull" på en Feistel struktur af SP typen, hvor P er en binær matrix, svarer til at konstruere umuligt differentiale på samme struktur bortset fra, at P er substitueret med den transponerede matrix. Derudover, ved hjælp af de nyligt etablerede forbindelser er de følgende resultater opnået:

- Den første kendte "integral distinguisher" af 5-runders Feistel struktur med bijektive runde funktioner og 3-runders Feistel struktur med runde funktioner, der ikke nødvendigvis er bijektive.

- De bedst kendte "integral distinguishers" af Camellia hidtil, dvs. 7-runders "integral distinguishers" af Camellia med FL/FL-1 lagene og 8-runders "integral distinguishers" af Camellia uden FL/FL-1 lagene.

Preface

Block ciphers are very essential and elemental component in any cryptographic or security structure. The design of such primitive encapsulates making sure that it has certain security level when it operates in different application environment (lightweight, cloud computing, big data structures, etc). Most of the designed block ciphers are meant to take into consideration the implementation environment of the cipher (hardware or software). The cryptographic community in this sense has invented a multitude level of cryptanalytic techniques that targets different possible designs and aim to exploit their weaknesses into successful and potentially practical attacks. The main focus of the research material and results provided in this dissertation is to analyse and evaluate the security of selected block ciphers. It consist of two main parts. The first part is a general introduction to cryptography, structures of cryptographic primitives, and cryptanalysis techniques. The second part is selected publications of block cipher cryptanalysis that were obtained throughout the PhD study period. In particular, the thesis assess and evaluate the security of the lightweight block cipher PRINTcipher, NSA's Family of lightweight block cipher SIMON, and block cipher Camellia.

The outline of this thesis is stated as the following:

- **Chapter 1.** This chapter will present a brief introduce around the formation and evolution of the concepts of cryptography, cipher design, and cryptanalysis inspired by the need of the community all over the years.

- **Chapter 2.** This chapter will briefly introduce the main symmetric-key cryptographic primitives, their cryptographic design strategy, and security requirements. Finally, it gives the description of cryptanalytic attacks, their goals, complexity and models.

- **Chapter 3.** This chapter will outline the main cryptanalytic techniques and methods used for symmetric-key primitives in general and block ciphers in specific. It mainly focus differential and linear cryptanalytic techniques which are used to

provide results in the rest of the chapters.

- **Chapter 4.** This chapter proposes a new attack named the Invariant Subspace Attack. It is utilized to break the full block cipher PRINTcipher for a significant fraction of its keys. This attack can be seen as a weak-key variant of a statistical saturation attack. For such weak keys, a chosen plaintext distinguishing attack can be mounted in unit time. In addition to breaking PRINTcipher, the new attack also gives us new insights into other, more well-established attacks.In addition, we also show that for weak keys, strongly biased linear approximations exists for any number of rounds. In this sense, PRINTcipher behaves very differently to what is usually assumed.

- **Chapter 5.** In this chapter we provide a variety of attacks on the family of lightweight block cipher SIMON that was published by the U.S National Security Agency (NSA). The ciphers are developed with optimization towards both hardware and software in mind. While the specification paper discusses design requirements and performance of the presented lightweight ciphers thoroughly, no security assessment is given. This chapter is a move towards filling that cryptanalysis gap for the SIMON family of ciphers. This chapter present a series of observations on the presented construction that, in some cases, yield attacks, while in other cases may provide basis of further analysis by the cryptographic community. Specifically, The attacks obtained are using classical- as well as truncated differentials. In the former case, this chapter show how the smallest version of SIMON, Simon32/64, exhibits a strong differential effect.

  In addition to that, this chapter also investigate the security of SIMON against different variants of linear cryptanalysis, i.e., classic linear, multiple linear and linear hull attacks. It presents a connection between linear characteristic and differential characteristic, multiple linear and differential and linear hull and differential, and employ it to adapt the current known results on differential cryptanalysis of SIMON to linear cryptanalysis results. Our best linear cryptanalysis results are using average squared correlation of the linear hull of SIMON based on correlation matrices. The results cover 21 rounds of SIMON 32/64 out of 32 rounds with the data complexity $2^{30.56}$ and time complexity $2^{54.56}$. We have implemented our attacks for small scale variants of SIMON and our experiments confirm the theo-

retical biases and correlation presented in this work. So far, The results presented are the best known with respect to linear cryptanalysis for any variant of SIMON.

- **Chapter 6.** In recent years, the discussion to establish links among different cryptanalytic techniques has been actively revisited. In this chapter, the known results on the links among integral, impossible differential and zero-correlation linear hulls presented by Bogdanov *et al.* and Blondeau *et al.* recently are considered. In this chapter, it is proved that constructing a zero-correlation linear hull always implies the existence of an integral distinguisher. Moreover, it shows that constructing zero-correlation linear hull on a Feistel structure with SP-type round functions, where P is a binary matrix, is equivalent to constructing impossible differential on the same structure except that P is substituted by the transposed matrix P^T. Additionally, with the help of the newly established links, the following results are obtained:
  - The first known integral distinguishers of 5-round Feistel structure with bijective round functions and 3-round Feistel structure with round functions not necessarily being bijective.
  - The best known integral distinguishers of Camellia so far, i.e., 7-round integral distinguishers of Camellia with FL/FL^{-1} layer and 8-round integral distinguishers of Camellia without FL/FL^{-1} layer.
- **Chapter 7.** In this chapter, the final brief conclusion and remarks around the different research topics discussed and approached will be presented.

The research conducted in this thesis was performed at the Department of Mathematics and Computer Science in the Technical University of Denmark (DTU-COMPUTE). The author was supervised by Professor Lars Knudsen and partially co-supervised by Associate Professor Gregor Leander in the first year and half of the PhD research project period. The author was funded by a scholarship from Emirates Advanced Investments, UAE.

During the three years of the PhD studies, the following papers were peer reviewed or published.

- Bing Sun, Zhiqiang Liu, Ruilin Li, Lei Cheng, Hoda A. Alkhzaimi, Vincent Rijmen, Chao Li. This paper has been submitted for Eurocrypt 2015 and have been accepted for a second review. The contents of this paper is reflected in Chapter 6. This paper has been peer reviwed.

- Mohamed Ahmed Abdelraheem, Javad Alizadeh, Hoda A. Alkhzaimi, Mohammad Reza Aref This paper is reflected in Chapter 5. It is currently under the process of submission for IEEE Transactions on Information Theory.

- Javad Alizadeh, Hoda A Alkhzaimi, Mohammad Reza Aref, Nasour Bagheri, Praveen Gaurav This Paper is reflected in Chapter 5.

- H. A. Alkhzaimi and M. M. Lauridsen. Cryptanalysis of the simon family of block ciphers. C This paper is reflected in Chapter 5. This paper was peer reviewed as it was submitted to multiple conferences.

- Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda Alkhzaimi, Erik Zenner. A Cryptana This paper reflects the work in Chapter 4.

Acknowledgments

First and foremost I would like to thank ALLAH the most merciful for all his blessings throughout my life, and for always being my strength and peace. To say that the past period that I have spent working on this PhD research project is a journey is a true understatement. I could not have achieved this much without the grace of ALLAH. In the whole process of this PhD research, I realized how truly blessed I am to have the multitude of people that helped me through it all, and the number of experiences that I have been through.

I do not think that I can express the deep and sincere gratitude that I have for my PhD research supervisor Prof.Lars R.Knudsen who has not only accepted me in his research team, but was a friend first throughout this research project. I am thankful for his professional, humble and friendly approach in mentoring his students. I consider myself fortunate to have the opportunity to be within his team. His guidance no matter how naive or annoying my questions are, motivation, encouragement, immense patience and endless support especially during my leave of absence, and trust in me were my backbone through an essential period in my PhD. You helped me form my research experience and obtain a clearer picture of what I know and I learned around my PhD research.

I would like to also thank Dr.Gregor Leander for being my co-supervisor for the first half of my PhD research period. Thank you for being an essential factor in my research with your ideas, advice that helped me identify my research interests.

I have volumes of gratitude for Prof.Tom Høholdt for his initial and crucial guidance around my PhD program not only when he was the head of the PhD school in DTU Department of Mathematics, but for being as much interested in my progress afterwards. Thank you for your discussions around coding theory and cryptography, and being genuinely interested in making me feel included in the group and being at home in Denmark. I am in debt for your constant assurance and care that I am having a good spirit to do this research especially during my leave of absence.

Special thanks to Prof.Carsten Thomassen for his support, encouragement, for the dis-

cussion, and being interested in knowing more about me, my culture and research. Most importantly, for having time to just stop and say ”Hi.how was your day?”.

I consider myself fortunate to be part of DTU Mathematics previously and now DTU-COMPUTE community. I can not think of my journey without having a deep sense of appreciation for my research team in the Cryptology section including members who are currently there and who are now part of another research teams. Thank you Christian, Andrey, Elmar,Praveen, Søren,Julia, Valérie, Christiana Peters, Christina Bora, Martin Albrecht, Arnab, Tyge, Stefan, Martin M. Lauridsen, Ann-Cathrin Dunker, and Subhadeep. Thank you for the kind encouragement, enlightening feedback, lunch-time discussions, and Crypto-Dinners/barbecues. You have all made the Cryptology research team a warm working environment. I would like especially thank my co-authors in the team Mohamed Ahmed Abdelraheem and Martin M. Lauridsen for being always available for questions and support, critically constructive feedback especially and proactive team spirit. Thank you Mohammed for reading parts of this thesis and providing such constructive feedback. I would like to especially thank as well Julia borghoff, Valérie Gauthier Umana and Christiana Peters for being friends, their discussions, feedback and constant assurance throughout the different phases of my PhD.

I would like to thank the extended PhD students and post doctoral researcher in DTU-COMPUTE especially Johan for being my mentor, Isabel for all the weekends spent in DTU-COMPUTE working on the thesis this summer, Ruta for all the help and feedback, Runa, Fernando, Morten and many others.

I would like to thank the IT and administrative staff of DTU-COMPUTE Poul-Erik, Dorte, Christina, Lotte, Fin, Anna, Wanja, Ulla, Camella and all the others that I might have forgot to mention. your constant support and always trying to make matters look seamless.

I am in deep gratitude for the High Performance Computing unit in DTU. For taking all my constant requests and demands professionally for an extra space or more time on the computing nodes and answering my endless stream of questions tirelessly and proving an amazing support for my implementations.

I am indebted to my co-authors for their fruitful discussions, feedback, constant support, and patience with all my questions.

I would also like to specially thank Prof.Vincent Rijmen for giving me the opportunity

to be a guest researcher in Computer Security and Industrial Cryptography (COSIC) research group at the Department of Electrical Engineering of the Katholieke Universiteit Leuven. Thank you for assisting me with your valuable advice, and making sure that I have a good schedule and fruitful research in your team. I am thankful for every one in COSIC team for their welcoming research environment. Especially, I am in gratitude for Qingju Wang for her immense care, kindness, and being my mentor who made sure that I know my way around the group. Thank you for the your feedback, discussion and suggestions for different research direction. I am thankful for Wentao Zhang,Bing Sun, Zhiqiang Liu, and Wenyu Zhang for the team discussion and feedback and discussion on the links between cryptanalytic techniques. Their feedback was instrumental in assisting a better understanding for this topic. I am in gratitude Péla Noé for her constant care, immediate responses and professional coordination while I was in the group. I kept hearing that you are the heart of the group and I tend to agree with that. I am thankful for the discussions, welcoming gestures and friendly conversations that I had with András Bohó, Atul Luykx, Begül Bilgin, Tomer Ashur, Prof. Bart Preneel, Prof.Ingrid Verbauwhede, and Svetla Petkova-Nikova and the rest of the members of the team.

I extend my gratitude for the second family that I have in KU Leuven that embraced me with kindness and support with many feedbacks, lunch-time discussions and motivational attitude Mai Sallam, Enas Mohamed, Costanza and Charito Herrera, Prof.Dominique Schreurs, Sara, Sokaina and Enas Abdulaziz.

Thank you for Emirates Advanced Investments for your continuous support. I am especially grateful for the infinite understanding, support and kindness that Obaid Al-Mansouri has always treated me with.

Heather Mia, Shasha and Kasper and even Holly, Thank you for trying to make Denmark a second home in the time that we have met. Thank you for embracing me with your kindness and love, having all these conversations, sharing all the artistic crafts, meals, green small friends, and memories. I will always cherish that.

Agi, I can not thank you enough for also trying to make it home in here in Denmark for me and for constantly helping me, reminding me to eat well and sleep well.

Robert and Inglise Thank you for helping me with my bike, sharing your musical sheets and taking in all my piano practice for the past one year and a half.I owe a Cello-Piano

session.

Nessy, Shereen, Zahid, Zayyana, Leyya, Ayden, Tabi, Nagwa, Nashwa, Hala, Eno, Emoz and Samah thank you you are true friends, your love and prayers are my true gift in Copenhagen.

I am in deep gratitude for Dr.Maryam AlQimzi, her team Amina, Sarah, Salma, and Fatma thank you for believing in a cure. Thank you for trying. I am here today because of the grace of god and your help.

Last but never the least, my true backbone in life my family, my tribe. Dad, moms, sisters and brothers and many more. Dad thank you for leaving at $5A.M$ to commute two hours for work for the past 30 years and never complain about it. I do not think that there will be a word in the dictionary every to describe my love gratitude. I am thankful that ALLAH choose you for me. 3000 kms away and you are still keeping up with me. I am here because of your believe in me, your prayers, constant motivations, all your $2A.M$ voice messages telling me to do well. I can not love you enough. My chosen family Hanadi, Fatma, Sahar and Asma. Thank you for being true friends. Thank you Hanadi for your constant motivation, being a true sister and friend, giving me your laptop when my stuff were stolen and always motivating me and pushing me toward the finishing line. Fatma Thank you for always being my rock, your advice, motivations and supoprt through everything is precious for me. Thank you both for being my Beta readers I needed that.

At the end I would just say Alhamdullella.

Contents

1 Introduction 1

1.1 Cryptology 1

1.1.1 Where Did it Begin? 2

1.1.2 Modern Cryptology 5

1.1.2.1 Kerckhoffs's Principle 5

1.1.2.2 Information Theory and Cryptography 6

1.1.3 Making the World a Better Cryptographic Paradise 9

1.1.4 Cryptology Meets Information Security 10

1.1.5 Goals of Cryptology 12

1.1.6 What Now? 12

1.2 Scope of This Dissertation 14

2 Cryptographic Primitives 17

2.1 Cryptographic Hash Functions 17

2.2 Public-Key Ciphers 20

2.3 Symmetric-Key Ciphers 22

2.3.1 Stream Ciphers 23

2.3.2 Block Ciphers 25

2.3.2.1 Feistel Scheme 26

2.3.2.2 Substitution Permutation Network (SPN) 27

2.3.2.3 Lai-Massey Scheme 29

2.3.2.4 Addition-Rotation-XOR (ARX) Scheme 30

2.4 Lightweight Cryptographic Primitives 31

2.5 Cryptanalysis 32

2.5.1 Goals of a Cryptanalyst 33

2.5.2 Attack Models . . . 35
2.5.3 Security Models . . . 36
2.5.3.1 Unconditional or Perfect Security . . . 36
2.5.3.2 Provable Security . . . 36
2.5.3.3 Practical Security . . . 36
2.5.4 Generic Attacks . . . 37
2.5.4.1 Exhaustive Search . . . 37
2.5.4.2 Table Look-Up . . . 38
2.5.4.3 Time-Memory Trade-off . . . 39
2.5.4.4 Meet-In-The-Middle Attacks: . . . 40

3 Cryptanalysis Methods . . . 43
3.1 Differential Cryptanalysis . . . 44
3.1.1 Estimating Differential Probability . . . 44
3.1.2 Differential Characteristic to Differential . . . 45
3.1.3 Key Recovery and Data Complexity . . . 48
3.1.4 Truncated Differential Cryptanalysis . . . 50
3.1.5 Impossible Differential Cryptanalysis . . . 51
3.1.6 Higher Order Differential Cryptanalysis . . . 51
3.2 Linear Cryptanalysis . . . 52
3.2.1 Linear Characteristic and Linear Hulls . . . 52
3.2.2 Linear Probability Estimations . . . 53
3.2.3 Key Recovery and Data Complexity . . . 55
3.2.4 Zero-Correlation Cryptanalysis . . . 56
3.3 Other Variants . . . 56
3.3.1 Cube Attacks . . . 56
3.3.2 Integral Cryptanalysis . . . 57

4 Cryptanalysis of PRINTcipher: The Invariant Subspace Attack . . . 59
4.1 Our Contributions . . . 60
4.2 General Idea . . . 60

4.3 Attack against PRINTcipher . . . 61
4.3.1 Description of PRINTcipher . . . 61
4.3.2 An Attack on PRINTcipher . . . 62
4.3.3 Invariant Subspace Description: . . . 63
4.4 Other Attack Profiles . . . 64
4.4.1 Other weak keys for PRINTcipher-48 . . . 64
4.4.2 Analysis of PRINTcipher-96 . . . 65
4.5 Countermeasuere Against the Attack . . . 65
4.6 Statistical Saturation Attacks and Multidimensional Linear Attacks . . . 66
4.6.1 Necessary Background Information . . . 66
4.6.1.1 Notations . . . 66
4.6.1.2 Statistical Saturation Attacks . . . 67
4.6.2 On the Choice of the Values of the Fixed Bits . . . 67
4.6.3 On the Existence of Highly Biased Approximations . . . 68
4.7 Related Work . . . 69
4.8 Conclusion . . . 70

5 Cryptanalysis of SIMON . . . 71
5.1 Our Contribution . . . 71
5.2 General Description of SIMON . . . 72
5.2.1 Structure and Variants . . . 72
5.2.2 Key Schedule . . . 73
5.3 Differential Cryptanalysis . . . 74
5.3.1 Difference Distribution Table . . . 75
5.3.2 Input/Output Differences over F . . . 77
5.3.3 Branch-and-Bound Approach to Differentials . . . 78
5.3.4 Differential Effect . . . 78
5.3.5 Generic Extension by Two Rounds on Top . . . 80
5.3.6 Key Recovery . . . 81
5.4 Impossible Differential Cryptanalysis . . . 83

5.4.1 Key Recovery . . . 85
5.4.2 Complexity . . . 86
5.4.2.1 Expected Size of $\mathcal{K}$ . . . 86
5.4.2.2 Time Complexity . . . 87
5.4.3 Practical Tests . . . 88
5.5 Linear Cryptanalysis . . . 90
5.5.1 Preliminaries . . . 90
5.5.2 Connections and Linear Cryptanalysis of SIMON . . . 92
5.5.2.1 Connections between Linear and Differential Characteristics for SIMON . . . 93
5.5.2.2 A Key Recovery Attack on SIMON Using the Matsui's Algorithm 2 . . . 95
5.5.2.3 Linear Hulls of SIMON . . . 98
5.5.2.3.1 Extending Linear Hulls and Key Recovery Attack on SIMON32/64. . . . 100
the Backwards Direction . . . 100
In the Forward Direction . . . 100
5.5.2.3.2 Attack Complexity . . . 101
5.5.2.3.3 Key Recovery Attack on Other Variants of SIMON . . . 101
5.5.3 Linear Hull Effect in SIMON . . . 102
5.5.3.1 Correlation of the SIMON F Function . . . 102
5.5.3.2 Constructing Correlation Submatrix for SIMON . . . 102
5.5.3.3 Improved Linear Approximations . . . 104
5.5.3.3.1 New 14-round Linear Hulls. . . . 104
5.5.3.4 Key Recovery Attack using Linear Hulls . . . 105
5.5.3.4.1 In the Backwards Direction . . . 105
5.5.3.4.2 In The Forward Direction . . . 106
5.5.3.4.3 Attack Complexity . . . 106
5.6 Related Work . . . 108
5.7 Conclusion . . . 109

6 Links among Integral, Impossible Differential and Zero-Correlation Linear Cryptanalysis . . . 110

6.1 Our Contributions. . . . 111

6.2 Preliminaries . . . 112

6.2.1 Boolean Functions . . . 112

6.2.2 Feistel Scheme Based Ciphers . . . 113

6.2.2.1 Camellia . . . 114

6.2.3 Structure and Dual Structure . . . 114

6.3 Links among Integral, Impossible Differential and Zero-Correlation Linear Hulls . . . 115

6.3.1 Links between Integral and Zero-Correlation Linear Hull . . . 115

6.3.2 Links between Impossible Differential and Zero-Correlation Linear Hull . . . 117

Part (I) . . . 117

Part (II) . . . 118

6.4 New Integral Distinguishers for Block Ciphers . . . 120

6.4.1 New Integral Distinguishers for a Feistel Structure . . . 120

6.4.2 New Integral Distinguishers for Camellia . . . 121

6.5 Conclusion . . . 123

7 Epilogue and Final Remarks . . . 124

Bibliography . . . 127

List of Figures

1.1 Symmetric-key cryptographic system 3

4.1 One round of PRINTCIPHER-48 illustrating the bit-mapping between the 16 3-bit S-boxes from one round to the next. sk_1 denotes the xor key, p the permutation key, and RC_i the round counter. 61

4.2 A subset of PRINTCIPHER-48 s-boxes mapping onto itself. 63

5.1 The SIMON round function 73

5.2 The SIMON key schedule for cases $m \in \{2, 3, 4\}$. The computation on round key k_i depends on k_{i-1} and k_{i-m}, and also k_{i-m+1} in the case of $m = 4$. 74

5.3 Type-1 and type-2 iterated differential characteristics for SIMON 75

5.4 Account of the number of characteristics of a certain probability p (left) and their accumulated probability (right). The first axis is determined as $\lfloor \log_2 p \rfloor$. 80

5.5 Total contribution to the EDP by characteristics of probability in $]p; 2p]$, for every 12-round Simon32/64 differential found with $\mathrm{EDP}(\alpha, \beta) > 2^{-33}$. Each plot represents a single differential. Note that the plots for some differentials overlap, due to identical counts for the characteristic occurences. 81

5.6 Differential Key Recovery Attack on SIMON 82

5.7 Key recovery attack with impossible differentials on SIMON 85

5.8 Progression of the size of $|\mathcal{K}|$ for the key recovery attack on Simon32/64 using the parameters of Table 5.7, as a function of the rotation amount on the input difference (input difference used is $\alpha = (\mathtt{0}\cdots\mathtt{01}) \lll x, x = 0, \ldots, n-1$. The progressions are from the experimental results of Table 5.8. 90

5.9 The keys (in *black*) that should be guessed to attack 17 rounds of SIMON32/64. The red bits are not required to be guessed and the blue bits cost guessing a half bit on average. 97

5.10 The subkey bits (in black) that should be guessed to attack 21 rounds of SIMON32/64. The red bits are not required to be guessed. 101

5.11 The subkey bits (in black) that should be guessed to attack 20 rounds of SIMON 32/64. The red bits are not required to be guessed and the blue bits cost guessing a half bit on average. 107

6.1 Differential Propagation of $\mathcal{F}_{SP}$ and Linear Propagation of $\mathcal{F}_{SP}^{\perp}$ 117

6.2 8-round impossible differential of E 122

List of Tables

2.1 Securiy offered by different key sizes, in the absence of further cryptographic weakness [131], [85] 38

4.1 Subsets of active bits for PRINTCIPHER-96, grouped according to s-boxes . 65

5.1 Members of the SIMON family with their parameters 73

5.2 The z_j vectors used in the SIMON key schedule 74

5.3 Best possible (a, b) pairs for type-1 differential characteristics obtained for Simon32/64 76

5.4 Best diagonal entries of the difference distribution table for $n \in \{16, 24\}$. 77

5.5 Summary of our classical differential cryptanalytic results on SIMON. . . 83

5.6 Truncated differential pattern propagation for SIMON using word sizes $n \in \{16, 24, 32\}$, with an input difference $(0 \cdots 01 \parallel 0 \cdots 0)$ 84

5.7 Results on key recovery attack on SIMON using $|Q| \cdot n$ impossible differentials. The number of pairs used, $n2^{\ell}$ is determined such that the expected size of $\mathcal{K}$, i.e. the remaining key candidates, is 1% of the total subkey space 2^n. The complexities indicated with a † are computed using the approximation of Equation (5.5). 89

5.8 Results from key recovery experiments on Simon32/64, using the parameters of Table 5.7. Note, that half the tests were run during the night, where the server was under less load, hence the difference in the runtimes. 89

5.9 Summary of our differential and Impossible cryptanalytic results on SIMON. Note, that entries with a † in the complexity column indicate results which are worse than brute-force search. The parameters for impossible differentials are such, that the expected fraction of remaining keys after the attack is 1%. 91

5.10 Summary of linear analysis for the different variants of SIMON [16]. In this table **KR** denotes a linear characteristic that can be used trough a key recovery attack, **Dis** denotes a linear characteristic that can be used trough a distinguishing attack and **App.** denotes approximation. 96

5.11 Summary of linear analysis for the different variants of SIMON such that one can mount a linear attack with the success probability of 0.997 [16]. In this table **App.** denotes approximation. 96

5.12 Linear characteristics based on the differential trials by Biryukov et al. for SIMON32/64 100

5.13 SIMON32/K matrices using masks with Hamming weight $\leq m$, nnz = number of nonzero elements

5.14 General analysis to the best and lowest squared correlations in SIMON32/64 for all possible Hamming weights entering the F function 105

5.15 14-round linear hulls for SIMON32/K found, using Hamming weight ≤ 9 106

5.16 Linear cryptanalysis of SIMON using Matsui's Algorithm 1 and 2, and linear hulls 108

Chapter 1

Introduction

Cryptology| κρυπτόλογία

> Information is the resolution of uncertainty.
>
> Claude Shannon:1948 [179]

Since the research and the study discussed in this thesis deal with the design and cryptanalysis of block ciphers. It is imperative to start where the journey has begun, by exploring the definition of cryptology and how the practice of building and analysing cryptographic primitives has evolved over the centuries from art into science. This chapter will briefly introduce the general perception of cryptography, cryptanalysis and the different concepts involved, their importance and the vital role of a cryptographic primitive known as block ciphers in this field.

1.1 Cryptology

As defined in literal and scientific texts, *cryptology* is the art and science of designing and analysing algorithms that serve as primitives to establish information security goals such as confidentiality, integrity, authentication and non repudiation in different information systems deployed in various application environments. These goals will be discussed in details subsequently in this chapter. In a view of the previous definition, cryptology has always been mapped to two main lines of study *cryptography* and *cryptanalysis*.

Today, cryptography can be finely defined as the aspect of the mathematical design and implementation of the fundamental components that will maintain information security goals within certain cryptographic and security margins. These fundamental components are described as cryptographic primitives.The other face of the coin is cryptanalysis which is defined as the art and science related to evaluating, verifying and testing the designed cryptographic primitives and pushing them through all possible claimed or non-claimed security margins. The exact definition of these security and cryptographic margins will follow within this text.

1.1.1 Where Did it Begin?

Cryptology is one of the fields that started as an art and morphed into science over the centuries and is contributed to through a mixture of different cultures and disciplines. Its existence was traced back to more than 4000 years ago where it was detected in the Sumerian scripts which is one of the earliest systems of writing in the form of logographic and syllabic units in the late fourth millennium B.C. It was also encountered in the Egyptian hieroglyphs around 1900 B.C. Hieroglyphs combine logographic and alphabetic components to establish a formal writing system that were used to express mainly the religious literature of that period. These two systems of writing were considered for a long course of time a hidden or secret approach of communication among two different civilizations.

The best known efforts to reveal these scripts were in the form of "decipherment" by the Egyptian Horapollo in the fifth century in the Greek text of Hieroglyphica that revealed 200 hieroglyphic symbols [108]. Additionally, in the 5th, 15th and the 17th centuries a collective Arabic, Persian, Italian, English, Danish and German efforts were also made over different periods of time to decipher the Sumerian scripts. The Greek have played a crucial role as well in establishing ancient cryptology. The word cryptology itself is originated from the Greek phrase kryptos-graphein or kryptos-logia which means the hidden or secret writting or study. The first notion of cryptology was resembled in the fact that hidden communication was established through concealing secret messages. For example, Herodotus stated that ancient Greek wrote secret messages on wooden plates then covered them with layers of wax to hide them. Additionally, in ancient China paper masks and hidden secret letter sheets were used to exchange public messages that contain secret ones.

It should be noted that the definition of cryptology in this sense was reduced to its ultimate minimum where the essential goal was to hide information and establish a level of secrecy or confidentiality. In current texts the science of hiding information and making sure that the secret information does not exist for unauthorized parties is referred to as *steganography and watermarking* [192]. One of the first ciphers detected was in the Hebrew scriptures in addition to the scytale *transposition cipher*. It was used by the Greek Spartan military in the 7th century B.C [164]. In such ciphers the positions of the original letter or group of letters will be permuted and rearranged to different positions according to a specific system. Scytale was originally in the design of a strip of leather that has the message written on it and wrapped around a cylinder of certain diameter. The correct diameter will enable reading the message.

At this stage, we are one step further at establishing confidentiality level of cryptology. A communication channel between two parties in the roles of sender and receiver is established. The secret messages are sent scrambled and remain so for unauthorized parties. The message which contains intelligible information is known as *plaintext*. It will be transformed, using a dedicated algorithm for this purpose known as *cipher*, to "secret" unreadable message called *ciphertext* or *cryptogram*. This method is known as

encryption. The generated message can only be read by parties that have the knowledge of the same secret information. This secret is combined with the algorithm to reverse the process of encryption and produce the original plaintext. This secret information is referred to as the *key* and the reverse algorithm is known as the *decryption* process. The transmission of the ciphertext can happen in the presence of an adversary or an unauthorized party. It is noteworthy to state that some texts differentiate between the word decipherment and decryption. The former is the process of authorised decryption and the later is the process of unauthorised decryption [192]. The cryptographic system presented at this scenario depends on one secret shared information between the communicating parties this is commonly referred to as *symmetric-key cryptography*, as shown in Figure 1.1.

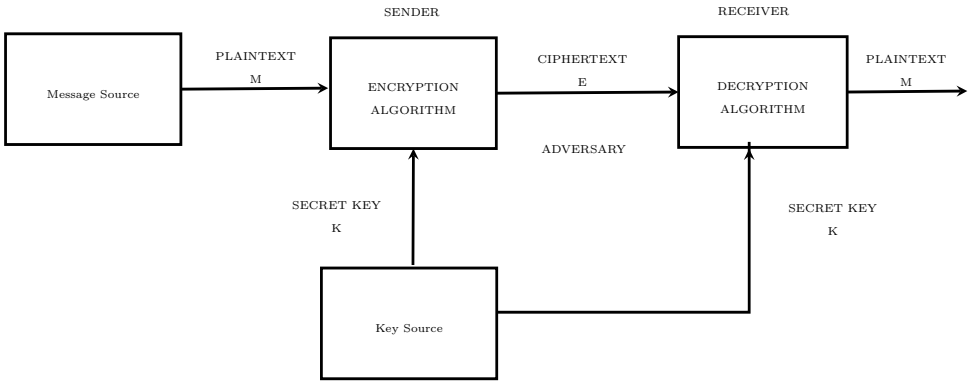


Figure 1.1: Symmetric-key cryptographic system

The Roman general, politician and consul Julius Caesar has used his famous Caesar *substitution cipher* to communicate with his ministers and army generals during combat and war periods. Each plaintext letter used to be replaced with letters of certain shifts from their original positions in the alphabets. In order to perform decryption it is important to know the cipher shifts performed which constitute the secret key. The number of possible secret keys for the texts based on English letters only at this stage is $26\,! = 2^{88.38}$ keys. This example of *substitution cipher* is referred to as *monoalphabetic* since fixed substitution is applied over the entire plaintext.

Encryption at this stage has become a very popular form of cryptology. On the contrary to the common belief, encryption was not limited to the application of military communication. It was conjointly used by different mathematician, poets and scientists to hide or encrypt secret messages meant for specific recipients or to hide their scientific findings as in mathematical equations or chemical potions and remedies. In the 700's A.D the Arabic philologist Al-Farahidi performed one of the basic examples of cryptanalysis. He used permutations and combinations to list all possible vowel and non-vowel based Arabic words to be used later on to apply frequency analysis on certain ciphertexts. In his *Book of Cryptographic Messages*, he describes using cribs which are schedules of plaintexts and their encrypted versions. Frequency Analysis method was

used to break some of the substitution ciphers in the 800s A.D by Ibn Ishaq AlKindi as described in his manuscript *On Deciphering Cryptographic Messages* [56] [**?**]. In this form of cryptanalysis statistical analysis is performed on the frequency of occurrences of certain letters or word combinations. A correlation of cipher frequencies to plaintext frequencies and letters distributions in the original system understudy will assist the adversary to guess the original message.

As a countermeasure to the application of frequency analysis on monoalphabetic substitution ciphers, an evolution to the concept of substitution was introduced . Homophonic substitutions were performed to substitute each letter with more than one ciphertext unit where high frequency letters are mapped to more lower frequency letters to influence the frequency distribution. Another example are diagraphic and polygraphic substitutions were plaintext is subtituted in pairs of letters or large group of letters in opposition to one letter at a time. In 1585, Blaise de Vigenère published his polyalphabetic substitution cipher that was known as *le chiffre indéchiffrable*. The method of the cipher is originally credited to Bellaso in his book *La Cifra Del* published in 1553. It is based on using square table for substitutions called *tabula recta*. The first row is the 26 plaintext alphabets in English and the rest of the rows are shifted subsequently one position to the left at each row. This is equivalent to adding plaintext to the key letters' position modulo 26. The encryption will be then based on a keyword ciphertext that will be used for the whole plaintext repeatedly. If certain amount of ciphertext obtained by an attacker under the same key, then it is possible to analyse gaps between repeated sections of the ciphertext, get information regarding the key length. Then the attacker can organize the ciphertext into blocks of the size of the key and perform frequency analysis on certain group of ciphertext words at a time. This is known as Kasiski examination [202].

Since World War I until the early 1960's there were several electro-mechanical implementations of polyalphabetic substitution ciphers which are referred to as rotor cipher machines. Their concept started as sole implementation of substitution ciphers, then they were combined with transpositions to morph into *product ciphers*. The design evolved to resemble modern ciphers with iterated application of the encryptions to the input at different stages called rounds. One of the most famous examples are the *Enigma* machine used by the Germans in the World War II in addition to *TypeX* by the British military, *RED* by the Japanese navy and *ECM Mark II* by the United States military among many others. The main set of elements of these machines are the rotors. They are fixed different substitutions of letters wired together to be selected in a composition of groups. After the encryption of each letter in the plainetxt the rotors will change positions to change the substitution chosen for the next letter. World War I and II were two of the main historical events that nudged the cryptographic communities, at this point being mainly the military and governmental intelligence, to utilize their cryptographic tools and use them, hence making them publicly available intentionally or coincidently. Consequently, risking them to be subjected to different cryptanalytic approaches. The Enigma was broken by the British Royal Navy and the Polish Cipher Bureau separately and this was declared in 1939. It enabled the Allies in World War II

to disclose encrypted Morse-coded messages between the Axis powers.

One of the tools that was also present, and enabled for cryptographic purposes at that stage is a family of codes. Codes in this context are the schemes that are used to convert data from one form to another to facilitate their transmission through certain bounded, mainly by the size of the data, communication channel. This conversion is usually governed by a *codebook* in which data conversion to codewords or encoding and decoding is stated. *Morse Code* is one of the earliest examples that converted alphabets to dots and dashes to transmit data efficiently over telegraph channels. However, for these codes to be used for cryptographic purposes, then the codebooks should be secret to be known only between the sender and the receiver as they will mimic the role of the key and the encryption and decryption algorithms [1]. In today's cryptographic systems a category of error-correcting codes are utilized. They are mainly used in recovering correct data from noisy channel's output which might contain accidental errors introduced by the channel. They are mainly used as an element in the design components of certain type of cryptographic algorithms as in Maximum Distance Separable codes in block ciphers. Their impact on the design will be discussed in later chapters. The form of cryptology discussed in this section is still restricted to providing confidentiality through using symmetric-key cryptographic systems.

1.1.2 Modern Cryptology

The modern formation of cryptology and cryptographic models started with certain articulation of rules, conditions, requirements around the structures of cryptographic units that were available at definite points of time. This indicated an unmistakable collaboration between the cryptographic design and cryptanalytic approaches. Cryptanalysis attempts assisted in maintaining a state-of-the-art design principles when it comes to the cryptographic structures available then and now. The presence of modern computational power combined with these new cryptographic models pushed for the availability of new designs with tighter security requirements and better complexities. Security margins of these designs are constantly pushed and redefined by the existence of a continuous cryptographer/cryptanalyst game. It is worth noting that up to this point of the discussion, the presented cryptographic system is mainly characterised by a system that provides confidentiality which is mainly represented by encryption algorithms or ciphers.

1.1.2.1 Kerckhoffs's Principle

One may trace these formulations to Auguste Kerckhoffs in 1883 when he published his article " La Cryptographie Militaire" in Le Journal des Sciences Militaires [122]. He summarized his point of view in the practical design principles of ciphers as follows

- The cryptographic system should be unbreakable either theoretically or practically.

- The design of the cryptographic system should not be a secret, and if compromised that should not be problematic to the corresponding parties. Only the value of the key should be secret.
- The key should be easy to remember and change.
- The cryptogram should be short to be efficiently transferred by telegraph.
- The related equipments or records of the cryptographic system should be transferable and operable by one personnel.
- The cryptographic system should be easy, which means it should not require extensive knowledge or effort to be appropriately used.

The second principle is what is commonly known as *Kerckhoffs's principle*. It is essentially interpreted into the fact that the security of the cryptographic system depends on keeping the secrecy of the value of the key not the design of the system itself. In this sense, the cryptographic system design can be publicly available and shall withstand cryptanalysis or reverse engineering approaches. This perspective opposes the direction of providing security through obscurity that was mainly maintained throughout the centuries especially by governmental agencies. Designs and implementations related to cryptographic or security applications are always meant to be secret.

However, Kerckhoffs openness meant that the trust around the security or cryptographic systems are earned rather than given by claimed design rationales if any were made available by the system designers. It introduced the possibility to reveal potential design flaws, weaknesses and exploits in the system by the cryptanalytic community. Such flaws and weaknesses may not be easily detected by the designers or might simply be ignored to support an invalid claim of trust in the security provided by the system. Nevertheless, Kerckhoffs's principle granted the opportunity to construct different levels of trust when it comes to different cryptographic designs. This trust is based on the level of scrutiny and thorough analysis, by the cryptographic community, that the designs are withstanding. It also enhanced the cycle of evolution between the complexity of the presented cryptographic designs, and cryptanalytic approaches and scrutiny around these designs. In addition to, enhancing the available designs and the associated inspection and cryptanalysis approaches. This openness introduced the opportunity to encapsulate new cryptographic structures and created a wider spectrum for constructing new design approaches. Since then an unwritten collaboration between designer and attackers were formed to provide better level of trust in the available cryptographic systems.

1.1.2.2 Information Theory and Cryptography

In 1945, Claude Shannon , being the father of information theory, introduced the next remarkable phase of modern cryptography where he presented a mathematical model for cryptographic and secret systems based on concepts of information theory [179], [180].

In his two articles "A mathematical theory of cryptography" and "A mathematical theory of communication", Shannon was seeking to investigate the issues of cryptographic and secrecy systems as an application of communication theory. While working on efficient transmission of information through communication channels he came up with what is known today as the fundamentals of cryptographic design. The cryptographic model of secret system presented by Shannon can be seen in Figure 1.1.

Shannon proposed the influential definitions of *theoretical secrecy*, *perfect secrecy* and *practical secrecy*. He was among the first to utilize the primary elements of information theory as in *entropy of information*. Entropy can be defined as the average extent of information within a single random variable or communicated information from a source. In other words, it can be used as a mean of measurement of randomness or uncertainty of an output of a source. It is believed that entropy provides a lower bound on the expected work needed to guess a single random variable from a source [71], [151]. This definition was extended to the fact that an ideally secure information system should not be differentiated from random information. Shannon also outlined the concept of a *random cipher* where he described it as the decryption possibilities of a ciphertext are a random selection from the possible space of messages. Furthermore, he also discussed the notion of an *ideal cipher* where he defined it as all statistics of the ciphertext are independent of the specific key being used for encryption.

Shannon also proposed the main five criteria to evaluate a secrecy system. They can be summarized as follows:

- **Amount of Secrecy:** The amount of secrecy in a system can be categorized as perfect, non-uniquely and uniquely solvable secrecy systems. In a perfect secrecy system the attacker has no advantage even if he intercepted the secret encrypted material. In non-uniquely solvable secrecy system the attacker can detected a secrecy system through obtaining some information, however no unique solution to the ciphertext is presented. In uniquely solvable secrecy system the attacker can detect the secrecy system and can get the unique solution of the ciphertext if certain amount of material is given and if the amount of labor to effect this solution is presented.

- **Size of Key:** The key should be as small as possible in order to support secure transmission from origin to destination.

- **Complexity of Enciphering and Deciphering Operations:** Simplicity is the main component in the enciphering and deciphering algorithm. This is meant to reduce complexity and mechanical cost of different implementations. In addition to avoiding errors and loss of time in the case that these processes are done manually.

- **Propagation of Errors:** It is essential to reduce the possibility of error propagation. Errors can propagate to ciphertexts, and expand when deciphering is applied on them. This might lead to a need for repeated transmission of the ciphertext or risk loss of information transmitted.

- **Expansion of Messages:** Encryption might increase the length of the message. This is an unfavourable feature in a secrecy system, as it is possible to flood the message statistics by addition of nulls or using multiple substitutes.

In the subject of cipher design, Shannon suggested an alternative system for an ideal secrecy system to make it difficult to apply common cryptanalysis approaches at that point of time. He has stated that a secrecy system should maintain components of *confusion* and *diffusion*. He referred to diffusion as the characteristic of hiding the redundancy in plaintext statistical structures and ciphertext by ensuring that they are fully dissipated which is aimed to control how the output bits are dependant on the input bits. On the other hand, confusion was referred to as making sure that the connection between statistical structure of the ciphertext and the description of the key is as complex and involved as possible.

Shannon outlined pure and mixed secrecy systems or ciphers. Pure systems meant that all keys are equally likely and for any given product transformations in the set of transformations. All keys are equivalent and they all lead to the same set of possibilities as in substitution cipher with random key.

To support the application of the previous concepts of confusion and diffusion, Shannon proposed the design of *product ciphers*. The operations were focused around mixing transformations of modular arithmetic, substitution and permutation or transposition being the main components to be used for a cipher with better security. The diffusion in this case is represented by the transposition or permutation component and the substitution component is representing the confusion property in a cipher. These components were applied repeatedly and alternatively in what are called mixing transformations. They represent what are today known in modern cipher design as encryption or decryption rounds.

According to Shannon the statistical methods that solve a secrecy system should be characterized by being simple, key dependent more than message dependant, and focused on the extraction of the right key or significant, simple and usable amount of information related to it. He used the concept of *unicity distance* of a cipher to estimate the minimum amount of ciphertext needed to recover the unique encryption key by a computationally unbounded enemy cryptanalyst.

It is adequate to state at this point that Shannon was a pioneer in his everlasting contributions to the foundations of the current state-of-art of the cryptographic and cryptanalytic designs.

Some of these concepts will be discussed further in the next chapter to provide a detailed analysis on how they influence the different concepts of cipher design and cryptanalysis approaches.

So far we have seen that cryptology is a constantly evolving discipline that has significant contributions from the different disciplines of mathematics, philology, linguistics, translation, information theory, communication, mechanical engineering and electrical

engineering. For a detailed investigation on the different cryptographic contributions throughout history the reader can refer to The Codebreakers by Kahn [120], and The Code Book by Singh [182]. In addition to the technical state of the art in The Handbook of Applied Cryptography by Menezes *et al.* [154], and the Encyclopaedia of Cryptography and Security [190].

1.1.3 Making the World a Better Cryptographic Paradise

The introduction of computers and computing power by Charles Babbage, Alan Turing and many others in the early 18th century and 1930's increased the reliance on digital communication systems [189]. Consequently, this initiated a demand to provide security and cryptologic structures with wider range of goals and services to secure data in the digital medium. Such efforts pushed cryptology from being a matter of study that is being mainly initiated and practised in confidential venues to an active, rich and scientific research topic in the public academic domain. In 1970's, Horst Feistel has initiated the design of *Feistel network for block ciphers* at IBM labs [86]. This structure of cipher design was used in 1971 to construct *Lucifer* family of block cipher with 16 rounds ,and different key and block sizes [184]. It is considered the first civilian block cipher where the message is divided into blocks of fixed length and each is encrypted using the same secret key. The combination of all of these encryptions will constitute the final ciphertext. It is worth mentioning at this point that block ciphers are clear examples of product ciphers that are based on an iterative round function. In 1973, the National Institute of Standards and Technology (NIST) and what is known then as the National Bureau of Standards (NBS) announced a call for symmetric-key encryption standard where the design is based on secret key encryption. The aim of the call was to provide a suitable candidate for encryption of digital, sensitive and unclassified government data. In 1974, a second call was issued due to the inadequacy of the submitted candidates to the first call. Lucifer based design was submitted to the second call of *Data Encryption Standard* (DES). The submission was reduced to a key size of 56 bits and block size of 64 bits by the National Security Agency(NSA) and was adapted as Data Encryption Standard [80]. It was published as a Federal Information Processing Standard (FIPS) in 1977 [165]. DES was widely adapted by different vital public, financial and governmental sectors all over the world. DES has been subjected to many cryptanalysis approaches in order to test and validate the security claims of the design. To mention but a few examples are the differential cryptanalysis by Shamir in 1980, linear cryptanalysis by Matsui in 1993 and Davies attack in 1997 [37], [153]. Although these approaches have mainly obtained theoretical results, they are unobtainable in practice. However, a brute force attack to search the key space of DES was possible first by using the Electronic Frontier Foundation (EFF) DES cracker in 39 days. Later, the decryption of a DES ciphered message was possible in 22 hours and 15 minutes [88].

In mass networks of communication, a large number of users will require the secret key to be exchanged to perform encryption and decryption based on symmetric-key encryption

as in DES. This distribution of keys requires a proper key sharing and management system. In 1976, Diffie and Hellman proposed a key exchange protocol to address this problem [200]. The protocol is based on discrete logarithm problem where modulo powers are defined with regards to certain multiplicative cyclic group and they are easy to compute and hard to invert. Sender and receiver have a pair of keys, private and public. These will be used to compute a shared key that will serve as the secret key for the symmetric-key encryption decryption algorithm. In 1978, asymmetric-key cryptology was initiated with the introduction of RSA encryption algorithm [171]. In this system, each communicating party will have a pair of keys, public and private. If used for encryption and decryption, the sender will use the recipient's public key to encrypt message, and the recipient will use his secret key to decrypt the message. Public keys in this context are tied to communication party and known by everyone interested in communicating with that party.

1.1.4 Cryptology Meets Information Security

The high reliance on digital information systems, computing systems and the internet, urged the scientific communities to provide better secure solutions for the different applications and user profiles available. There is a huge scale of digital infrastructure today in e-commerce and banking systems, education systems, political voting systems, governmental and defence systems, health systems, entertainment and gaming systems, energy and power systems, transportation systems, communication systems and many more. Information processed in these systems are enormous, sensitive, classified into different privacy structures and essential to employ in our daily activities. The volumes varies from between huge database structures to embedded systems solutions that made it possible to present and operate the information in different real-time and computationally constrained environments. The examples of common applications are endless. To mention a few applications that are used most frequently, there are electronic mails, mobile communication, electronic passports, mobile pay solutions, Short Messaging Systems (SMS), banking cards as in Automated Teller Machine (ATM), credit and debit cards, Subscriber Identity Module (SIM) cards, Radio-frequency identification (RFID) Tags and electronic video games.

In order for these infrastructures, systems and applications to operate as expected successfully with the least possible risk of disruption (either accidental or intentional), the information entailed in these systems shall be protected. The protection level provided should match the present environment constraints and requirements in addition to the different profiles and classifications of users and data to be accessed. Therefore, the scope of *information security* and cryptology definition and objectives was escalated from being centred around providing secure confidential system to being able to provide a volume of services that will ensure the protection of information. *Information security* can be roughly defined as the collaboration of different disciplines to provide protection for data and information in their digital or physical form from loss, unautho-

rized disclosure, access, use, modification, disruption, inspection and recording. This is achieved through providing different algorithms, implementations (either in hardware or software), mechanisms, protocols, risk assessments, policies, standards, laws and regulations to ensure that the goals of securing information system are achieved. The vital goals of information security are privacy of confidentiality, data integrity, entity or data authentication, authorization, validation, access control, non-repudiation, anonymity, timestamping, revocation and many others. Cryptology in this sense has morphed from a focus of confidentiality and secrecy systems to become the backbone of information security that provides different primitives suitable for specific different application environments with specific criteria [154].

For example, security protocols that are meant to provide security services for different layers of communication networks are based on different *cryptographic primitives*. They are well-defined, fundamental and well-established cryptographic algorithms that are meant to serve as detailed components and building blocks for these protocols in order to provide essential security goals within an information system. This classification of cryptographic primitives includes symmetric-key primitives, asymmetric-key primitives (or public key primitives) and unkeyed primitives. *Symmetric-key primitives* are defined as a set of cryptographic algorithms that depends on using the same cryptographic key or set of keys produced by a key scheduling algorithm for single process as in encryption and decryption or hashing. They include symmetric-key ciphers as in block ciphers and stream ciphers, keyed hash functions as in Message Authentication Codes (MACs), and pseudorandom sequences. As for asymmetric-key primitives they are the set of cryptographic algorithms that depends on using two different profiles of cryptographic keys produced by a key generation algorithm. Secret or private keys and public keys are used to perform two different functions as in encryption and decryption of signature and verification. This class of primitives include public-key ciphers and signatures. Typically, the suite of algorithms that are mainly used to perform specific functionality as in decryption and encryption are referred to as a *cryptosystem*. It is used in reference to suite of key generation or key scheduling, encryption and decryption algorithms, especially in public key ciphers.

Furthermore, unkeyed primitives are the set of cryptographic primitives that are not dependent on the existence of a key in its structure for certain functionality to be performed. This includes one-way permutations, hash functions and random and pseudorandom sequences and their generators. It is worth noting that the previous provided examples are not the only existing cryptographic algorithms in each category [154], [168], [135].

Some of the prominent security protocols that rely on these primitives are Transportation Layer Security (TLS), Secure Socket Layer (SSL), Secure Shell (SSH), Internet Protocol Security (IPsec). These protocols and primitives can be combined together to establish a system that implements them and their associated infrastructure including security for the users, data, key management. This is what can be referred to as *cryptographic scheme or cryptographic system*. Public Key Infrastructure (PKI) and their certification authorities can be considered a clear example of a cryptographic system [1].

1.1.5 Goals of Cryptology

In order for any cryptographic primitive or cryptographic framework to be used for any security purposes, It should achieve at least one of the following goals:

- **Confidentiality:** This goal aims to make sure that data are being accessed and read only by authorized parties. This is mainly achieved through using primitives that supports encryption and decryption algorithms .
- **Data Integrity:** This goal aims to detect alterations and manipulations in the data, messages or communication origins initiated by unauthorized parties. Hash functions can be used to achieve this goal. Data integrity can be also referred to as *data authentication*.
- **Entity Authentication:** This goal aims to provide verification to the identity of communication parties. This is achieved by using combination of different primitives as in public key ciphers, hash functions, signatures and certifications.
- **Non-repudiation:** This goal aims to prevent any communication party from denying previous liabilities, actions or commitments. This can be achieved through using a combination of signatures, hash functions or PKIs.

It should be mentioned that it is possible to construct a certain primitive using another. For example, we can use block ciphers to construct hash functions through different hash construction methods as in Merkle–Damgård, Davis-Meyer, Matyas-Meyer-Oseas and Miyaguchi-Preneel [154]. Moreover, block ciphers can be built from hash functions as well using Luby-Rackoff constructions [149].

1.1.6 What Now?

In recent years, global surveillance programs as PRISM, ECHELON, Carnivore, DISHFIRE, STONEGHOST, Tempora, Frenchelon, Fairview and MYSTIC that are being operated by different intelligence agencies and others around the world have been revealed. The main goal of such programs is to breach users' privacy regardless of the security products, tools and services instilled on their systems [201]. Consequently, such activity has motivated a cautious study to the current state of art of security frameworks provided across all platforms.

It has been established throughout this chapter that cryptology has matured to certain framework of well designed components. Since 1980's cryptographic primitives are under a constant, rigorous and innovative cycle of analysis, and improvement to reach a certain acceptable security level. One of the factors that plays a crucial role in this cycle is the *security margin* of these primitives which is usually difficult to quantify. It is the measure that can indicate how much effort will be needed to analyse certain

cryptographic primitive in order to break it, so that it cannot maintain its functionality (security claims, or cryptographic goals and requirements). It is mainly estimated based on the cryptanalytic effort spent to break that primitive (data and time complexity) which is usually defined as an upper bound to break that primitive. In addition to how simple, complex or efficient is the design structure given in particular environment or mode of operation. For example, the percentage of round broken compared to the overall design, the ease of implementation in hardware of software, the number of bits to encrypt per second are all considered when the suitability of primitive for a specific environment is evaluated [154] [1]. The contribution of the different academic, industrial and governmental communities has constantly pushed and modified these security margins, hence improved in average cases the quality of cryptographic primitives provided. This was established through public research calls or privately funded research projects.

To mention a few, an Advanced Encryption Standard (AES) call was initiated by NIST in 1997 to find an alternative symmetric-key encryption standard other than DES. DES was subjected to the scrutiny a volume of cryptanalytic attacks especially due to its small key space of 56 bits that made it vulnerable to brute force attacks that we will discuss in the next chapter. After two rounds of analysis on fifteen different designs, NIST announced in 2000 that Rijndael is the selected proposal for AES. Rijndael was adapted in 2001 in the Federal Information Processing Standard (FIPS) PUB-197 as AES [163], [72].

Moreover,the New European Schemes for Signatures, Integrity and Encryption (NISSEE) project in 2000 initiated a call to identify and evaluate cryptographic primitives in the categories of block ciphers, stream ciphers, public-key encryption, cryptographic hash functions, digital signatures and identification schemes. It received 42 submissions and in 2003 out of which 12 submissions were selected. Selected designs included Camellia, MISTY, AES, RSA with Key Exchange Mechanism (RSA-KEM), HMAC, CBC-MAC, WHIRLPOOL, SHA-256, SHA-348, SHA-512 and Elliptic Curve Digital Signature Algorithm (ECDSA). However, none of the submitted six stream ciphers were selected and this lead to the eStream project call in 2004. It was organized by the European Network of Excellence in Cryptology(ECRYPT) to identify stream ciphers that can be suitable for different application profiles and environment [5]. The project was carried out into three different phases and resulted in identifying stream ciphers for software environments with high throughput demands, and stream ciphers for hardware environment with limited and constrained resources (storage, power and energy consumption and gate count). It is worth noting that few of the candidates submitted to this project have provided authentication in addition to encryption in their proposed designs, however non of the ciphers with such profiles made it to the final phase. Finalists in 2008 included Trivium, Salsa20/12, Rabit, HC and Grain.

In parallel to NISSEE project, the Japanese government announced the Cryptography Research and Evaluation Committees (CRYPTREC) project in 2000 with the same profile to evaluate and recommend cryptographic techniques for governmental and industrial purposes. This resulted in a recommended cipher list for different purposes

and environments that is different from NISSEE's project selected list. The proposed list and recommendations are still being updated and revised as the state of the art of these primitives change over the span of time. As in the formed recommendations in ECRYPT workgroup for lightweight cryptographic algorithms in 2010 that are meant to evaluate cryptographic primitives meant for constrained environment including sensor nodes, smart cards and RFIDs. These recommendations initiated the presence of lightweight hash functions, block ciphers and protocols as in HIGHT, PRESENT, KATAN, KTANTAN, mCrypton, PHOTON and SPONGENT. Moreover, NIST announced a hash function competition call to construct a new hash function called SHA-3 in 2007. The competition was composed of two rounds where 51 designs were received for the first round ,and 14 survived the analysis for the second round. At the end of the second round the only five finalists that were selected were Keccak, BLAKE, Grøstl, JH and Skein. In 2012, the final announcement was made to declare that Keccak is the selected SHA-3 standard in FIPS-PUB-202.

One of the latest additions to the cycle of improvement is the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) which was recently started in early 2014. The aim of CAESAR is to identify and provide authenticated-cipher designs that are better than the current alternatives (AES-GCM), and will easily be used in different implementation environments [6].

1.2 Scope of This Dissertation

The main focus of the research material and results provided in this dissertation is to analyse and evaluate the security of selected block ciphers. It consist of two main parts. The first part is a general introduction to cryptography, structures of cryptographic primitives, and cryptanalysis techniques. The second part is selected publications of block cipher cryptanalysis that were obtained throughout the PhD study period. In particular, the thesis assess and evaluate the security of the lightweight block cipher PRINTcipher, NSA's Family of lightweight block cipher SIMON, block cipher Camellia.

The outline of this thesis is stated as the following:

- **Chapter 2.** This chapter will briefly introduce the main symmetric-key cryptographic primitives, their cryptographic design strategy, and security requirements. Finally, it gives the description of cryptanalytic attacks, their goals, complexity and models.
- **Chapter 3.** This chapter will outline the main cryptanalytic techniques and methods used for symmetric-key primitives in general and block ciphers in specific. It mainly focus differential and linear cryptanalytic techniques which are used to provide results in the rest of the chapters.

- **Chapter 4.** This chapter proposes a new attack named the Invariant Subspace Attack. It is utilized to break the full block cipher PRINTcipher for a significant fraction of its keys. This attack can be seen as a weak-key variant of a statistical saturation attack. For such weak keys, a chosen plaintext distinguishing attack can be mounted in unit time. In addition to breaking PRINTcipher, the new attack also gives us new insights into other, more well-established attacks.In addition, we also show that for weak keys, strongly biased linear approximations exists for any number of rounds. In this sense, PRINTcipher behaves very differently to what is usually assumed.

- **Chapter 5.** In this chapter we provide a variety of attacks on the family of lightweight block cipher SIMON that was published by the U.S National Security Agency (NSA). The ciphers are developed with optimization towards both hardware and software in mind. While the specification paper discusses design requirements and performance of the presented lightweight ciphers thoroughly, no security assessment is given. This chapter is a move towards filling that cryptanalysis gap for the SIMON family of ciphers. This chapter present a series of observations on the presented construction that, in some cases, yield attacks, while in other cases may provide basis of further analysis by the cryptographic community. Specifically, The attacks obtained are using classical- as well as truncated differentials. In the former case, this chapter show how the smallest version of SIMON, exhibits a strong differential effect.

  In addition to that, this chapter also investigate the security of SIMON against different variants of linear cryptanalysis, i.e., classic linear and linear hull attacks. It presents a connection between linear characteristic and differential characteristic, multiple linear and differential and linear hull and differential, and employ it to adapt the current known results on differential cryptanalysis of SIMON to linear cryptanalysis results. Our best linear cryptanalysis results are using average squared correlation of the linear hull of SIMON based on correlation matrices. The results cover 21 rounds of SIMON 32/64 out of 32 rounds with the data complexity $2^{30.56}$ and time complexity $2^{54.56}$. We have implemented our attacks for small scale variants of SIMON and our experiments confirm the theoretical biases and correlation presented in this work. So far, The results presented are the best known with respect to linear cryptanalysis for any variant of SIMON.

- **Chapter 6.** In recent years, the discussion to establish links among different cryptanalytic techniques has been actively revisited. In this chapter, the known results on the links among integral, impossible differential and zero-correlation linear hulls presented by Bogdanov *et al.* and Blondeau *et al.* recently are considered. In this chapter, it is proved that constructing a zero-correlation linear hull always implies the existence of an integral distinguisher. Moreover, it shows that constructing zero-correlation linear hull on a Feistel structure with SP-type round functions, where P is a binary matrix, is equivalent to constructing impossible differential on the same structure except that P is substituted by the

transposed matrix P^T. Additionally, with the help of the newly established links, the following results are obtained:

  - The first known integral distinguishers of 5-round Feistel structure with bijective round functions and 3-round Feistel structure with round functions not necessarily being bijective.
  - The best known integral distinguishers of Camellia so far, i.e., 7-round integral distinguishers of Camellia with FL/FL^{-1} layer and 8-round integral distinguishers of Camellia without FL/FL^{-1} layer.

- **Chapter 7.** In this chapter, the final brief conclusion and remarks around the different research topics discussed and approached will be presented.

Chapter 2

Cryptographic Primitives

> Cryptography is about communication in the presence of an adversary.
>
> Ronald Rivest:1990 [172]

Cryptographic primitives are the mathematical functions or algorithms that serve as the building blocks in different cryptographic structures and frameworks in order to achieve cryptographic goals as we have mentioned in the previous chapter. These primitives belong to three classes of algorithms as in symmetric-key primitives, public-key primitives and unkeyed primitives.The design of these cryptographic primitives are always focused on security and performance and the possibility to provide a perfect balance, or a better trade-off between them both. This chapter presents an overview on structures within each of these classes such as *cryptographic hash functions*, *public-key ciphers*, and *symmetric-key ciphers*. It also presents the general concept of cryptanalysis and attacks that can be applied on these structures, and the different attack models and their complexities.

2.1 Cryptographic Hash Functions

A hash function is a computationally efficient mathematical algorithm that maps an input of a message (binary strings) of an arbitrary length to a small and unique output of fixed length that is called *message digest*, *hash value* or *fingerprint*.

Definition 1. *A cryptographic hash function $H : X \to D$ is a one-way mathematical structure that maps input of arbitrary length $X \in \{0,1\}^*$ to a unique output bit strings of fixed length $H(X) \in \{0,1\}^n$ where n is a positive integer less than * value.*

The classes of hash functions can be divided into *keyed* and *unkeyed* constructions. Unkeyed hash functions accept a single input which is the arbitrary message to produce the hash value. A prominent subclass of unkeyed hash functions are Manipulation Detection Codes (MDCs). Alternatively, keyed hash functions will take two inputs a

secret key of fixed length, and an arbitrary message as inputs to produce the output also known as hash value. Message Authentication Codes (MACs) are important subclass of keyed hash functions.

Definition 2. *A keyed cryptographic hash function* $H_k : (X, K) \rightarrow D$ *is a one-way mathematical structure that maps inputs of arbitrary length* $X \in \{0,1\}^*$ *and* $K \in \{0,1\}^k$ *to a unique output bit strings of fixed length* $H_K(X) \in \{0,1\}^n$ *where* n *is a positive integer less than * value, and* k *is the size of the key bits used.*

The main two essential properties portrayed in the definitions of the two classes of hash functions are compression of any input to a hash value of fixed length, and ease of computation of the hash value of any input.

Hash functions are among the core components in many cryptographic applications as in *digital signatures, entity authentication and identification.* For efficiency and security purposes, in digital signatures the message is being hashed then signed. This way the signing algorithm will process less amount of data, and any tampering or forgery will be detected. They are usually used in combination with public-key ciphers to achieve the previous result.

Moreover, hash functions are also used in *message authentication* to validate the integrity of the data sent. This is achieved through securely sending a hash value of the message with the message to the recipient. It enables validating the integrity of the message by verifying the hash value sent. An example of a hash function that is used for this purpose is keyed-Hash Message Authentication Code (HMAC) [27] which uses a hash function along a secret key. Additionally, hash functions are used in *password protection* where the user's password is hashed and stored instead of the real value. Once the user login to the system, a hash is computed and compared with the digest saved in the database. if the database was exposed for any reason the password protections depends on how strong and secure this hash function is. Furthermore, hash functions can be also employed in generating pseudorandom sequences or to derive new keys given a single secret information as a seed. Hash functions can also be used as an efficient and fast method for records' lookup. They can serve as a mean to identify records in a database, and detect any changes that might have taken place [190] [154].

The mapping of the hash function is always many-to-one as the size of the input's domain arbitrarily larger than the output's fixed range. Thus, there is always the possibility that two messages will obtain the same hash value which is referred to formally as a *collision.* In sound cryptographic hash functions this possibility should not be obtainable.

Cryptographic hash functions have essential desired cryptographic properties and requirements that should be maintained. The security level of these properties is usually obtained in relation to the length of the hash value produced [78] [169]. This is explained below:

- **Collision Resistance:** A hash function H is collision resistant, if it is computationally infeasible to find $x = x^{'} \in X$ such that $H(x) = H(x^{'}) \in D$. It is not

possible to obtain two distinct messages that have similar hash value. For a hash function with n-bit size hash value to be resistance to collisions it should survive the attempts to find collisions for at least $2^{\frac{n}{2}}$ hash evaluations. An alternative reference to this property is a strong collision resistance. There are weaker forms of collisions as in *near-collision* in which collisions are found only with respect to parts of the hash values. In addition to *semi-free-start collision* where the adversary will change the Initial Value (IV) that is being used in the hashing algorithm for the two messages from the one originally indicated. As well as, *free-start-collision or pseudo-collision* which is a collision obtained when the the IVs used to generate two hash values are different.

- **Preimage Resistance:** A hash function H is preimage resistant, if it is computationally infeasible given $d \in D$ to obtain $x \in X$ such that $H(x) = d$. In other words, it is not possible to obtain a message associated with a specific given hash value, or given the output of a hash the input can not be obtained. This requirement translates the one-way property of a cryptographic hash function. For a hash function with n-bit size hash value to be resistance to preimage attacks it should survive the attempts to find preimages for at least $2^{\frac{n}{2}}$ hash evaluations.

- **Second Preimage Resistance:** A hash function is resistant to second preimage if given $x_1 \in X$ and its associated hash value $H(x_1) = d_1$ it should be impossible to obtain $x_2 \in X$ such that $x_2 \neq x_1$ and $H(x_1) = H(x_2)$. In other words, it is infeasible given a distinct message and its hash value to find a different message with the same hash value. For a hash function with n-bit size hash value to be resistance to second preimage attacks it should survive attempts to find second preimages for at least 2^n hash evaluations. An alternative reference to this property is a weak collision resistance.

These requirements form a hierarchy in relation to the security implied, if any of them were achieved, in a cryptographic hash function. It has been proven that a collision resistance implies second preimage resistance. Collision resistance implies preimage resistance under specific criteria investigated in [175] [187].

Based on the cryptographic requirements provided above, the class of MDC can be also divided into three main categories [200], [155], [156] and [154]. The first category is One-Way Hash Function (OWHF) which is a hash function that will follow Definition 1, and it is preimage and second preimage resistant. The second category is Collision Resistance Hash Function (CRHF) which is a hash function that will follow Definition 1, and it is collision, preimage and second preimage resistant. The third category is Universal One-Way Hash Function (UOWHF) [160]. It is a class of hash functions that is considered a weaker form of CRHF, and finding a second preimage is computationally infeasible. It is based on public parameter where a challenge input will be selected in the first phase. Then, a hash function will be selected from a family of hash functions, and a different input with the same hash as the challenge will be computed with negligible probability.

In addition to Definition 2, MACs are required to provide computation resistance which can be translated to a second preimage resistance. It is defined as the computation infeasibility to produce any input-MAC pair $(x, H_k(x))$ given zero or more pairs $(x_i, H_k(x_i))$ where $x \neq x_i$ and $h_k(x) = h_k(x_i)$. This property will ensure resistance of MACs to some possible forgery attacks.

There are several construction mechanisms for hash functions as in Merkle–Damgård construction method, and block cipher based methods(Matyas-Meyer-Oseas, Davies-Meyer, Miyaguchi-Preneel), sponge construction method and some other customized designs [169], [78], [93]. A recent and popular example on a sponge construction is SHAS−3 or Keccak hash function [31]. There are also some hash functions that are based on stream ciphers such as LUX [161], Shabal [55], and SHAMATA [20]. However, due to the difficulty of analysis, a proper mathematical security proof is not provided. Second preimage attack was provided on SHAMATA-512 by Kota Ideguchi and Dai Watanabe [109]. In addition to low-weight pseudo collision attacks on Shabal by Takanori Isobe and Taizo Shirai [111]. Moreover, free-start collisions were found on LUX-256 by Shuang Wu *et al.* [203].

As other primitives used to construct hash functions, hash functions are used to construct other primitives. They are used to construct block ciphers and stream ciphers as in BEAR, LION [19], Shacal [98] and SEAL, [154].

It is worth noting that there are a minimum level of cryptographic requirements when it comes to the the hashing algorithms to be used in the state-of-the-art that is constantly reviewed and monitored. These requirements when it comes to the parameters to be used in different hashing algorithms, and their sizes are discussed in [85]. It is currently stated that for OWHF the hash value should be larger than or equal 80 bits. As for CRHF the hash value should be larger than or equal 160 bits. Finally, for a MAC, it is required to have at least 64 bits size and 80 bits size for these hash functions to be used in most application environment [154]. Hash functions are not within the direct scope of this thesis. The reader can refer to the provided references for further details on hash functions.

The analysis given in this thesis does not cover the scope of hash functions. Further investigation on the state-of-the-art on hash functions is left to the reader.

2.2 Public-Key Ciphers

Public key ciphers are the class of basic algorithms that are mainly used for encryption, digital signatures and key agreement protocols to provide services of confidentiality, authentication and non-repudiation. They are also referred to as *asymmetric-key ciphers*. This is due to the fact that unlike the case of symmetric ciphers, they rely on two profiles of keys to perform a cryptographic action. The first is known as a *public* key which is shared with anyone who is interested in carrying an encrypted communicating with

the owner of the keys. The second is a *secret or private* key that is known only to the owner of the keys. It is securely stored to avoid being compromised by any unauthorized party. They are usually mathematically connected. However, the algorithm is designed to ensure that given a public key it is computationally infeasible to obtain the private key.

Definition 3. *A public-key cipher is a set of encryption and decryption transformations* $C = Enc_e(M)$ *and* $M = Dec_d(C)$ *based on a pair of keys* $(e, d) \in K$*. A public key* e *is used for encryption by any one who has it. The private key* d *is used for decryption of a message only by the owner of the private key. It should be computationally infeasible to obtain* d *from* e*.*

The key motivations behind the initiation of public-key cryptography are first the need to provide a secure mean to share secret keys before using them in symmetric communication. An attacker can possibly intercept the secret key, or brute force the key space if the key size was short and known. Secondly, the problem of the scalability of the symmetric system as in how can it support a huge number keys exchanged. For example, if n people are in need to securely communicate with each other, then $\frac{n(n-1)}{2}$ keys will be needed for all parties to communicate securely with each other. This can cause a problem if n is large. As we have two keys used in public-key ciphers there is no need to agree on different keys with n parties for secret communication. They can all use the public key of the receiver. In this sense, only n key pairs are needed for n communicating parties.

The idea of public-key ciphers was traced back to the late 1800's when factorization problem was considered as an application of one-way functions, and to construct trapdoor functions [112]. In 1970's, the idea and the theory of non secret encryption was proposed by different cryptographers in the British Government Communications Headquarters (GCHQ). They proposed algorithms for public-key encryptions and key exchange, yet these efforts were not made public until 1997. Whitfield Diffie and Martin Hellman in 1976 published fundamental approaches for public-key cryptosystem, and key agreement or exchange protocol [200]. Moreover, RSA cryptosystem is considered the first practical public-key cryptosystem published in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman [171].

These public-key ciphers are usually based on one-way trapdoor functions were the plaintext will be easily mapped to a ciphertext using the public key e. However, the inverse operation (decryption) is computationally infeasible unless a private key d is used which means an adversary can not achieve this goal. This computation infeasibility is based on the use of a set of mathematical problems that belong to a complexity class that can not be solved in polynomial time that is Non-deterministic Polynomial time problems or NP-problem. Clear examples are integer factorization to prime factors and Discrete Logarithms Problems that are used in RSA public-key cryptosystem, ElGamal Signature scheme and Diffie-Hellman Key establishment. In RSA, the security of the system is obtained by assuming the hardness of what is known as RSA problem.

Briefly, it states that an attacker can not produce the ciphertext $C = P^e mod n$ given the plaintext P. The attacker has knowledge of (e, n) which is the public known key. The algorithm states that $n = pq$ such that p and q are distinct primes, $2 < e < n$, $0 \leq C < n$, and e and $\Phi(n)$ are relatively prime to each other. It is stated that RSA problem is related but not equivalent to the factoring problem. It is usually assumed that the existence of an efficient factoring algorithm will solve the RSA problem and break RSA encryption. Nevertheless, an efficient algorithm for breaking RSA does not mean that an efficient factoring algorithm exists which is considered an open problem.

In addition to the application of public-key ciphers in encryption they are also used with hash functions to sign messages with the private key of the user. In this since signatures can provide non-repudiation, for an owner of the private key can not deny his signature of a message. The verification process is achieved usually through the use of the public key associated with that private key by the receiver.

In comparison with symmetric ciphers, public-key ciphers need larger key sizes than symmetric keys, yet this is not a direct indicator that they are necessarily stronger. The computation effort for an attacker marginally differs depending on the cipher structure under study. An 80 bits key in symmetric-key ciphers will mean that there is 2^{80} possibilities to brute force. In public key ciphers it will depend on the efficiency of the factorization algorithm used. For example, for 512 bits RSA key were factored in almost six months using General Number Field Sieve algorithm (GNFS).

Since public keys tend to be slower than symmetric keys, they are used in combination with symmetric-key ciphers. They are mainly used to encrypt session secret keys while symmetric-key ciphers are used to provide encryption for large volume of data. Still public-key ciphers ensures efficient signature and key management approaches. They play a vital role in PKIs to provide a framework for secure communication that provides encryption and authentication for entities and data among other services. Such framework is used in banking applications, and public identity and social security systems.

Nowadays, public-key cipher present a wide spectrum of construction methods and designs that aim to improve the efficiency and the security of the different suggested constructions in comparisons to what is available in the-state-of-the-art. This includes the use of elliptic curves, code-based cryptography and lattices among others.

As the analysis given in this thesis does not cover the scope of public-key ciphers. Further investigation on the state-of-the-art on public-key ciphers is left to the reader.

2.3 Symmetric-Key Ciphers

Since symmetric-key cryptography depends on establishing a cryptographic function based on a shared secret key. Then symmetric ciphers are the the set of algorithms that will provide encryption and decryption functionalities between two parties based on a shared secret key. They are mainly utilized to provide confidentiality and authentication.

This will introduce certain requirements such as the existence of a key establishment or agreement scheme that will enable communicating parties to negotiate and agree on a secret communication key before the communication occurs. This is a direct implementation to Kerckhoffs's Principle as long as the key is secret the primitive is relatively secure and can only be operated by authorized parties. The main two forms of symmetric-key ciphers that are deployed in different communication protocols are *block ciphers* and stream ciphers. Both structures have different security requirement as they are designed differently, yet they also share common requirements. The next sections will provide a general overview about their design structures.

2.3.1 Stream Ciphers

Stream ciphers were recognized in the early symmetric cryptosystem introduced in 1882 by Frank Miller to secure telegraphy [28]. That cryptosystem were reinvented later by Gilbert Vernam and Joseph Mauborgne in early 1900's as Vernam ciphers [194]. The ciphertext C will be produced using a plaintext message P and a "random" or pseudorandom secret key K of the same size as the message as shown below

$$C = P \oplus K$$

where $\oplus$ indicates the operation of bitwise xor between the bits in the plaintext and the key. The receiver will use the xor operation on the same key and given ciphertext to obtain the plaintext message. Initially Vernam was structured to have the key read from a tape which is on a loop to accommodate the size of the message. Consequently, it meant that there is the possibility of redundancy in the key segments used. Mauborgne suggested to use the key only once to ensure the randomness in the key stream used for encryption. This design structure of stream ciphers is referred to as *one-time pad (OTP)*. OTP was declared by Claude Shannon in 1949 to be information-theoretically secure or to have *perfect secrecy* where an enemy cryptanalyst with unlimited computational power will not gain any additional information about the plaintext from the ciphertext except the length of the message. The essential focus in this cryptosystem is to use a true random key only once, which means a ciphertext can be decrypted to any plaintext of the same length, and They all will be likely equal in a case of a brute force attack on the key. The probability of the plaintext equals the probability of the plaintext given the ciphertext. This can be translated to

$$H(M) = H(M|C)$$

where $H(M)$ is the entropy function for the plaintext and $H(M|C)$ is the conditional entropy of the plaintext given the ciphertext. The practical implementation of OTP is hard to achieve as it requires a perfect, true random and one-time keys. In addition to the consideration of secure transmission and management of these long keys material where keys should be as long as the message itself.

Regardless of the fact that OTP is hard to obtain in real life implementations, current modern designs of stream ciphers are aiming to practically approximate its benefit. This is mainly achieved through having an initialization phase where k-bit key is used as an input to a keystream generation algorithm. It is also seeded a random Initialization Value (IV) in order to finally generate a long unique pseudorandom keystream sequences that are used as an OTP, thus xored to plaintext to output the ciphertext. The role of a public IV is important to maintain the freshness of the keystream if the same secret key is used for multiple encryptions. The current constructions of stream ciphers designs are using *Linear Feedback Shift Registers (LFSR)* and *Non-Linear Feedback Shift Registers (NFSR)* in the keystream generators as Pseudo-Random Number Generators (PRNGs). LFSRs are used to generate a deterministic long period sequences with good statistical properties (uniformly distributed), and suitable for hardware environment [92], [145]. Nevertheless, the output sequences are easily predictable due to the linearity of the system.

To provide a general overview, an LFSR of length L, has L stages s_0 to s_{L-1}, and a feedback or connection polynomial of certain properties,

$$C(X) = \sum_{i=0}^{i=L} c_i X^i$$

Where $c_i \in \mathbb{F}_2$, being either 0 or 1, and each stage can process one bit at a time. An LFSR of length L will have a maximal sequence of length $2^L - 1$ *iff* its connection polynomial is primitive. The output sequence obtained will be updated based on a clock synchronization, where the output of stage s_0 will belong to the output sequence. Stage s_{i-1} is updated from s_i for all stages, and stage s_{L-1} is updated by s_n that is referred to as the feedback bit. It is updated using stages $s_0.....s_{L-1}$ based on connecting polynomial coefficients used,

$$s_n = \bigoplus_{i=0}^{L-1} c_i s_{n-i}$$

Berlekamp-Massey algorithm for determining a linear finite binary sequence can be used to determine the coefficients of the connection polynomial used from any subsequence of length at least $2L$ with a linear complexity of $O(L^2)$ [154]. This result can be used in attack models called *known or chosen-plaintext attacks* to obtain the keystream bits.

Additional components have been introduced to the design of LFSR to obtain better nonlinearity properties. As in introducing nonlinear combinations either in bits from the LFSR state itself or of the output bits of number of different LFSRs (nonlinear boolean functions). In addition to using the output of of several LFSRs to regulate the clock of certain LFSR. Furthermore, NFSRs are another design component for stream ciphers that is similar to LFSR but based on a feedback function that is nonlinear. A special case of NFSR is Feedback with Carry Shift Register (FCSR) which their innate nonlinearity is credited to the use of integer addition with carry instead of an xor with regards to the different stages in the shift register and the content of an additional memory for storing

integer carry. Stream ciphers are usually divided into synchronous stream ciphers and asynchronous stream ciphers. In synchronous stream ciphers the keystream generated is independent of the plaintext and the ciphertext and only depends on the key and the IV. While in asynchronous stream ciphers, the next state of the keystream depends on the previous one and the ciphertexts.

Stream ciphers are mainly used in real time communication applications where transmission errors are high because they do not introduce propagated errors, in addition to the environments where equipment have limited buffering or memory available. Examples for stream ciphers include RC4 that is used in SSL and Wired Equivalent Privacy (WEP), E0 that is used in Bluetooth protocol [105], SNOW [84], SCREAM [97], SEAL, the lightweight ciphers Grain [100] and Trivium [61].

2.3.2 Block Ciphers

A block cipher is a symmetric structure that takes as an input a plaintext and divides it into blocks of certain size (for example n bits usually ≤ 64) to apply the encryption on each block using a secret key of certain size (for example k bits). This key is usually assumed to be randomly chosen and the output of the encryption to look like a random structure. The final ciphertext of the whole plaintext is a concatenation of all these n sized ciphertetxts, if we assumed that a block cipher maps input of size n bits to output of size n bits. The encryption operation must be bijective or invertible using the same secret key in order to be able to perform the decryption operation by authorized users. The key space is assumed to be 2^k with general effective key length of k bits if all bits are actively used in the encryption of the bits.

Definition 4. *A block cipher is an invertible mapping which has an input and output of block size n bits and key size of k bits.This mapping is characterised by $E : \mathcal{MK} \rightarrow C$, where $\mathcal{M} \in E_2^n$ is the message space, $\mathcal{C} \in E_2^n$ is the ciphertext space, and $\mathcal{K} \in E_2^k$ is the key space. The inverse mapping is donated by the decryption function that $E^{-1} : \mathcal{CK} \rightarrow M$*

An *ideal cipher* is usually described as a *random cipher*. This definition entails that this cipher be a permutation from n-bit message space to n-bit ciphertext space where there will be $(2^n!)$ permutations on 2^n possibilities. Each secret key from $2^n!$ keys will select one permutation. In order for such model to represent all possible permutations and to make it possible to select a total random permutation from the set of permutations of n-bit size. Then, the key size should be $\log_2(2^n!) \approx (n - 1.44)2^n$ bits which is too big for practical terms. Consequently, using an ideal cipher model would not be achievable in practical sense. This will direct the focus towards an achievable cipher design which resolves to using an encryption function with a key that is selected at random. It would be an invertible function (permutation) uniformly chosen at random from 2^k permutations. In short, a block cipher is considered to be a family of 2^k n-bit permutations selected uniformly at random out of $(2^n!)$ n-bit permutations.

As we have discussed in the previous chapter, Shannon has introduced many principles and requirements of modern block ciphers design [180]. Prominent design principles he discussed include product ciphers, iterative structures of block ciphers, and using multiple transformations (substitutions and permutations) to achieve confusion and diffusion properties.

Modern designs of block ciphers have been enhanced to include strategies that will maximize the random appearance of the output of encryption process.

Almost all block ciphers designs today are based on *iterated block ciphers* which are based on iterating a round function of specific structure for a number of times. Each iteration is referred to as a *round*. The round function will be composed of key mixing phase where round subkeys will be mixed with the encrypted data. In addition to a linear layer of permutations to add diffusion phase where a change in the input bit will influence all output bits. In other words, dissipating the bits in the message such that any redundancy, or statistical structure in the plaintext, is dissipated over a long range in the ciphertext. Having this property a change in the plaintext/secret key bits will influence all (as many as possible) bits of the ciphertexts. The final component of iterated round is a non-linear layers of substitutions to add confusion phase that is meant to make the relationship between the key and ciphertext as unrecognisable as possible. Alternatively, to make it harder to recover the key in the case of obtaining large amount of plaintext/ciphertext. These properties are provided through using different design components that vary from one cipher structure to another.

Definition 5. *An iterated block cipher generate the ciphertext C through applying $C = E_K = R_{k_r}^{(r)} \circ R_{k_{r-1}}^{(r-1)} ..... \circ R_{k_1}^{(1)}$ where $r > 0$ integer value, $K \in \mathcal{K}$, and $k_1, k_2, ..., k_r$ are round keys generated from K through key scheduling algorithm.*

The round keys are derived from a secret key through a deterministic algorithm called key scheduling. Some of the specific cases of round structures are Feistel scheme, Substitution-Permutation Networks (SPN), and Lai-Massey scheme. The different schemes that are used to construct each round can be briefly described as follows:

2.3.2.1 Feistel Scheme

In this scheme, a permutation on $2n$-bit output based on an internal round function F that operates on n-bit will be built and might resemble a product cipher where substitution boxes are part of its construction. Note that F does not have to be invertible to allow inversion of the Feistel cipher. Basically, the state at round i is split into halves (L_i, R_i) of n-bit length each. Then the round function F will take as an input one of the halves (for example R_i) and a round key k_i. The output of this round function is then xored with the other half (for example L_i). The output of the single round will be $R_{i+1} = L_i \oplus F(R_i, K_i)$ and $L_{i+1} = R_i$ as the two original halves of the round input are swapped at the output [121]. There is an exception for this case, as

usually in the last round the output is not swapped. This description resembles a classic balanced Feistel network, yet there exists as well different types of Feistel networks under the term generalized Feistel networks where they use more than two branches and different operations. They are also commonly used in block cipher design structure [107]. Feistel schemes started as an appealing design choice for ciphers intended for hardware environment as different implementation for encryption and decryption processes will not be needed. Decryption is obtained with the same r-round encryption process, yet using the subkeys in reverse order.

Practical examples on block cipher that are based on Feistel scheme are DES, Blowfish, Camellia,CAST-256 and SIMON. Cryptanalytic properties of some of these ciphers will be explored in the next chapters.

2.3.2.2 Substitution Permutation Network (SPN)

This cipher design structure or scheme is the most accurate modern translation to Shannon cipher design principles (as in product ciphers). SPN consists of series of key mixing, confusion layer and diffusion layer applied repetitively to achieve certain level of security. It subjects the full state to a non-linear layer of substitutions, linear layer of permutations, and key mixing layer.

The substitution and non-linear layer is composed of set of *substitution boxes* or *S-boxes* that have certain properties to ensure that they provide a level of resistance to cryptanalytic attacks especially differential and linear cryptanalysis. Hence, they are considered sound to be used within the design as they do not introduce exploitable structures [65] [131].

S-boxes can basically be viewed as vectorial boolean functions that map a vector of small size m to another of small size n i.e $F : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ where sometime m equals to n usually both are ≤ 8 bits. This can be described by a vector $(f_0, ..., f_{n-1})$ where f_i for $0 \leq i < n$ are boolean functions from $\mathbb{F}_2^m \rightarrow \mathbb{F}_2$ that are usually referred to as *coordinate, component or output functions* of the S-box.

These S-boxes can have certain cryptographic criteria as, but not limited to, bijection, completeness, high algebraic degree,non-linearity, balancedness and Strict Avalanche Criteria (SAC). The detailed analysis of these properties is out of the scope of this thesis, yet a brief introduction of some of them will follow. Bijection can be defined as one-to-one and onto mappings between the domain to the range of the S-box function that will ensure the presence of an inverse to the S-box. Moreover, completeness of S-boxes refers to the fact that every output bit will rely on all input bits using a simple boolean expression for each output bit that uses all input bits. This can be alternatively expressed as $\sum_{x \in \{0,1\}^m} F(x) \oplus F(x \oplus d_i^{(m)})$. It donate that for F to be complete, then there is at least one pair of plaintext that differs in one bit i such that $d_i^{(m)}$ is vector of size m of *hamming weight* one i.e $wt(d_i^{(m)}) = 1$,and $1 \leq i \leq m$. The hamming

weight refers to the number of bits with value one in the vector. The resulting S-box transformation differs at least in one bit j for all possible bits.

Furthermore, the Algebraic Degree (AD) of an S-box refers to the maximum algebraic degree of the component functions of the S-box i.e $AD(\text{S-box}) = \max \deg((f_0, ..., f_{n-1}))$. The degree of f_i is the maximum number of variables of the terms used to describe $f(x) = \sum_{i \in \mathbb{F}_2^m} c_i x^i$ where $c_i \in \mathbb{F}_2$. The higher this property is the better it is, since this property as well indicates that at least one of the output bits relies on many input bits.

In addition, the Non-Linearity ($NL(f)$) of an S-box measures the hamming distance between the S-box function and the set of linear combinations or affine functions or how often the function will satisfy an affine property. It indicates that there is no linear mapping from input to output. This will ensure certain resilience to linear and differential cryptanalysis. Walsh-Hadamard is usually used to indicate non-linearity of a function i.e $\hat{f}(a) = \sum_{x \in \mathbb{F}_2^m} (-1)^{f(x)+a(x)}$ where $a(x)$ is a boolean linear function. The higher $\hat{f}(a)$ the closer f to the linear function $a(x)$. Thus, non linearity of f can be measured as $NL(f) = 2^{m-1} - \frac{1}{2} max|\hat{f}(a)|$ which donates the hamming distance between f and the set of affine linear functions. The higher non linearity the harder it is to get a linear relation to describe the S-box. Alternatively, non-linearity can be also expressed as the minimum algebraic degree of all affine linear combinations of the components functions of the S-box. The relationship between algebraic degree and non-linearity can be expressed as $NL(f) \leq AD(f))$.

On the other hand, avalanche effect can be defined as $\sum_{x \in \{0,1\}^m} wt(F(x) \oplus F(x \oplus d_i^{(m)})) = n2^{m-1}$. It indicates that an average of one half of the output bits will be flipped if any input bit was changed. Furthermore, strict avalanche criteria or SAC is the property that combines completeness and avalanche effect. It refers to the effect of changing one bit in the input will change every output bit with probability $\frac{1}{2}$. This will indicate that the S-box function will satisfies $\sum_{x \in \{0,1\}^m} F(x) \oplus F(x \oplus d_i^{(m)}) = (2^{m-1}, 2^{m-1}, 2^{m-1}, ..., 2^{m-1})$ which means that$F(x) \oplus F(x \oplus d_i^{(m)})$ is balanced. Balancedness means that the boolean functions defining the S-box will have the same number of 0's and 1's [125] [79].

A trade-off between these properties is important, since obtaining the maximum potential of each of them at once is difficult as they might conflict with each other.

It should be noted that S-boxes can be constructed using different approaches. A potential first approach is to use certain mathematical functions of certain properties as in the power function over Galois field as it is the case with AES S-box where $f(x) = x^{-1} = x^{254}$. The second alternative design strategy for S-boxes is to choose an S-box at random from a family of cryptographic boolean functions and test its properties against S-box criteria for specific design. Finally, S-boxes can also be generated through algebraic construction of certain functions until an S-box with intended properties is found.

The permutation layer of SPN applies a linear transformation to the output of substitution layer (S-boxes). It diffuses the output of each S-box such that it becomes an input

to the maximum number of S-boxes in the next round. This can be achieved either through ad-hoc selection of this layer as in Serpent [33] and SHA-3 [31] or through following *wide-trail strategy*proposed by Daemen and Rijmen in the design of SHARK [170], SQUARE [73]and AES [72] [75].

Wide-trail design strategy aims to bound the probability of a differential and linear characteristics or trails that spans few number of rounds in the cipher. This achieved through having nonlinear (S-box), linear (permutation) and key mixing components that are wisely selected to maximize the benefits of non-linear operations by maximizing the diffusion property. Thus, making it harder for differential and linear cryptanalysis methods to achieve practical results within the proposed design specifically differential and linear characteristics of low hamming weights and possibly high probability. Diffusion is measured in terms of *branch numbers* which indicates the minimum number of active S-boxes over certain number of rounds. Active S-boxes are S-boxes which are involved in the linear or differential characteristic or trail especially after applying the linear layer. For example, S-boxes with non-zero input difference for differential cryptanalysis is active while S-boxes with non-zero output mask in linear cryptanalysis is active. Increasing the number of active S-boxes in these trails over a certain number of rounds will lower the probabilities of differential and linear characteristics. Thus, making it harder for the cryptanalyst to utilize them in a successful practical attack. Linear layers sometimes use bitwise permutations or families of linear codes that are chosen to provide specific properties (i.e certain minimum distance) in order to provide the maximum number of active S-boxes possible. An example of these codes are Maximum Distance Separable Codes (MDS). Wide-trail design strategy is meant to provide a connection between the number of active S-boxes (in differential and linear cryptanalysis approaches) and the minimum distance of these families. It provides simple and solid security analysis when it comes to the effect of the linear layer on the overall structure. However, the quest to design a perfect and efficient linear layer is still a work in progress to provide a suitable balance between the security margin provided and the efficiency of the proposed design.

It is worth noting that, SPN has been also used in the construction of other cryptographic permutations that were used in stream ciphers and hash functions as in sponge-based hash functions among others as in Keccak or SHA-3 and other hash functions construction as in Whirlpool [75], [31], [25].

2.3.2.3 Lai-Massey Scheme

This scheme was proposed by Xuejia Lai and James Massey when designing International Data Encryption Algorithm (IDEA) [138]. As with Feistel scheme, rounds functions that are based on Lai-Massey scheme are based on constructing a permutation from functions. There are two types of round functions, F that is considered a full round function, and H that is considered a half round function. The scheme is based on commutative and associative law, and mixes operations from different alge-

braic groups. For example in IDEA, the scheme uses addition modulus 2 (alternatively xor $\oplus$) , addition modulus 2^{16} ($\boxplus$) and multiplication modulus $2^{16}+1$ ($\odot$). Simply, the F function takes the subtraction of both branches of the scheme, and adds its output to both branches. The scheme is not consider secure enough even if the round functions are. This is due to a simple distinguishing attack for any number of rounds that use the following possible relation,

$$X_{left} \boxminus X_{right} = Y_{left} \boxminus Y_{right}$$

where $\boxminus$ is a difference or subtraction operation, the input of the scheme is $X = X_{left}||X_{right}$, and output of the scheme is $Y = Y_{left}||Y_{right}$. In [191], Vaudenay investigated this property and proposed a solution to eliminate it. A fixed permutation σ will be used at the output of each round. This permutation should be of a property that $x \rightarrow \sigma(x) - x$ is also a permutation which is called orthomorphism for addition law. Additional examples on ciphers that are based on Lai-Massey scheme other than IDEA which was used in Pretty Good Privacy standard (PGP) are FOX and Walnut [118].

2.3.2.4 Addition-Rotation-XOR (ARX) Scheme

This scheme of block cipher design is dated back to the design of FEAL [181] as the term initiated as AXR by Weinmann [199]. It depends on specific set of basic operations to build up the non-linear, linear and key mixing components of the cipher structure. They basically rely on modular addition ($\boxplus$), bitwise XOR ($\oplus$) and rotations between the different words or bits of the cipher ($\ll$) or ($\gg$). Addition will be the key component that provides diffusion and non-linearity. Because it is based on a bitwise operation the diffusion is relatively slow. The rotations is meant to speed up the diffusion and provide balanced mixing between left and right bits. However, the cost of these operations in general is relatively low in hardware and software which means that designs with high number of operations are fast and efficient. It is believed that this combination of operations gives a secure primitive given a good number of rounds sue to their resistance to standard linear and differential cryptanalysis [152], [36]. Since basic operations run in constant time, the design provides certain resistance to timing attacks. However, differential analysis is still applicable through rotational cryptanalysis approaches. It should be noted that for certain layers designers tend to use $\oplus$ rather than $\boxplus$ because it is easier to analyse when it comes to differential cryptanalysis, cheaper and faster in hardware architecture.

Examples of cryptographic primitives that are based on ARX constructions are the block ciphers Threefish [87], SIMON and SPECK, stream ciphers Salsa20 [29] and ChaCha [30] and hash functions BLAKE [102] and Skein [87].

It is worth noting that a block cipher structure can mix different components of the presented design schemes.

2.4 Lightweight Cryptographic Primitives

Lightweight cryptography has become an active field of study in recent years. Most of the current proposed designs of cryptographic primitives tends to take into consideration lightweight design metrics when certain cryptographic structure is proposed. The aim of such design is to provide efficient cryptographic primitives suitable for a minimalistic resource environment such as ubiquitous computing environments. A direct demanding example are Radio-Frequency IDentification devices that are being used in commerce, public health, transportation and many other domains. These environments require very low cost in hardware and software solutions that are going to withstand the different possible attacks models including physical or hardware attacks. An essential trade-off paradigm between security, performance and cost-effective designs are applied on the proposed cryptographic structures. Key length, number of rounds and the hardware architecture used are being constantly tailored to fit a better lightweight-security margin with an optimized trade-off. For example, pipelined architectures can be used to achieve a secure and high performance hardware implementations through using side-channel countermeasures. However, it imposes certain area requirements that will increase the cost of the design. Since chip area has become relatively inexpensive in some designs this level of trade-off is acceptable. To evaluate the efficiency of the implemented design for lightweight purposes certain metrics are used which include the following:

- **Power Consumption.** This metric is measured using detailed analysis of the different circuit components usually through a certain simulation. the estimations are carried out on the gate level typically in micro Watts. Place and route steps are used to provide accurate power readings on the transistor level.
- **Current.** This metric indicates the individual voltage consumed by the different cell standard library components.
- **Area.** This value is highly dependent on the fabrication technology (i.e CMOS) and the dedicated cell library used in the associated technology. This standard library contains many logic gates as in NAND gates. In order to consider area requirements independently from the complexity of digital electronic circuits and its manufacturing technology the concept of *Gate Equivalence (GE)* is used. The area of GE is measured by dividing the area in μm^2 by the area of a two-input NAND gate (which is one GE)
- **Throughput.** This metric indicates the rate of output units produced per certain time metric. As in the number of bits over certain time period usually expressed as *bps*
- **Cycles.** This metric indicates the number of clock cycles needed to perform a computation or execution of an instruction.

- **Time.** This metric indicates the needed time for an operation. It is computed by dividing the number of cycles over the frequency.
- **Energy.** This metric indicates the energy consumption or the power consumption over certain time lapse. For efficient design it might be worth noting the consumed energy for each bit of an output. This metric is measured in micro Jouls (μJ).
- **Efficiency.** This metric indicates the hardware efficiency by calculating ratio between the area and the throughput $\frac{\text{Area}}{\text{Throughput}}$. It is measured in $\frac{\text{GE}}{\text{bps}}$

In some cases, instruction sets are modified by microprocessors manufacturers to accommodate faster implementations of encryptions and decryption operations as in AES New Instruction (AES-NI). It improved the throughput from 28 cycles per byte to 3.5 cycles per byte. An efficient design should take into consideration a reasonable compromise between the security margins of the primitives in the given environment and the efficiency metrics presented. However, most of the current designs proposed are mainly tailored to fit better chip area constraints given a certain level of security. This might not be an optimal choice as execution time might be scarified to produce light but slow primitive in certain environments. A proper line of study would be to provide lightweight designs that are tailored to take into accounts different metrics for an optimal secure performance on the intended target environment.

Lightweight cryptographic designs might follow the proposed design schemes in the previous sections with consideration to move them toward the efficient boundaries of the implementation in hardware and software environments. There are a surge of lightweight designs for different cryptographic primitives. For examples, lightweight hash functions include Quark [22] and Photon [94]. While examples on block ciphers include PRESENT [**?**], PRINCE [52], KLEIN [91], mCrypton [146], SIMON and SPECK. In addition to stream ciphers that include Trivium [61] and Grain [100].

2.5 Cryptanalysis

As stated in the previous chapter, *Cryptanalysis* is the main instrument used to evaluate, verify and test the designed cryptographic primitives and push them through all possible claimed or non-claimed security margins. This section will explore the general definition of cryptanalytic attack on block cipher designs, different attack models, their goals, associated complexities to these attacks, and certain examples on generic attacks. Generally, attacks on block ciphers are divided to generic attacks that are independent of the structure of the target design, or cryptanalytic methods that are dedicated to certain design or structural weakness of certain component in the construction. An *attack* on a block cipher is an algorithm that exploits a certain weakness in the security requirements of the cipher design. The exploitation should be non trivial and it should be less than exhaustive if we would like to indicate an attack. The complexity of the attack is evaluated mostly in terms of the following factors:

- **Data Complexity**: This refers to the expected data needed either off-line or on-line to perform an attack successfully. It is usually reflected in the number of plaintext/ciphertext pairs needed to amount an attack.
- **Time Complexity**: This refers to the time needed to perform an attack successfully. It is usually reflected in the number of evaluations, operation or encryptions needed to have a successful attack.
- **Memory Complexity**: This refers to the amount of memory or storage needed to have a successful attack. It is reflected usually by the size of memory units (for example bits) of the output or intermediate values needed to be stored for an attack to succeed.

Typically, *the success probability of the attack* is dependent on these factors. The attack is considered to be more practical if the resources consumed for it to be successful are low.

The success probability for an attack can be evaluated using empirical observations of probabilistic results on specific attack as in signal-to-noise ratio (S/N) in differential cryptanalysis [37] [131],

$$\frac{S}{N} = \frac{P_{accept}}{P_{reject}} \simeq \frac{(2^k - l)p}{\gamma\delta - p}.$$

Where P_{accept} is the probability that the correct key will be among the candidate keys obtained by an instance of attack. Moreover, P_{reject} indicates that the probability of finding the incorrect key value among the candidate keys obtained by an instance of an attack. Furthermore, k refers to the size of the key to be recovered, γ refers to the probability that randomly chosen pairs will survive the filtration of candidate keys using the generated pairs, p probability that a statistical relation (for example a differential) holds, and δ refers to the average number of keys associated with the correct pairs that survived the filtration.

In terms of a general definition of success, consider an attack on a key of size k gets the right candidate among the 2^k potential keys. Then the bit advantage obtained over an exhaustive search is $k - log(k_c)$ where k_c is the number of key candidates considered until the right key obtained. It is said that if the right key was the first candidate, then the bit advantage obtained is k for k-bit key.

A general analysis of the calculations of success probabilities to different cryptanalysis methods as in differential and linear cryptanalysis was presented by Selcuk *et al.* and and many others in [178] [44].

2.5.1 Goals of a Cryptanalyst

For an exploitation on a cryptosystem to be qualified as an attack, it should provide a potential practical feasibility less than brute force. It should be clarified that the

general goal of a cryptanalyst or an attacker is to recover the secret key of the cryptosystem. However, in some cryptosystems this is not always achievable which introduce an alternative taxonomy of attacks based on the goals, achieved results and obtained information by an attacker [129], [131], [90]. They can be listed as the following ordered from the strongest to the weakest:

- **Total Break:** The attacker achieves the goal of retrieving the key of the user or the secret key used in the cryptosystem. It is alternatively referred to as *key recovery attacks*. This type of attack might need a high data complexity as in a large number of plaintext/ciphertext pairs. Brute force is considered a possible type of key recovery attacks. If a few pairs of plaintext/ciphertext was given along with the size of the key for certain block cipher. Then an attacker can guess all possible keys used for to generate one pair and test the candidate key on the other pairs. It is believed that the right key will be found after trying 2^{k-1} keys where k is the size of the key. The previous value will be considered as a benchmark to indicate whether an attack is qualified as a key recovery attack if it ran faster than brute force.

- **Global Deduction:** The attacker will be able to obtain an equivalent algorithm for encryption or decryption without further knowledge on the key.

- **Local Deduction:** The attack will be able to generate the ciphertext to a given plaintext, or plaintext to certain ciphertext. This can translate to state recovery in stream ciphers where an internal state can be recovered given partial keystream and additional public information.

- **Distinguishing Algorithm:** The attacker has access to a black box of the cryptosystem. He/she can distinguish between block cipher using a randomly chosen secret key, and randomly selected permutation. For example, this can be achieved also through formal statistical hypothesis testing as Neyman-Pearson paradigm [117]. The main concept revolves around the fact that a cipher should exhibit a random behaviour. This is indicated by producing a ciphertext that can not be distinguished from uniformly random distribution of a source. Neyman-Pearson paradigm will be used to decide which of two given probability distributions on the basis of some samples or random variables generated by one of these distributions is the cipher or uniform random distribution.

  Practical distinguishing algorithms can be extended to key recovery attacks, if they can be used to extract non-trivial information from the cipher structure.

  In some cases, distinguishing algorithms include the possibility of detecting non-randomness using parts of the secret keys as potential input in addition to public parameters. This is considered as a weak form of distinguishing attacks and includes related key attacks [186].

2.5.2 Attack Models

There are different *attack models* that are commonly used when a cryptanalytic method is applied on a cryptographic primitive or more specifically a block cipher. These models are based on the level of knowledge (or accessibility to information) that the attacker have when attempting an attack on the cryptographic structure. The choice of an attack model depends on the complexity (data, time, memory) and success probability achieved by the model for specific cryptanalysis method. They can be classified as follows:

- **Ciphertext-Only Attack (COA) or Known Ciphertext Attack**: This model of attack has closer resemblance to a real life scenario. It assumes that the attacker has only an access to the ciphertext produced by the encryption algorithm. The attacker might deduce and rely on partial or full knowledge of the plaintext, as a result of some redundancy in the ciphertext using frequency analysis. A direct example on attack under this model is a brute force attack.

- **Known Plaintext Attack (KPA)**: This model of attack assumes that the attacker has knowledge of pairs of plaintext/ciphertext. A prominent cryptanalysis method that uses this attack model is linear cryptanalysis.

- **Chosen Plaintext Attack (CPA)**: This attack model assumes that the attacker has access to ciphertexts after requesting encryption of a selected set of plaintexts before launching an attack. A prominent cryptanalysis method that uses this attack model is differential cryptanalysis.

- **Chosen Ciphertext Attack (CCA)**: This attack model assumes that the attacker has access to plaintexts after requesting decryption of a selected set of ciphertexts before launching an attack. A prominent cryptanalysis method that uses this attack model is differential cryptanalysis on stream ciphers.

- **Adaptive Chosen Plaintext Attack (CPA2)**: This attack model assumes that the attacker as in CPA hash an interactive access to chosen set of plaintexts and their associated encryptions. However, in this model an attacker has access to encryption machine/algorithm for unlimited time. Then, an attacker can modify the selection of the plaintext to be encrypted based on his previous observations on plaintext/ciphertexts results used in his attack.

- **Adaptive Chosen Ciphertext Attack (CCA2)**: This attack model assumes that the attacker as in CCA hash an interactive access to chosen set of ciphertexts and their associated decryptions. However, in this model an attacker has access to decryption machine/algorithm for unlimited time. Then, an attacker can modify the selection of the ciphertext to be decrypted based on his previous observations on plaintext/ciphertexts results used in his attack.

- **Related-Key Attack**: In this attack model the attacker is assumed to have an access to plaintext and its different encryptions under different unknown related keys that have a certain known or chosen relation (for example differential one) with the target key to be recovered.

2.5.3 Security Models

In perspective, when designing and studying the security of a cryptographic primitives the efforts of the cryptanalyst/designer usually fall into the following contexts:

2.5.3.1 Unconditional or Perfect Security

Unconditional security indicates that a cryptographic system is secure regardless of the power of the attacker. It means that the attacker can have unlimited computational power, yet the cryptographic system will remain secure. Perfect secrecy falls under this category. Perfect secrecy indicates that the system is secure since the ciphertext produced by this system provides no information about the plaintext, unless the key is known. OTP is considered an example of perfect secrecy systems.

2.5.3.2 Provable Security

Provable Secrecy refers to the possibility of forming certain assumptions and proofs around the security properties and margins of the secrecy system. Cryptographic secrecy systems are based on a hard computational or mathematical problem (i.e integer factorization, DLP, and Modular roots) in which compromising the cryptographic secrecy system relates to solving the problem. The proofs are built around security goals and attack models to constitute a reduction for a polynomial time adversary. As discussed earlier public key ciphers as in RSA are examples on such model. RSA security is mapped to the difficulty of large integers factorization. Provable models are considered when cryptographic primitives are considered by a cryptographic scheme or protocol. These models are considered ideal and the attacker can only attack them in specific manner. Example of these models are Random oracles and ideal ciphers. The practicality of these proofs are subject to debate as they are relative to the provided assumptions, goals and models definitions.

2.5.3.3 Practical Security

Practical security of a system will be based on the verifiable, experimental observations made around the security margins in the cryptographic primitive. For example, symmetric ciphers are bounded in security to the results of an exhaustive search on the key space for a key or size k which is 2^{k-1} in the absence of other structural flaws in

the cipher that can be exploited by an attack. The designers approach this model by providing families of certain primitive structure of different rounds and key sizes. In the case of a practical break on a reduced rounds due to iterative constant cryptanalytic effort, the rest of the family will provide potential alternative with different level of security.

2.5.4 Generic Attacks

This section will introduce a selection of *generic attacks* that are commonly used for cryptographic primitives in general and specifically block ciphers. They are a class of attacks that are not dependent on the detailed structure of the primitive under study. This is because they treat the algorithm as a block box. The complexity (mostly the number of evaluations required for a successful result) of these attacks are mainly influenced by the parameters of the algorithm as in the size of the key. Usually the designers of cryptographic primitives are constrained to provide algorithm parameters that will not make these attacks feasible. As in choosing keys of specific length to make sure that the primitive sustain a certain security margin. In this section we will explore common examples of generic attacks as in *exhaustive search or brute force attack*, *table lookup attack*, *Time-Memory Trade-Off attack (TMTO)* and *Meet-In-the-Middle (MIM)*.

2.5.4.1 Exhaustive Search

Exhaustive search or brute force is considered the most intuitive way to approach the attack of a primitive or cipher design and it can be always mounted on a given structure except a design with perfect secrecy as in OTP.

If we consider COA on a block cipher where the encryption operation of a message $Enc_K(m)$ is performed using a secret key K of size of k bits. The attacker, knowing the key size, can search through all possible key space of 2^k keys until he/she can decrypt these ciphertexts to a relevant plaintext. A relevant plaintext can be defined as a clear distinction from a random data, and it is achieved when approximated to natural language given the ciphertext is longer than a unicity distance. If the attacker was able to find the key in the first half of the keys then the associated time complexity will be around 2^{k-l} evaluations (decryptions/encryptions). The worst case scenario happens if the key was in the second half of the key space. Then, the attacker will need 2^k evaluations (decryptions/encryptions) using one pair of plaintext/ciphertext. The attack will always return a candidate key when it comes to a given pair of plaintext/ciphertext. The likelihood of a certain cadidate key being the secret key to be recovered can be increased by verifying this key against more plaintext/ciphertext pairs. To determine if this candidate key is the correct secret key depends on the block and key lengths. If the size of the key k is larger than the block size n of the ciphertext, then we might obtain more than one candidate for the secret key to be recovered. This is noted due to the fact that if we encrypted a block of message of size n to ciphertext of size n.

Then, we obtain 2^n potential pairs with probability of 2^{-n} for each pair. The number of keys used for encryption here is 2^{k-n} keys which indicates that we need test $N = \lceil \frac{k}{n} \rceil$ plaintext/ciphertext pairs to obtain a unique key.

The average time complexity can be expressed as the following,

$$\sum_{i=1}^{2^k} i \cdot Pr[K_{secret} = K_{candidate_i}] = \sum_{i=1}^{2^k} \frac{i}{2^k} \approx 2^{k-1}$$

Noting that the secret key to be recovered is chosen at random for probability of $\frac{1}{2^k}$. Each $K_{candidate_i}$ needs i evaluations (encryption/decryption), so on average we need 2^{k-1} evaluations.

The designers of block ciphers take in consideration exhaustive search as a threshold when they consider their designs. They mostly aim to make sure that the size of the key used is large enough to survive key recovery search using exhaustive measures. However, the evolution of computational power has constantly pushed the limits of these key sizes. As we have earlier discussed when it comes to DES search, high performance computing can play a role in accelerating exhaustive search.

Such computation power will be considered when key length is decided. For an attacker with $Comp_{power}$ computation power, then the time taken for a an exhaustive search is a factor of $\frac{2^k}{Comp_{power}}$. Consequently,the recommended secret key length k is chosen such that the life time of the data under protection by this key is less than $\frac{2^k}{Comp_{power}}$ [110], [131], [12].

Table 2.1: Securiy offered by different key sizes, in the absence of further cryptographic weakness [131], [85]

Key Length -in bits-	Search Time	Status (2010)
40	2^{40}	Easy to break
64	2^{64}	Practical to break
80	2^{80}	Not very feasible
128	2^{128}	Very strong
256	2^{256}	Exceptionally strong

2.5.4.2 Table Look-Up

This is a variant attack of exhaustive search, and it is sometimes referred to as *dictionary attack*. It is composed of two main phases: precomputation off-line phase to build a table of plaintext/ciphertext pairs and on-line phases to look up in these tables for the correct decryption key.

In the precomputation phase, the attacker will generate all possible encryptions of a chosen plaintext. Then store the values in a table ordered by ciphertext value. This phase will need 2^k evaluations and 2^k memory units in the size of the ciphertext.

In the on-line phase, the attacker will intercept a ciphertext then lookup for a match in the table, and the associated key will be the candidate secret key. This phase time complexity will be based on the search algorithm used on a sorted list which is very low. This poses a possible trade-off in comparison to exhaustive search(no memory and 2^k evaluations) when it comes to memory and time complexities invested in the two attacks

2.5.4.3 Time-Memory Trade-off

Time-Memory Trade-off was first proposed by Hellman in 1980 [101]. The attack is divided as well to a precomputation phase and an on-line phase where the attacker will precompute tables, and store them in memory in order to use them for an attack to recover the secret key faster.

Assuming that a block cipher with key of size k and block size of n exist. The attack will be carried out as follows:

- **Precomputation Phase:**
  - The attacker starts with choosing randomly m initial choices for encryption keys and plaintext P.
  - These keys will be used in creating m chains each of successive t encryptions $ch_j(SP, EP)$ where $0 \leq j \leq m - l$.
  - Each chain will have a starting point $SP = k_j$ and endpoint $EP = E_{k_{t-1}}(P)$. The point computations can be alternatively explained using R function $R_{i+1} = Enc_{R_i}(P)$ where $R_0 = SP = K_0$, $0 \leq i \leq t - l$ and $EP = R_{t-1} = Enc_{R_{t-2}}(P)$. It should be noted that R_i function will map n-bit block cipher to k-bit key. The attack will proceed with an initial assumption that there is no overlap between the m chains.
  - All the chains will create mt matrix, each row represents $(t - l)$ encryptions. The total distinct key values represented in the matrix are mt. In order for the chains to represent the full key space $mt = 2^k$ which reflect the time complexity of the off-line phase.
  - The attacker will only save (SP_j, EP_j) for all m chains which will reduce the memory requirement to $2m$ instead of the matrix.
- **On-line Phase:**
  - The attacker intercept the ciphertext C and uses a successive encryption to generate a possible endpoint $EP' = K_i = E_{K_{i-1}}(P)$ where $K_0 = C$ and

$0 \leq i \leq t-l$. Then check the generated endpoint against the m saved chains for a match. If we assumed that the original chains cover all key space then a match will be found.

- Once a match is found then the attacker will rebuild that chain. Using SP_j and computing encryption forward. The attacker will stop when $C = K_i = E_{K_{i-1}}(P)$.
- The secret key for C will be $K_{i-1} = Enc_{k_{i-2}}(P)$ which was already computed. The online phase will require at most $(t-1)$ encryptions as time complexity

Regardless of our initial assumption in the application of the attacks, the overlaps between the chains are an important issue to be considered. In [101], Hellman considered that $n > k$ because of the application on DES where $k = 56$ bits and $n = 64$. As explained before reduction function was used to match through truncation of padding (for $n < k$) the length of the block to the length of the key. This introduced false-positive and false-negative values because of the overlap initiated merges in the chains. This indicated that the number of the keys under study by the matrix are less than mt.

The time-memory trade-off attack success probability depends on time and memory complexities that are invested in the attack. If the secret key is covered by the generated chains it will be obtained. The secret key is recovered with probability $\frac{mt}{2^k}$ assuming that all the mt unique key values. This is the reason that the choice of the correct m and t is crucial for the success of this attack. Hellman discussed that this choice should be $m = t = 2^{\frac{k}{3}}$. The success probability of the attack will be $2^{-\frac{k}{3}}$. In order to improve this probability Hellman proposed to use $T = 2^{\frac{k}{3}}$ different tables with different reduction functions for the key lookup to avoid overlaps between tables. The memory complexity at this point will be $2mT = 2^{\frac{2k}{3}+1} \approx 2^{\frac{2k}{3}}$, and the time complexity of performing $(t-l)$ encryptions for each table is $(t-1)T \approx 2^{\frac{2k}{3}}$ evaluations (encryptions and reductions). To prevent such an attack the random values can be inserted for encryption along with the plaintext.

This technique was improved by Oechslin [162] through using *rainbow tables*. These table are generated using different $(t-1)$ reduction functions at each point in the chain to reduce the possibility of reduction and possible merge between the chains. In [131] used distinguished points (DPs) that holds certain properties (for example certain number of bits to be set to zero) in addition to the original SPs and EPs in the chain. These points will bring the number of table lookups by the original TMTO down, yet will introduce a variable length of the chain generated.

2.5.4.4 Meet-In-The-Middle Attacks:

Meet-in-The-Middle (MiTM or MIM) attack algorithm was first proposed by Diffie and Hellman in 1977 to evaluate the level of security of multiple encryptions of block ciphers as in variants of DES referred to as *Double DES (2DES)* and *Triple DES* [81]. These

multiple encryptions constructions are meant to potentially increase the key security of the design from k to nk based on n number of applications of the original encryption. Each application might has a different key of length k. For example, in DES variant 3DES3

$$3DES(K_1||K_2||K_3, P) = DES(K_3, DES(K_2, DES(K_1, P))) = C$$

which will increase the key length to 168 bits and possibly the security margin of the primitive. However, MIM attacks illustrated that this is not the case in all situations. This attack reduced the security of 2DES from key of 112 bits to a key of 57 bits. The attacker will be required to have a set of chosen plaintexts and compute a match in an intermediate middle value in the forward direction (encryptions) and in the backward direction (decryptions).

If we assumed that the attacker has a set of plaintexts and ciphertexts (P, C) where $C = E_{K_2}(E_{K_1}(P)$ and $P = D_{k_1}(D_{k_2}(C))$, E is the encryption function , and D is the decryption function. Then the attacker can compute in the forward direction all possible $(E_{K_1}(P))$ for all possible values of K_1. Then in the backward direction $(D_{k_2}(C))$ all possible values of K_2. A match will be found between these two intermediate values to indicate the correct keys (k_1, k_2). These keys are checked against the available plaintexts/ciphertexts. To recover a unique key this set should be of size $\lceil \frac{k}{n} \rceil$ where k is the size of all distinct keys used in the multiple encryptions and n is the block size. For 2DES it is 112 bits in the key and 64 bits in the block size. The number of computations is 2^{k_i+1} evaluations(encryptions or decryption). k_i is the size of a single distinct key involved in the application of encryption in a single direction. The memory complexity corresponds to the size of the table that saves one of the forward or backward direction values $(PorC, k_i)$ where $0 \leq i \leq 2^{k_i} - 1$ is $(k_i + n)2_i^k$ memory unit (or in bits). This is a considerable amount of memory in some cases and TMTO can be used to present a reasonable trade-off in the attack [154]).

Although 2DES was followed up by 3DESNK where three applications of DES is applied for an encryption and NK is the number of keys used. 3DES2 which is based on two distinct keys as follows,

$$3DES2(K_1||K_2, P) = DES(K_2, DES^{-1}(K_1, DES(K_2, P)))$$

In addition to 3DES3 which is based on three distinct keys as follows,

$$3DES3(K_1||K_2||K_3, P) = DES(K_3, DES^{-1}(K_2, DES(K_1, P)))$$

Note that DES^{-1} is the decryption function which meant for 3DES3 and 3DES2 to become easily compatible with DES since

$$DES(K, P) = 3DES3(K||K||K, P) = 3DES2(K||K, P)$$

Still 3DES3 is a target of MIM attacks and its actual security level is bounded by the length of two keys instead of three (112 bits instead of 168 bits) [154].

Since they are a class of generic attacks, the application of MIM attacks spans beyond block ciphers as it also covers other cryptographic primitives as in hash functions. The core idea is also utilized in many other cryptanalytic techniques as in biclique cryptanalysis method [48]

Chapter 3

Cryptanalysis Methods

> Clearly, confidence in the security of any cryptographic design must be based on the resistance against effective cryptanalysis after intense public scrutiny.
>
> J.Daemen, L.R.Knudsen and V.Rijmen:1997 [73]

As stated in the previous chapters, cryptanalysis comes into different forms in order to support that rigorous analysis of the structure cryptographic primitive to evaluate and verify its claimed security margins. This analysis will follow the attack models represented previously in order to exploit possible weakness in the primitive (i.e weak confusion or diffusion layers). Thus, achieving the associated attack goals which will vary from a distinguishing attack to a total break that is defined based on the security margins or claims of the primitive under study. For example, for a hash function, total break constitutes finding a collision or obtaining the message from the hash value. While in block ciphers it revolve around recovering the secret key. When it comes to the claimed security margins, the design approaches will follow certain security models as in provable security or practical security or a mixture of both. The role of cryptanalyst is to subject these primitives to different existing categories of cryptanalysis approaches and tailor new ones that will push the design's security margins if possible to new limits where these attacks are not applicable any more As it is stated previously, These cryptanalytic approaches followed will be either a generic types that treats the primitives as a black box especially under the assumption that there is no weakness in the underlying structure. On the other hand, they can be also customised approaches that targets certain weakness in the design considering the underlying structure of the primitive in addition to the potential weakened design on a reduced version with respect the number of rounds. The lack of attacks does not imply that the proposed design is secure in absolute terms. It just indicate that it is relatively secure in terms of available cryptanalytic approaches at that moment. These collective efforts will serve to achieve better understanding about the actual security margins of the full algorithm.

This chapter will introduce the prominent methods of cryptanalysis that utilize certain

behaviour in the cipher structure. Such behaviour disturbs the assumed randomness of the output or the ciphertext. This chapter will explore the basic definitions of prominent cryptanalysis methods that targets the specific structure of a cipher namely differential and linear cryptanalysis and their different variants. It will also discuss other potential crytpanalytic methods that are usually used in symmetric-key ciphers analysis especially block ciphers.

3.1 Differential Cryptanalysis

Differential cryptanalysis is mainly a chosen plaintext attack that is considered one of the most utilized tools in achieving favourable attack results on different cryptographic primitives in general. It has been initially identified by the designers of Data Encryption Standard (DES) in [80] and was later invented and published by Biham and Shamir in [37]. The key goal is to trace the input/output difference propagation through the cipher structure, for a specific number of rounds, and detect the non-random behaviour exhibited in the final output, with a certain probability usually hight. It is considered to be a much effective alternative to considering the values of a plaintext and its corresponding ciphertext. These difference are utilized through an XOR operation in general, yet it is potentially applicable to use arbitrary group operations, modular addition (i.e as in IDEA and SAFER) or Unsigned Non-Adjacent Forms (UNAF) as in ARX structures to indicate these differences [139], [193]. The differential property can be utilized to recover the parts of the subkeys, typically the first or the last, in a reduced r-round version of the cipher, or alternatively deduce information about the secret key. r indicates the number of rounds under study. Several chosen plaintext pairs are used, in a combination with trying all candidates for the sub-key under attack, and the expected net result is that the correct sub-key is suggested more frequently than the wrong ones, allowing the attacker to detect which is correct. Differential cryptanalysis has evolved into different variants of cryptanalysis methods that are tailored to make the attack works where differential cryptanalysis did not achieve desirables results. Such cryptanalysis methods include truncated differential cryptanalysis [130], higher-order differential cryptanalysis [130], boomerang attacks [196]and impossible differential cryptanalysis [126], [131].

3.1.1 Estimating Differential Probability

A differential property over f is defined as input/output difference over f which is indicated by $(\triangle a, \triangle b)$ or alternatively (α, β) where $\alpha = \triangle a$ is the input difference and $\beta = \triangle b$ is the output difference. The calculation of the differential probability follows the following definition,

Definition 6. *Differential probability (DP) of a differential relation over f which is indicated as (α, β) can be expressed as*

$$DP(\alpha, \beta) = 2^{-n}\#\{x \in \mathbb{F}_2^n | (f(x \oplus \alpha) = f(x) \oplus \beta)\}.$$

Where n is the input/output size of f, x is an input of f. This probability reflects the number of possible pairs that satisfies the differential relation over f.

This difference in the differential property is usually achieved through an XOR, since it is the common operation for the key mixing layer. In addition to other modular group operations as in modular addition as mentioned earlier. The difference operation is chosen to eliminate the effect of the key used in the system when applying differential relation to undergo the attack. For example, if we have two ciphertexts/plaintext pairs (c, m) and $(\hat{c}, \hat{m})$ produced by $c = f(m) = m \oplus k$ where the key is k and f is the encryption process. We will obtain $c = m \oplus k$ and $\hat{c} = \hat{m} \oplus k$. Applying the difference relation we will obtain $\hat{c} \oplus c = \hat{m} \oplus m \oplus k \oplus k = \hat{m} \oplus m$. Similarly if we considered modular addition over certain integer value ($\boxplus$) the same concept will hold, yet the additive inverse of an element over the group (or ring of integers) should be considered when the difference is applied. In this sense, $\hat{c} \boxplus c^{-1} = \hat{m} \boxplus m^{-1} \boxplus k^{-1} \boxplus k = \hat{m} \boxplus m^{-1}$ given that $c = m \boxplus k$ and $\hat{c} = \hat{m} \boxplus k$.

It should be noted that the non-linear layer in cryptographic primitive or specifically block ciphers as in S-box layer or ($\boxplus$) modular addition is the main target to reflect the probability of holding such difference. Since the input difference will yield a certain output difference with specific probability when passing this layer. This is not the case for the linear layer which does not affect the difference operation. This indicates that the probability of this differences existing after passing the linear layer is 1. This is not the case for the non-linear layer. The differences will propagate through this layer with certain probability. A difference distribution table for an S-box can be constructed where all probabilities of all possible input/output differences are stated. Similarly to Definition 6, For an S-box $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ the difference over the S is defined as

$$DP(\alpha, \beta) = 2^{-n} \#\{x \in \mathbb{F}_2^n | (S(x \oplus \alpha) = S(x) \oplus \beta)\}$$

If f is parametrized by a key K of size k, then the differential probability $DP(\alpha, \beta; K)$ can be defined for random variable K uniformly distributed over the key space of size 2^k where $K \in \kappa$. This will motivate the computation of the average differential probability over the full key space [77]. This term is referred to as *Expected Differential Probability (EDP)* which can be defined as the following:

$$EDP(\alpha, \beta) = \frac{\sum_{K \in \kappa} DP(\alpha, \beta; K)}{2^k}$$

3.1.2 Differential Characteristic to Differential

In practice symmetric-key primitives in general and block ciphers in specific are built from compositions of a single transformation or round function (encryption of decryption). This concept is what is referred to as iterated cipher in Definition 5 in the previous chapter. For iterated block ciphers an encryption and similarly decryption is defined

by a composition of a function (i.e rounds) $E_K = R_{k_r}^{(r)} \circ R_{k_{r-1}}^{(r-1)} ..... \circ R_{k_1}^{(1)}$. Usually R is parametrized by a round key or certain round constant which means its the same application of R using different keys and corresponding round input and output. In [139], Lai and Massey analyzed the security of iterated block ciphers against differential cryptanalysis in a class of iterated block ciphers referred to as Marcov cipher which can be defined as follows

Definition 7. *An iterated cipher with round function $y = f_{k_i}(x)$ is a Markov cipher if there exist a difference operation defining $\triangle$ such that $Pr(\triangle y = \beta | \triangle x = \alpha, x = \gamma)$ is independent of γ for all possible input/output differences masks α and β given that the round key k_i is chosen uniformly at random.*

Differential properties (similarly linear properties) are examined on a scale of one round and then extended on the rest of the rounds. This concept is referred to as *Differential Characteristic (DC)* in differential cryptanalysis.

Definition 8. *A differential characteristic DC is a sequence of intermediate differences through the different steps of encryptions at each round of the composition of rounds under study.*

$$\triangle_0 \rightarrow R_1 \rightarrow \triangle_1 \rightarrow ..... \rightarrow \triangle_{r-1} \rightarrow R_r \rightarrow \triangle_r$$

The sequence is indicated by an input difference and a collection of output differences of each step.

An element in this sequence can be alternatively denoted as $\alpha \xrightarrow{R} \beta$, where α is the input difference that will result into an output difference β over R. Each input/output difference in the sequence (element) exists with certain probability that corresponds to the differential characteristic probability.

$$DP(DC) = Pr(\triangle_0 \rightarrow R_1 \rightarrow \triangle_1 \rightarrow ..... \rightarrow \triangle_{r-1} \rightarrow R_r \rightarrow \triangle_r) = \prod_{i=1}^{r} Pr(\triangle_{i-1} \rightarrow R_i \rightarrow \triangle_i)$$

or alternatively

$$DP(DC) = \frac{\#\{x \in \mathbb{F}_2^n | (R_r \circ R_{r-1} ..... \circ R_1)(x \oplus \triangle_0) = (R_r \circ R_{r-1} ..... \circ R_1)(x) \oplus \triangle_r\}}{2^n}$$

Assuming these characteristics are independent, the probability of accumulating characteristics that will take an input difference to output difference over multiple applications of R and intermediate differences is the product of these characteristics. This is because of ciphers that follows Definition refMarkov and form a homogeneous Markov chain. Such that sequence of differences in the chain of differences where every intermediate consecutive differences are independent of each other, and the probability distribution for all consecutive differences is the same. It is worth noting that the differential probability of a differential characteristic indicates the fraction number of *right pairs* that will satisfy given input/output difference.

When the differential probability of a r-round differential characteristic is too small to be used directly, the cryptanalyst uses $\alpha = \triangle_0 \xrightarrow{r-rounds} \beta = \triangle_r$ where differences in between the rounds are not considered. Only input and output difference will be taken into consideration. This is usually referred to as *r-round differential* which are a collection of several r-round differential characteristics that starts and end with the same input/output difference. They can be also referred to as *differential paths or trails*. The probability of such differentials is

$$DP(\alpha, \beta) = Pr(\triangle_r = \beta | \triangle_0 = \alpha) = \sum_{DC \in (\alpha, \beta)} DP(DC)$$

where a differential would have an input $\alpha = \triangle_0$ and output $\beta = \triangle_r$ of the difference approximation (α, β). This can be viewed as having many trails between $\alpha \rightarrow \beta$ where each r-round differential characteristic with the specific intermediate differences is a single differential trail or differential path of $\alpha \rightarrow \beta$ [131].

These differential probability of the various differential relations over R can be expressed in a matrix that is referred to as difference transition probability matrix (M) [139]. This matrix will be constructed for n-bit Markov cipher with dimensions of $(2^n - 1)(2^n - 1)$. each entry will constitute the probability of the differential characteristic with corresponding input/output differences to a matrix row and column respectively. For r-round differential between the input/output differences, the entries in M^r will reflect their probabilities.

In the application of differential cryptanalysis, cryptanalyst aims to obtain the best or highest possible differential probability given that it obtains reasonable attack complexity. This can be also referred to Maximum Differential Probability (MDP)

$$MDP(R) = MAX_{\alpha \neq 0, \beta}(DP_R(\alpha, \beta))$$

where the maximum is taken over all possible differential probabilities over all possible input/output difference except zero input differences. The same concept can be applied to obtain the maximum differential r-round characteristic probability under the assumption of rounds independence as follows,

$$MDCP(R) = \prod_{i=1}^{r} MDP(R_i)$$

This can be utilized in evaluating the security of a cryptographic primitive, in this context SPN-based block ciphers or Feistel ciphers with SP components, against differential cryptanalysis. Deciding the minimum number of active S-boxes (with non-zero input differences) or a lower bound on this number is a practical approach of this evaluation. This can be achieved through approximating an upper bound on maximum differential characteristic probability.

$$MDCP(R) \leq MDP(S)^{\#activeS-boxes}$$

As discussed earlier, the branch number (Bn_D) is referred to as the minimum number of active S-boxes of non-zero input difference for Substitution-Permutation-Substitution layers. $Bn_D \leq NS + 1$ where NS is the number of parallel S-boxes in the substitution layer. If both linear and differential branch numbers (Bn_D, Bn_L) are within their maximum values, the diffusion layer is referred to as MDS The maximum differential characteristic probability for r rounds of SPN encryption is expressed as

$$MDCP(R) \leq MDP(S)^{Bn_D \frac{r}{2}}$$

for even number of rounds, and

$$MDCP(R) \leq MDP(S)^{Bn_D \frac{r-1}{2}+1}$$

for odd number of rounds [76], [166].

As for ARX structures, bit-level differential analysis is carried out which indicates that finding differential paths might be complex [41], [123]. This form of differential analysis contains analysis on the probability distribution of for integer addition with carry [185], differential probability for modular addition [147], and differential probability of XOR for differences that use modular addition where an algorithm using matrix multiplication is utilized [144], [148].

3.1.3 Key Recovery and Data Complexity

Considering the target of the cryptanalysis is block cipher with an input block of size n, a secret key of size k, and iterating on r number of rounds. The cryptanalyst will aim to construct an effective $(r-1)$-round differential (α, β) or $(r-1)$-round differential characteristic (α, β) for a reduced number of rounds usually $r-1$. The attack complexity achieved by this differential characteristic or differential should be considerably less than the full codebook 2^n to be deemed usable in less than exhaustive search terms. In other words, the differential probability p for this characteristic or differential should be $p = DP(DC) >> 2^{-n}$.

Given a good differential $(r-1)$-round differential (α, β) exits with probability p. The attacker will need $\mathcal{O}(p^{-1})$ plaintext pairs that hold the difference α for a successful attack. Given plaintext pairs, output difference for the corresponding ciphertext pairs that hold difference β will be searched. The number of the matches for N given plaintext pairs is pN.

The attacker's goal of key recovery is to use the filtered plaintext/ciphertext pairs, and the established differential relation (α, β) to (partially or fully) guess the last round r key k_r. Using the guessed key k_r for a partial decryption for the given ciphertext pair $(c_r, c_r^{'})$ which obtains $\beta = \triangle_{r-1}$. This indicates that the guessed key is a candidate key. This will be achieved for each key in $2^{|k_r|}$ where $|k_r|$ is the size of the round key and every pair of the filtered set. The candidates keys with the highest number of

matches for the differential characteristic on the set of filtered pairs will be the right key. The time complexity for the guessing phase is less than $2^{|k_r|}$. In principal values of input differences, intermediate differences, output differences, and the keys will affect the differential probability of a certain $(r-1)$-round characteristic. The attacker will have to evaluate probability of a differential or a characteristic in correspondence to the all possible key values obtained and pairs of plaintexts. Since this is not practical scenario as most of the pairs of plaintext/ciphertext obtained using some fixed key. Stochastic key equivalence is established under the assumption that round keys behave independently and most of the secret keys will behave similarly. It states that the differential probability of obtaining a certain output difference after $r-1$ rounds given a certain input difference is approximately the same to the differential probability of obtaining the same output difference after $r-1$ rounds given the same input difference and the associated rounds keys. This is not true for all ciphers as in IDEA [53] , [139].

As mentioned in the previous chapter, the probability that the correct key will fall within the presented guesses is $P_{correct}$. The probability that the wrong key will fall within the presented guesses is P_{wrong}. If $P_{correct} > P_{wrong}$, then the right key will be identified against the obtained wrong keys by counting the number of pairs that will verify the right key guesses where the highest count will specify the right key. To distinguish if the presented differential analysis is effective to recover the right key value, the Signal-to-Noise (S/N) value is estimated as $S/N = \frac{P_{correct}}{P_{wrong}}$. If this value is not 1 then the differential attack's success rate will be higher the further the value from 1. If this value $S/N > 1$ the value of the counter indicating the right key will be the highest value. If this value $S/N < 1$ the value of the counter indicating the right key will be the lowest value. If $S/N = 0$ then a variant of differential cryptanalysis called impossible differential cryptanalysis can be used [37], [131], [36]. Consequently, the S/N value is used to evaluate the success probability of the differential cryptanalysis by computing the bit advantage given by the attack. This shows that the success probability depends on the key bits guessed, the data complexity and the differential probability. As mentioned in the previous chapter, the advantage of the attack can be evaluated with respect to the guessed k key bits in the attack where the correct value is ordered f among the possible space 2^k. Then the attack will have $(k - log_2(f))$-bit advantage over exhaustive search or brute force attack. In [178], the success probability (P_S) with a-bit advantage or higher of a differential attack is stated to be

$$P_S = \Phi\left(\frac{\sqrt{pN(S/N)} - \Phi^{-1}(1-2^{-a})}{\sqrt{(S/N)+1}}\right)$$

This is under the assumption that key counters are independent and they are identically distributed for all wrong keys.

θ is a cumulative distribution function of the standard normal distribution. While the data complexity of differential cryptanalysis associated with this P_S is reciprocal to p as follows

$$N = \frac{(\sqrt{(S/N)+1}\Phi^{-1}(1-2^{-a}))^2}{(S/N)}p^{-1}$$

Further estimates on the success probability and data complexity is provided in [44]. There are several extensions and variations of differential cryptanalysis that were introduced to target different cryptographic primitives or specifically block ciphers based on their accurate structure and differential behaviour. This include truncated differential cryptanalysis, impossible differential cryptanalysis and higher-order differential cryptanalysis. Although they might be restricted to certain differential behaviour or structure, yet when performed they yield relatively better results than classical differential cryptanalysis in terms of data complexity and number of rounds attacked.

3.1.4 Truncated Differential Cryptanalysis

Introduced by Knudsen in [130], *truncated differential cryptanalysis* is an extension to differential cryptanalysis. It tends to be used on ciphers that appear to be resistant to classical differential cryptanalysis.

The basic idea is to construct a partially known input/output differences where after certain number of rounds only a part of the output difference is known. It shows that to construct differential property it is not necessary to know the full n-bit difference as few bits will be enough. It is achieved by applying a truncation for the differential property that represents the bits as known bits value as (0) which indicates same value in the same bit location. In addition to $(*)$ where it denotes an unknown or free value that could be 1 or 0 so values at the indicated position can be similar or different. This can be also applied to bytes where non-zero byte differences will be grouped together. Truncated differential characteristic can be viewed as set of characteristics which resembles the case of differential. Considering bits that are the same in value and in the same location in the output differences will obtain higher probability than classical scenario of focusing only one output difference. Truncated differential approach might not be practical for certain designs as they tend to achieve best results for ciphers with slow diffusion layers as truncated differentials tend to spread the differential properties to the whole state faster than differentials. They tend to provide better results in comparison to classical differentials when applied as they might provide better differential probability, lower data complexity or an increment to the number of rounds attacks [46].

Several extensions to this approach have been initiated as in multiple differential cryptanalysis where multiple input differences are taken into consideration such that the respective output difference is based on the specific input difference [43]. Truncated differential cryptanalysis approach was applied on block ciphers SAFER [132], IDEA [127], Crypton [124] and Skipjack [134].

3.1.5 Impossible Differential Cryptanalysis

Impossible differential cryptanalysis was first mentioned in 1998 by Knudsen in his analysis of DEAL [**?**], and further extended to an attack on IDEA by Biham et al. at FSE 1999 [35]. Instead of approaching differentials that have high differential probability, impossible differentials rely on zero differential probabilities as applied on SKIPJACK [34]. The approach combines two certain properties (two differentials with probability 1), one in the forward direction and one in the backward direction, and uses a resulting conflict (miss-in-the-middle) when both directions are joined. This miss-in-the-middle approach is used to obtain an impossibility result. This can be utilized in a chosen-plaintext attack by requesting encryptions of plaintext pairs with a fixed difference, guessing key material and checking for the impossibility property to discard wrong guesses. The forward and backward can be differentials which are truncated.

The details of this attack will be revisited in Chapter **??**.

3.1.6 Higher Order Differential Cryptanalysis

Higher order differential cryptanalysis is considered to be another extension or a generalization of differential cryptanalysis. It was initially proposed by Lai [137] when he showed that higher order derivatives can be used as a general case to differentials. Knudsen in [130], was able to initiate higher order differential cryptanalysis on five round Feistel cipher with a round function that have quadratic polynomial. Higher order differential cryptanalysis is usually used against ciphers that claims security against differential attack. They aim is to establish derivatives on the differences considered on the round function to reduce the algebraic degree of f making it easy to be analysed.

A first order derivative at point α is the classic output difference used in differential cryptanalysis. This indicated by $\triangle_\alpha f(x) = \beta = f(x) \oplus f(x \oplus \alpha)$ where $\alpha \neq 0$ is an input difference, $\beta \neq 0$ is an output difference, $\alpha \neq \beta$,and $x \in \mathbb{F}_2^n$. Following the same line of thought, d^{th} derivative or d-th order differential at the input differences points $\alpha_1, ..., \alpha_d$ is

$$\Delta_{\alpha_1 ... \alpha_d} f(x) = \triangle_{\alpha_d}(\Delta_{\alpha_1 ... \alpha_{d-1}} f(x)) = \bigoplus_{c_i \in \{0,1\}} f(x \oplus c_1\alpha_1 .. \oplus c_d\alpha_d)$$

Which is a recursive application of the derivative to the points from α_1 to α_d. Instead of working with pairs to establish the differences as in classical differential cryptanalysis, the attack focuses on application of 2^d states. Once the d^{th}-order differential is established, it can be used to recover the right key knowing the algebraic degree of f is d and d is low. Then the key guess that will yield a zero value of $(d+1)$-order derivative is the right key. This approach of cryptanalysis is used for ciphers with low degree round functions (i.e small S-boxes) [63]. It is recommended to establish resistance against such attacks to have a maximum algebraic degree of f which is not low, and increase the

number of rounds especially for the effective differential trail. This will increase d hence the number of chosen plaintext pairs needed and the computational complexity of the attack.

3.2 Linear Cryptanalysis

Linear cryptanalysis [153] is a well-known cryptanalytic technique that has been employed on several block ciphers. Examples include the DES, FEAL-4, Serpent, Shannon and SAFER [67, 96, 116, 153, 181]. The most important fact about the linear cryptanalysis is that it is a known plaintext attack, which is a more practical and realistic attack model that that of differential cryptanalysis which works under the chosen plaintext model. Linear cryptanalysis tries to find a highly probable linear expressions involving plaintext bits, ciphertext bits and the subkey bits as

$$\bigoplus_{i\in\mathcal{P}} P_i \bigoplus_{j\in\mathcal{C}} C_j = \bigoplus_{w\in\mathcal{K}} K_w.$$

for some sets $\mathcal{P}, \mathcal{C}, \mathcal{K} \subset \{0, \ldots, N-1\}$, and P, C, and K represents the plaintext, ciphertext and key, respectively. In this scenario, the attacker has no way to select which plaintexts (and corresponding ciphertexts) are available, which is a reasonable assumption in many applications and scenarios. Then it tries to utilize the linear bias or the correlation of this linear relation to detect a certain non random behaviour in the cipher.

The linear approximation can be alternatively expressed as

$$\langle \alpha, P \rangle \oplus \langle \beta, C \rangle = \langle \gamma, K \rangle, \quad \text{where} \quad k \in \mathbb{F}_2^{|K|}$$

Where α and $\beta \in \mathbb{F}_2^n$ are the input and output linear masks such that probability of the presented linear approximation vary from $\frac{1}{2}$ in order for this approximation to be used for an attack and to be distinguished from random stream of data.

3.2.1 Linear Characteristic and Linear Hulls

As in differential characteristic, these approximations are traced within the cipher structure (rounds) to form a *linear characteristic*

Definition 9. *A linear characteristic is a sequence of linear approximations or relations for a given number of rounds. Each element in this list will determine the linear bias or the correlation of the respective linear characteristic.*

These approximations can be expressed as $\Gamma_{I_0}, ..., \Gamma_{I_r}$ where the list of input/output/intermediate values are stated as the following $I_0 = P, ..., I_r = C$.

As in the relation between differentials and differential characteristics, the set of linear characteristics with input/output masks ($\alpha = \Gamma_{I_0} = \alpha_0, \beta = \Gamma_{I_r} = \alpha_r$) respectively is referred to as a linear hull of the approximation (α_0, α_r). In this respect as with differential trails, each r-round characteristic with the following sequence $(\alpha_0, ..., \alpha_r)$ is a trail between the starting input/output masks where there can be more than one trail between these two masks.

3.2.2 Linear Probability Estimations

As we have discussed in the previous chapter, the linear layer (i.e permutation layer) will preserve the linear property with probability 1, while the non-linear layer (i.e S-box or Modular addition) will propagate it with certain probability.

The combined probability for these linear approximations are calculated using the piling-up lemma introduced by Matsui in [152]. It states that accumulated probability of $n$independent linear approximations such that $\Gamma_{I_{i+1}}.I_{i+1} = \Gamma_{I_i}.I_i$ which follows probability $p_i = \frac{1}{2} + \epsilon_i$ is

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

Where ϵ_i is the respective linear bias of the presented approximation. Similarly, a correlation of the linear expression can be expressed in terms C which can be defined as

$$C = \prod_{i=1}^{n} C_i(\alpha_{i-1}, \alpha i)$$

Where $C_i = 2.\epsilon_i = 2p_i - 1$. The best linear approximation is the one that is obtained with the highest bias, since the data complexity N (number of plaintexts) required by the attack will be reduced as proved by Matsui $N = \frac{c}{\epsilon^2}$ where c is a negligible constant [152]. As pointed out previously there can be more than one linear path between input/output mask (α, β) each path is denoted as r-round characteristic $(\alpha = \Gamma_{I_0} = \alpha_0, ..., \beta = \Gamma_{I_r} = \alpha_r)$. As a result the correlation of j-th r-round linear characteristic can be expressed as previously stated as

$$C_i = \prod_{i=1}^{r} C(\alpha_{(i-1)j}, \alpha_{ij}).$$

Hence, the correlation of the linear approximation$C(\alpha, \beta)$ over the number of all possible trails NT with the same input/output mask (α, β) is $C(\alpha, \beta) = \sum_{i=1}^{NT} C_i$.

It is worth noting that linear probability of an approximation (α, β) is the square of its correlation $Pr(\alpha, \beta) = C(\alpha, \beta)^2 = 4\epsilon^2$ and it usually ranges from 0 to 1 [77].

The correlation of input/output linear masks $C(\alpha, \beta)$ can be also expressed using Walsh-Hadamard transform of the function understudy F as well

$$C(\alpha, \beta) = 2.\epsilon = \frac{\hat{F}(\alpha, \beta)}{2^n} \tag{3.1}$$

To go through this deduction certain notations shall be introduced [65]. First, starting with Fourier transfer on a boolean function which can be defined as $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$\hat{f}(a) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{\langle a,x \rangle} \quad \text{where} \quad a \in \mathbb{F}_2^n$$

In this case, $\langle a, x \rangle$ is the dot product between the two values. Applying this Fourier transform on the sign function $f_x = (-1)^f = 1 - 2f$ is going to obtain the difference between the number of times the function f and $\langle a, x \rangle$ are equal and differ from each other. This is referred to as Walsh Transform of f

$$\hat{f}_x(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \langle a,x \rangle}$$

To denote the probability that $f(x) = \langle \alpha, x \rangle$ which is $P_{eq} = Pr(f(x) = \langle \alpha, x \rangle)$. Let us assume that $x = u$, then $N_{eq} = \#\{f(u) = \langle \alpha, u \rangle\}$, and $N_{\text{diff}} = 2^n - \#\{f(u) = \langle \alpha, u \rangle\} = \#\{f(u) \neq \langle \alpha, u \rangle\}$. This indicates that $\hat{f}_x(\alpha) = Neq - N\text{diff} = 2Neq - 2^n$. Consequently, $P_{eq} = Pr(f(x) = \langle \alpha, x \rangle) = \frac{N_{eq}}{2^n}$. Applying this in the previous Walsh transform result yield $\hat{f}_x(\alpha) = 2Neq - 2^n = 2^{n+1}(p_{eq} - \frac{1}{2})$. Alternatively, $p_{eq} = \frac{1}{2} + \frac{1}{2^{n+1}} \hat{f}_x(\alpha)$ Similarly, Walsh transform can be applied to a vectorial Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ for input/output linear masks $(\alpha, \beta) \in \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined by

$$\hat{f}_x(\alpha, \beta) = \sum_x (-1)^{\langle \beta, f(x) \rangle \oplus \langle \alpha, x \rangle}$$

Where the $p_{(\alpha,\beta)} = \epsilon + \frac{1}{2}$ is the probability of approximating $\langle \beta, f(x) \rangle$ to $\langle \alpha, x \rangle$. Following the previous probability calculation, $p_{(\alpha,\beta)} = \frac{1}{2} + \frac{1}{2^{n+1}} \hat{f}_x(\alpha, \beta)$ which will help us reaching the correlation calculation indicated by the expression 3.1

These correlation and bias calculations are usually used to construct a linear approximation table on the the non-linear function in the cipher which is usually the S-box in SPN structures of block ciphers. The linear approximation table will represent all possible linear input/output masks and their associated probabilities. For example, if $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ for small integers m and n with input/output masks (α, β) then

$$S_{(\alpha,\beta)} = \#\{X \in \mathbb{F}_2^n \text{such that} \langle \beta, S(X) \rangle = \langle X, \alpha \rangle\}$$

The probability and correlation of the linear input/output masks (α, β) can be expressed as the following [2, 76]:

$$P_{(\alpha,\beta)} = \Pr(\langle \beta, S(x) \rangle = \langle \alpha, x \rangle) = \frac{S_{\alpha,\beta}}{2^n} \quad \text{and} \quad C(\alpha, \beta) = 2 \Pr(\langle \beta, S(x) \rangle = \langle \alpha, x \rangle) - 1$$

In [76], An estimation of linear trails correlation was stated stated for key alternating ciphers which is a subclass of Markov ciphers that will alternate the use of number of

keys (XORing or adding the keys to the rounds) with a key independent instances of round functions . It was noted that when considering linear trails it should be noted that they are key dependent where only the sign of the correlation depends on the key. The correlation of the linear hull will be the sum of these trails' correlations.

3.2.3 Key Recovery and Data Complexity

Matsui proposed two algorithms for recovering key information using linear cryptanalysis. The first algorithm which is referred to in the literature as Matsui's Algorithm 1 which is considered a distinguishing attack since it aims to find one bit of the key based on the obtained linear probability or correlation of the linear approximation (α, β). While Matsui Algorithm 2 will find part of the last round key based on the obtained linear probability or correlation of the linear approximation of the r rounds given a large number of plaintext and cipher text pairs.

The data complexity and the success probability of linear cryptanalysis over exhaustive search is dicussed in [178]. Assuming that the presented linear approximation probability will not reveal anything for the wrong key ($Pr = \frac{1}{2}$), and will be independent for each candidate key. The success probability P_S with respect to at least a-bit advantage is measured with respect to the needed N plaintexts/ciphertexts where a and N are large is as follows:

$$P_S = \Phi(2\sqrt{N\epsilon} - \Phi^{-1}(1 - 2^{-a-1})$$

This corresponds to the following N number of pairs needed for a successful attack which is proportional to ϵ^2:

$$N = (\frac{\Phi(P_S) + \Phi^{-1}(1 - 2^{-a-1})}{2})^2 \epsilon^{-2}$$

Linear Cryptanalysis based on chosen plaintexts was proposed on DES to reduce the data complexity of the original attack [133]. There are various extensions to classical linear cryptanalysis that aims as well to reduce the complexity or improve the attack by recovering more key bits. One example is using multiple linear approximations to obtain more information about the key bits [115], [39], [24], [103], [104]. These results were improved in multidimensional linear attacks because they involve many key, several key bits of K can be derived at the same time [103], [104]. Thus, the data complexity will be reduced to the capacity based on the bias of all biases of different approximations $Cap = \sqrt{\sum \epsilon_i^2}$. However, due to the simultaneous processing of multiple approximation the time complexity of the approximation analysis has increased.

3.2.4 Zero-Correlation Cryptanalysis

Zero-correlation was proposed by Bogdanov and Rijmen in [50]. This attack can be viewed as the linear alternative of impossible differential attack. It utilizes linear approximations with probability $\frac{1}{2}$ for any key value. Thus, linear approximations having a zero correlation for any key value. The distinguishing algorithm relies on having 2^{n-1} chosen plaintext/ciphertext pairs with unknown fixed key, and zero-correlation linear hull (α, β). The algorithm then computes the correlation and check if it is zero using:

$$C(\alpha x, \beta F(x)) = \frac{|\{(x, F(x))|\alpha x = 0 \text{ and } \beta F(x) = 0\}|}{2^{n-2}} - 1$$

or

$$C(\alpha x, \beta F(x)) = \frac{|\{(x, F(x))|\alpha x = 1 \text{ and } \beta F(x) = 1\}|}{2^{n-2}} - 1$$

where n is the cipher's block size.

A key recovery attack using zero-correlation distinguisher was used on reduced round AES-192 and AES-256. The data complexity of this attack is 2^{n-1} chosen ciphertext/plaintext pairs and the same for time complexity to evaluate the input masks or output masks. The attack was improved with lower data complexity using multiple zero-correlation approximations in [51] where for l zero-correlation linear approximations the data complexity will reduced to order of $\frac{2^n}{\sqrt{l}}$. An extension based on invariant key biases was proposed in [**?**, 47].

3.3 Other Variants

Cryptographic attacks goes beyond what has been discussed so far in this chapter. As there are many variants that target certain components as in key scheduling(i.e slide attacks [42]), or the structure of the cipher as integral attacks. There attacks which targets the behaviour of the cipher in the implementation environement as in hardware attacks (i.e Differential Power Analysis, timing attacks, etc). The following sections will briefly describe integral and cube attacks.

3.3.1 Cube Attacks

A variant of higher order differentials is *cube attack*. The attack was proposed by Dinur and Shamir in [83]. The aim of the attack is to get a linear function of the secret key bits by summing over different inputs even under the assumption that attacker does not know specifics of the cipher design. An evaluation of unknown multivariate polynomial $f_K(x_1, ..., x_n)$ given inputs bits that are public and secret will be performed as follows

$$f_k(x_1, ..., x_n) = X_I.p_k(x_1, ..., x_n) + q_k(x_1, ..., x_n)$$

Where $x_1, ..., x_n$ is a vector input and K is the secret parameter, and index set $I = i_1, ..., i_k \subseteq 1, .., n$. Some of the terms in $f_k(x_1, ..., x_n)$ will be factored by the common subterm X_I. $p_k(I)$ is called a superpoly of I in $f_k(x_1, ..., x_n)$ which is a polynomial with no common variables with X_I and no monomials in q_k are in X_I.

$$\bigoplus_I f_k(x_1, ..., x_n) = p_k(x_1, ..., x_n)$$

Once $p_k(x_1, ..., x_n)$ is linear with degree one then X_I is a maxterm. Having enough linear equations the system can be solved for K. If a single output bit can be expressed in a low degree polynomial then algebraic attacks have a better chance of success. It should be noted that the attack will have a higher complexity with a bigger size of index set I, so getting the appropriate maxterm will be challenging for long maxterms yet easier for short ones. When looking for an appropriate maxterms, I is set randomly then it will be updated with indices until the superpoly is a constant value (linear relation) then the search for another maxterm will done all over. Linearity tests on various maxterms is applied to get a linear superpoly, this is referred to as cube testers [21]. Cube attacks work well for polynomials of low degrees. However, most block ciphers have relatively high algebraic degree. To be utilized for output bits might have low degree, The can be used on the outputs of NLFSR of stream ciphers. These attacks were applied on reduced round Trivium [61].

3.3.2 Integral Cryptanalysis

Integral attacks were first introduced by Knudsen in their application on SQUARE in [73] and later on was applied and generalized under different references as in multiset attacks and saturation attacks on Twofish [150]. The attack relies on constructing sets of or multisets of chosen plaintexts that either sum to a constant or differs in certain parts of the set. Thus exploiting relations between various encryptions. The main goal of the attack is to follow the preservable nature of certain properties of the sets. For example, in integral attacks, the set I of internal states are constructed such that they differ in only one byte d_0 which covers all 2^8possibilities. It is noted that this property will hold after an application of an S-box layer (or a bijective transformation) to the state I. While the diffusion layer will make the rest of the bytes active.

An n-bit value multiset can be defined as unordered tuple where values might recur. The multiset might have certain properties that can be utilized as a distinguishing factor in an attack. For instance, if all values in the multiset are the same then this multiset will have the property C that refers to a constant. If all values are different then the multiset is referred to as A or all. S refers to the fact that the sum of the values can be predicated while ? refers to the fact that it can not be predicted. There are other properties as well

that are utilized in integral or multiset attacks. If each value in the mutliset occurs zero or even number of times, then this multiset is even or E. In addition to that a multiset is a permutation or P if it holds only once every 2^m possibilities. Finally, a multiset has a property Balanced B if the XOR of all values is zero this is closer to the S property. Any multiset which is a permutation or even multiset is called dual or D multiset which is also balanced [128].

Although the original attacks were applied on byte-oriented ciphers, bit pattern based integral attacks were introduced in [204].

As stated throughout this chapter, resistance to cryptanalytic methods depends on the selected designs parameters by the designers which takes into consideration the implementation environment as well. The goal of the designers is to establish an infeasible attack margins (i.e. differential and linear probabilities) through pushing the probabilities of the characteristics presented to be small. In addition to raising the data complexity of the potential attacks. An example of such design strategies that presents certain bounds on the probabilities of differential and linear characteristics for certain design structure and given number of rounds is the wide-trail strategy which is used in AES [75]. It shows that the composite choice of nonlinear layer, linear layer and key mixing layer can control the differential and linear bounds of the attack. As important it is to select an S-box (non-linear layer) which achieve the smallest potential differential and linear probabilities, it is crucial to choose the appropriate linear layer. In common terms such linear layer shall provide the a good diffusion properties where the number of active S-boxes is maximized in the next round over certain number of rounds. Hence, affecting the propagation of linear and differential properties to be minimal.

In the next chapters we will present different cryptanalysis methods applied on different block cipher structures. Some obtain attack results while others present an observation on the security of the primitive understudy.

Chapter 4

Cryptanalysis of PRINTcipher: The Invariant Subspace Attack

The scope of the design of block ciphers is considered a mature material when compared to other primitives design as in hash functions and stream ciphers. However, the challenge is still rising when it comes to have a perfect trade-off between security and efficiency.

As discussed in previous chapters, most lightweight block cipher designs aim for better cost trade-off in hardware given various metrics (i.e. area, power, energy, throughput,..,etc). This has shed the light on a competitive approach to provide lightweight designs that can support better security margins in a very constrained environments. The focus of the design approaches were either to improve current known designs to fit them to lightweight requirements, or provide various customized and specific designs to support better performance. Both approaches might provide less understood designs, more complex, and less standardized designs in term of security when it comes to security margins. Consequently, such designs were a feast to many attacks (standard ones or new). Regardless, if they utilized a full break on lightweight primitives or partial one, such attacks improved the understanding of the security margins of lightweight ciphers in specific and block ciphers in general. Note, that an attack the breaks lightweight cipher might be prevented for regular block ciphers just because of the security margin of the later designs (more rounds, longer keys).

This chapter will present an attack on the block cipher PRINTcipherthat breaks the full cipher for significant fracture of its keys. This attack is referred to as *invariant subspace attack* which can be also viewed as a weak-key variant of a statistical saturation attack. For such weak keys, a chosen plaintext distinguishing attack can be mounted in unit time. In addition to breaking PRINTcipher, the new attack also gives a new insights into other, more well-established attacks. In addition, we also show that for weak keys, strongly biased linear approximations exists for any number of rounds. In this sense, PRINTcipher behaves very differently to what is usually – often implicitly – assumed. PRINTcipher is an example of a non-toy cipher where attacks do not behave as we usually expect them to. The bias for statistical saturation attacks, and the bias of linear hulls are extremely key-dependent. For a weak key, increasing the number of rounds up to the full number of rounds does not increase

the security of the cipher with respect to these attacks.

4.1 Our Contributions

The main contributions of this chapter can be summerized in the following:

- Presenting the new *invariant subspace attack* on PRINTCIPHER. In a nutshell, the attack is based on the observation that for PRINTCIPHER there exist cosets of subspaces of $\mathbb{F}_2^n$ that the round function maps to cosets of the same subspace. The exact coset is determined by the round key only. Now, if the round key is such that a coset gets mapped to itself, the fact that all round keys are identical in PRINTCIPHER (almost) immediately leads to the conclusion that for certain (weak) keys some affine subspaces are invariant under encryption. The round constants, mainly introduced to avoid slide attacks, do not prevent the attack as the round constants are included in the subspace. The principle of the attack is described in Section 4.2.
- Using this attack, we demonstrate the existence of 2^{52} weak keys (out of 2^{80}) for PRINTCIPHER-48 and 2^{102} weak keys (out of 2^{160}) for PRINTCIPHER-96. If a key is weak, our attack results in a distinguisher using less than 5 chosen plain- or ciphertexts. That is, even in the case of RFID-tags, where the amount of data available for a practical attack is strictly limited, our attacks apply. In a known plain- or ciphertext scenario the data complexity increases by a factor of 2^{16} (PRINTCIPHER-48) resp. 2^{32} (PRINTCIPHER-96).
- Besides the low data complexity of the distinguisher, the attack technique has interesting relations to more established attacks which we like to highlight. The invariant subspace attack can be interpreted as a statistical saturation attack [68, 69]. Here a weak key, together with a special choice of the fixed bits in a statistical saturation attack, leads to a maximal bias, independent of the number of rounds. Taking into account the close relation of statistical saturation attacks to multi-dimensional linear attacks, we show that the invariant subspace attack implies the existence of strongly biased linear approximations for weak keys, again independent of the number of rounds. Details can be found in Section 4.6.

It is worth mentioning that the presented work in this chapter is published in [143].

4.2 General Idea

Consider an n-bit block cipher with a round function E_k consisting of a key addition and an SP-layer

$$E : \mathbb{F}_2^n \to \mathbb{F}_2^n,$$

that is E_k is defined by $E_k(x) = E(x + k)$. Assume that the SP-layer E is such that there exists a subspace $U \subseteq \mathbb{F}_2^n$ and two constants $c, d \in \mathbb{F}_2^n$ with the property:

$$E(U + c) = U + d.$$

Then, given a (round) key $k = u + c + d$ with $u \in U$, the following holds:

$$E_k(U + d) = E((U + d) + (u + c + d)) = E(U + c) = U + d,$$

i.e. the round function maps the affine subspace $U + d$ onto itself. If all round keys are in $k \in U + (c + d)$ (in particular if a constant round key is used), then this property is iterative over an arbitrary number of rounds. This yields a very efficient distinguisher for a fraction of the keys. U should be as large as possible to increase this fraction. We call this new attack technique an *invariant subspace attack*. In the next section we show an example of how to apply it to the light-weight block cipher PRINTcipher.

4.3 Attack against PRINTcipher

4.3.1 Description of PRINTcipher

PRINTcipher is a block cipher proposed by Knudsen et al. at CHES 2010 [140]. It is a class of two SP-networks with a block size of $n = 48$ (resp. $n = 96$) bits, a key size of $l = 80$ (resp. $l = 160$) bit, and 48 (resp. 96) rounds. One round of PRINTcipher-48 is shown in Figure 4.1.

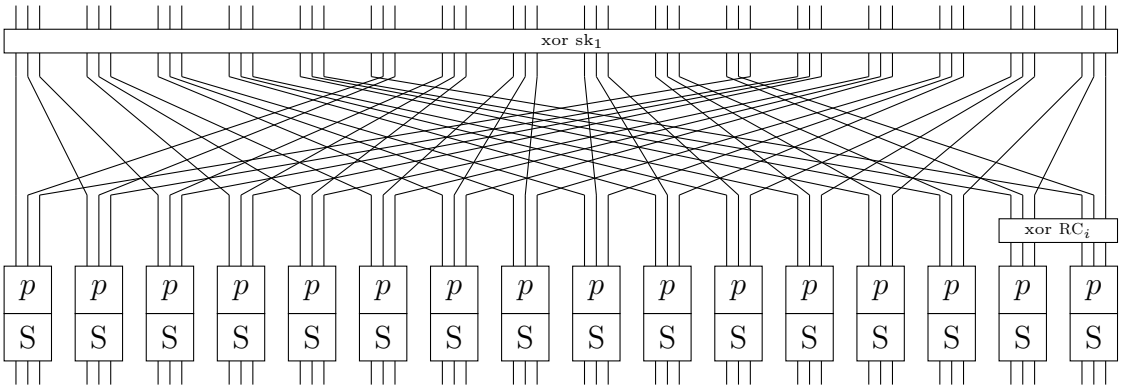


Figure 4.1: One round of PRINTcipher-48 illustrating the bit-mapping between the 16 3-bit S-boxes from one round to the next. sk_1 denotes the xor key, p the permutation key, and RC_i the round counter.

PRINTcipher uses the same key for all rounds. It is split into two parts: The first n bits are used as an xor key, the remaining $l - n$ bits control the permutations p. In order to introduce differences between the rounds, a round counter RC_i is used which is generated by an LFSR (for details, see [140]). The other elements of the round function are defined as follows.

The **linear layer** consists of a bit permutation, where bit i of the current state is moved to bit position $P(i)$ where

$$P(i) = \begin{cases} 3i \bmod n - 1 & \text{for } 0 \leq i \leq n - 2, \\ n - 1 & \text{for } i = n - 1, \end{cases}$$

where $n \in \{48, 96\}$ is the block size.

Then the state bits are arranged in 16 (resp. 32) blocks of 3 bits each, which are permuted individually in the **permutation layer**. Out of 6 possible permutations on 3 bits, only four are valid permutations for PRINTcipher. Specifically, the three input bits $c_2||c_1||c_0$ are permuted to give the following output bits according to two key bits $a_1||a_0$.

nr.	$a_1\|\|a_0$	p
0	00	$c_2\|\|c_1\|\|c_0$
1	01	$c_1\|\|c_2\|\|c_0$
2	10	$c_2\|\|c_0\|\|c_1$
3	11	$c_0\|\|c_1\|\|c_2$

Finally, in the **non-linear layer**, each 3-bit block is processed by the same s-box, which is shown in the following table.

x	0	1	2	3	4	5	6	7
$S[x]$	0	1	3	6	7	4	5	2

4.3.2 An Attack on PRINTcipher

One interesting property of the PRINTcipher s-box is that a one bit difference in the input causes a one bit difference in the same bit in the output with probability 2/8. That is, there exists exactly one pair for each one bit input difference resulting in a one bit output difference (at the same position). More precisely, denoting by $*$ an arbitrary value in $\mathbb{F}_2$, the following holds for the PRINTcipher s-box:

S(000) = 000 S(001) = 001	$\Leftrightarrow$	S(00*) = 00*
S(100) = 111 S(110) = 101	$\Leftrightarrow$	S(1*0) = 1*1
S(011) = 110 S(111) = 010	$\Leftrightarrow$	S(*11) = *10

In addition, there exists a subset of s-boxes such that (1) two output bits of those s-boxes map onto two input bits of the same s-boxes in the next round and (2) the round-dependent RC_i is not involved (see Figure 4.2).

Now consider an xor-key sk_1 of the form

Xor key = 01* *11 *** *** 01* *11 *** *** 01* *11 *** *** 01* *11 *** ***,

and a permutation key with the following restrictions:

Perm. key = 0* 11 ** ** 10 01 ** ** 11 *0 ** ** *0 11 ** **,

where again $*$ denotes an arbitrary value in $\mathbb{F}_2$. For those keys the following structural *iterative* one round property holds:

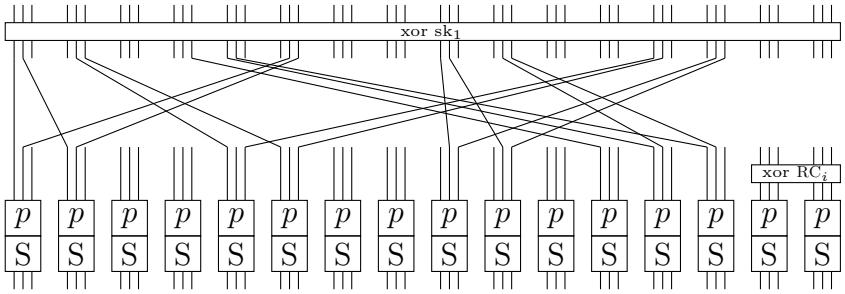


Figure 4.2: A subset of PRINTcipher-48 s-boxes mapping onto itself.

Start	00* *10 *** *** 00* *10 *** *** 00* *10 *** *** 00* *10 *** ***
Key xoring	01* *01 *** *** 01* *01 *** *** 01* *01 *** *** 01* *01 *** ***
Lin. layer	00* 11* *** *** 0*0 1*1 *** *** *00 *11 *** *** 00* 11* *** ***
RC	00* 11* *** *** 0*0 1*1 *** *** *00 *11 *** *** 00* 11* *** ***
Perm. layer	00* *11 *** *** 00* *11 *** *** 00* *11 *** *** 00* *11 *** ***
S-box layer	00* *10 *** *** 00* *10 *** *** 00* *10 *** *** 00* *10 *** ***

This property holds with probability one if both keys are of the above form. The fraction of those keys is $(1/2)^{16}$ for the XOR key and $(1/2)^{13}$ for the permutation key, meaning that the property is met for a fraction of $(1/2)^{29}$ of all keys. In other words, there exist 2^{51} weak keys of this form.

Thus, one can very efficiently check if a key of the above form is used by encrypting a few texts of the above form and check if the ciphertext is again of the same form. Given that the probability for false positives is $\approx 2^{-16}$, trial encrypting just a handful of selected plaintexts will uniquely identify such a weak key. If such a key is found, we do of course immediately have a distinguisher on PRINTcipher.

4.3.3 Invariant Subspace Description:

Let us briefly rephrase the attack in terms of an invariant subspace attack. For this we fix a permutation key of the above form. Remember that the inner state at the beginning and the end of each round was

$$\text{Start} = \texttt{00* *10 *** *** 00* *10 *** *** 00* *10 *** *** 00* *10 *** ***}.$$

This means that the relevant subspace $U \subset \mathbb{F}_2^{48}$ is defined by

$$U = \{\texttt{00* *00 *** *** 00* *00 *** *** 00* *00 *** *** 00* *00 *** ***}\}, \quad (4.1)$$

and that the affine subspace is defined by any fixed vector d of the form

$$d = \texttt{00* *10 *** *** 00* *10 *** *** 00* *10 *** *** 00* *10 *** ***}. \quad (4.2)$$

Then for any fixed vector c of the form

$$c = \texttt{01* *01 *** *** 01* *01 *** *** 01* *01 *** *** 01* *01 *** ***}, \quad (4.3)$$

and any xor-key $k \in (U + c + d)$, the round function does indeed map $U + d$ onto itself.

4.4 Other Attack Profiles

In the following we describe other sets of weak keys for PRINTcipher-48 and similar ones for PRINTcipher-96.

4.4.1 Other weak keys for PRINTcipher-48

As it turns out, there are some more invariant subspaces that also can be used for PRINTcipher-48. They are all of the form

`00* XXX *** 1*1 00* *10 *** *** 00* XXX *** 1*1 00* *10 *** ***,`

where an 'X' marks a bit position where the attacker has to make an arbitrary assignment. Note that each position can be filled independently of the others. Thus, we have 2^6 possible plaintexts that we can work with, each of which targets another class of weak keys.

For each such assignment, the cipher behaves as follows:

Start	(1)	`00* XXX *** 1*1 00* *10 *** *** 00* XXX *** 1*1 00* *10 *** ***`
Key xoring	(2)	`0X* X01 *** X*1 01* *0X *** *** 0X* 001 *** X*X 01* *0X *** ***`
Lin. layer	(3)	`00* XXX *** X*X 0*0 1*1 *** *** *00 XXX *** 10* 00* 11* *** ***`
RC	(4)	`00* XXX *** X*X 0*0 1*1 *** *** *00 XXX *** 10* 00* 11* *** ***`
Perm. layer	(5)	`00* XXX *** 1*0 00* *11 *** *** 00* XXX *** 1*0 00* *11 *** ***`
S-box layer	(6)	`00* XXX *** 1*1 00* *10 *** *** 00* XXX *** 1*1 00* *10 *** ***`

The behaviour is best understood by traversing the cipher in the inverse direction, i.e. by starting from the end and then finding the key bits that ensure that all fixed bits in line (1) match their counterparts in line (6).

Let us start with the output of the s-box, i.e. line (6), and let the bit positions marked by 'X' be arbitrarily and independently fixed to either 0 or 1. Then going backwards through the s-box uniquely determines the bits in line (5). We then use a permutation key of the form

Perm. Key = `0* ** ** (00 or 11) 10 01 ** ** 11 ** ** 10 0* 11 ** **`

to obtain line (4), noting that 2^{-13} of all permutation keys meet this property. We then apply round counter and linear layer to obtain line (2). Now note that line (2) contains 22 bits that are fixed and that have to match the corresponding bits in line (1). Thus, 22 key bits of the xoring key are determined, meaning that 2^{-22} of all xoring keys are suitable for the attack.

Summing up, for each of the 2^6 possible assignments to the bits marked by 'X' in line (1) or (6), a fraction of exactly 2^{-35} keys are weak, meaning that in total, we have found another fraction of 2^{-29} weak keys that can be attacked by the above technique.

4.4.2 Analysis of PRINTcipher-96

As it turns out, the same attack can also be applied to PRINTcipher-96. Again, there are two types of weak keys. The first type is based on 32 active bits and is met by a fraction of 2^{-59} of all keys. The second type is based on 44 active bits and has an additional 12 freely chosable input bits. Each of the resulting 2^{12} inputs targets a fraction of 2^{-71} keys, meaning that this group, too, contains a fraction of 2^{-59} weak keys in total. The active bits for these weak keys are given in Table 4.1.

Table 4.1: Subsets of active bits for PRINTcipher-96, grouped according to s-boxes

Subset 1	**Active input bits for linear layer:** (0 1) (4 5) (12 13) (16 17) (24 25) (28 29) (36 37) (40 41) (48 49) (52 53) (60 61) (64 65) (72 73) (76 77) (84 85) (88 89)
	Active output bits for linear layer: (0 2) (3 5) (12 13) (15 16) (25 26) (28 29) (36 38) (39 41) (48 49) (51 52) (61 62) (64 65) (72 74) (75 77) (84 85) (87 88)
Subset 2	**Active input bits for linear layer:** (0 1) (3 4 5) (9 11) (12 13) (16 17) (24 25) (27 28 29) (33 35) (36 37) (40 41) (48 49) (51 52 53) (57 59) (60 61) (64 65) (72 73) (75 76 77) (81 83) (84 85) (88 89)
	Active output bits for linear layer: (0 2) (3 4 5) (9 10) (12 13) (15 16) (25 26) (27 28 29) (33 35) (36 38) (39 41) (48 49) (51 52 53) (58 59) (61 62) (64 65) (72 74) (75 76 77) (81 82) (84 85) (87 88)

4.5 Countermeasuere Against the Attack

The above attack against PRINTcipher is a special case of the general attack described in the beginning of the section, since the subspace is described by simply fixing some of its bits. In theory, describing the subspace by a set of linear equations is possible, opening for a wide range of attacks. The full potential of this generalized attack is yet to be determined.

As for the special case used against PRINTcipher, it is relatively easy to protect the design against the attack. Note that the list of attack profiles by fixing bits given here is complete, and that all attack profiles fix two of the bits 39-41 (PRINTcipher-48) resp. 87-89 (PRINTcipher-96). Thus, it would suffice to spread the round counter over the last three s-boxes, e.g. by assigning two counter bits to each s-box. This would destroy the only attack profiles available, at no extra hardware cost.

We also analysed the block cipher NOEKEON, which was proposed by Daemen et al. in 2000 [113]. NOEKEON is a 16-round block cipher with a constant round key, making it a

particularly tempting target for the attack. However, as it turns out, the linear mixing layer of NOEKEON is much more resistant against the above type of attack. Here, the stronger round function (necessary for a cipher with only 16 rounds) works to the advantage of the cipher. As it turns out, even if there was no round counter involved in NOEKEON, the simple attack described above – i.e. where the subspace is defined by fixing certain bits – could not be applied. Whether or not the generalized attack has a better chance of succeeding remains yet to be determined.

4.6 Statistical Saturation Attacks and Multidimensional Linear Attacks

The attack on PRINTCIPHER discussed in Section 4.3 is clearly strongly related to statistical saturation attacks as described in [68]. In this section, after briefly recalling some of the principles of statistical saturation attacks, we elaborate on the details of this relation. Maybe the most interesting finding here is that for PRINTCIPHER there exist strongly biased linear approximations for any number of rounds, if the key is weak in the sense of the invariant subspace attack. This result follows using a link between statistical saturation attacks and multi-dimensional linear attacks (see [142]). Understanding these strongly biased linear approximations by studying the linear hulls directly is an interesting problem that we leave open for further investigation.

4.6.1 Necessary Background Information

4.6.1.1 Notations

The canonical inner product on $\mathbb{F}_2^n$ is denoted by $\langle \cdot, \cdot \rangle$, i.e.

$$\langle (a_0, \ldots, a_{n-1}), (b_0, \ldots, b_{n-1}) \rangle := \sum_{i=0}^{n-1} a_i b_i.$$

We note that all linear forms, i.e. all linear functions $l : \mathbb{F}_2^n \to \mathbb{F}_2$, can be described as $\ell(x) = \langle a, x \rangle$ for a suitable $a \in \mathbb{F}_2^n$. Given a (vectorial Boolean) function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ the *Fourier coefficient* of F at the pair $(a, b) \in \mathbb{F}_2^n \, \mathbb{F}_2^m$ is defined by

$$\widehat{F}(a, b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle b, F(x) \rangle + \langle a, x \rangle}.$$

The *bias* $\epsilon_F(a, b)$ of the linear approximation $\langle a, x \rangle$ of $\langle b, F(x) \rangle$ is defined as

$$\epsilon_F(a, b) := \frac{|\{x \mid \langle b, F(x) \rangle + \langle a, x \rangle = 0\}|}{2^n} - \frac{1}{2}.$$

The fundamental relation between the Fourier transformation of F and the bias of a linear approximation is given by

$$\epsilon_F(a, b) \quad = \quad \frac{\widehat{F}(a, b)}{2^{n+1}} \tag{4.4}$$

Given $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the value used to determine the complexity of both multidimensional linear attacks and statistical saturation attacks is the capacity of F given by

$$\mathrm{Cap}(F) = \sum_{z \in \mathbb{F}_2^m} \frac{(2^{-n} \cdot |\{x \in F_2^n \mid F(x) = z\}| - 2^{-m})^2}{2^{-m}}.$$

4.6.1.2 Statistical Saturation Attacks

Let us first briefly recall some concepts from statistical saturation attacks. We refer to [68] for details. Given an encryption function

$$e : \mathbb{F}_2^n \to \mathbb{F}_2^n,$$

statistical saturation attacks study the distribution of e when some of its input bits are fixed. Up to a fixed bijective linear transformation before and after the cipher, we can restrict ourselves without loss of generality to the case where one fixes the first r bits in the inputs and considers only the first t bits of the output[1]. Thus we write

$$\begin{aligned} e : \mathbb{F}_2^r \mathbb{F}_2^s &\to \mathbb{F}_2^t \mathbb{F}_2^u \\ e(y, x) &= \left(e^{(1)}(y, x), e^{(2)}(y, x)\right), \end{aligned}$$

where $r + s = t + u = n$ and $e^{(1)}(y, x) \in \mathbb{F}_2^t$, $e^{(2)}(y, x) \in \mathbb{F}_2^u$. For convenience we denote by h_y the restriction of e by fixing the first r bits to y and considering only the first t bits of the output, that is

$$\begin{aligned} h_y : \mathbb{F}_2^s &\to \mathbb{F}_2^t \\ h_y(x) &= e^{(1)}(y, x). \end{aligned}$$

In a statistical saturation attack one considers the capacity of h_y, and the attack complexity is usually a constant times $1/\mathrm{Cap}(h_y)$. Computing this capacity is difficult in general. However, when averaging over all possible fixings y the following has been proven in [142]:

Theorem 4.6.1. *The average capacity in statistical saturation attacks where the average is taken over all possible fixations y is given by*

$$\overline{\mathrm{Cap}(h_y)} = 2^{-r} \sum_{y \in \mathbb{F}_2^r} \mathrm{Cap}(h_y) = 2^{-2n} \sum_{\substack{a \in \mathbb{F}_2^r \{0\} \\ b \in \mathbb{F}_2^t \{0\}, b \neq 0}} (\widehat{e}(a, b))^2 \tag{4.5}$$

4.6.2 On the Choice of the Values of the Fixed Bits

We now focus on the case where $r = t$, that is the number of fixed bits is the same as the number of bits considered at the output.

[1] This differs slightly from the notation in [142]

Assume a cipher is vulnerable to an invariant subspace attack. As for statistical saturation attacks, up to a fixed bijective linear transformation before and after the cipher, we can assume that, for a weak key, the affine subspace of the form $\{d\}\,\mathbb{F}_2^s$ is mapped to an affine subspace of the form $\{d\}\,\mathbb{F}_2^s$. It then follows immediately that (for a weak key) the function of the restriction h_y for $y = d$ is a constant, more precisely

$$h_d(x) = e^{(1)}(d, x) = d.$$

For the special choice of the values of the fixed bits the capacity is maximal. Hence for a weak key this special fixing of the bits leads to an optimal statistical saturation attack. Note that Theorem 4.6.1 does not reveal the existence of such extreme cases, as it only considers the average capacity of the restrictions.

While in an invariant subspace attack, given the subspace, the choice of the coset is crucial, for statistical saturation attacks the fixed bits are usually assigned with random values. As the invariant subspace attack on PRINTCIPHER does not imply that PRINTCIPHER is in general vulnerable to a statistical saturation attack, it does not come as a surprise that the experiments in [140] did not reveal any weakness of PRINTCIPHER with respect to those attacks.

4.6.3 On the Existence of Highly Biased Approximations

Theorem 4.6.1 was used to compute the average capacity using the Fourier coefficients. However, for us, the reciprocal is of interest as it implies the following corollary.

Corollary 4.6.2. *Assume an n-bit block cipher E_k is vulnerable to an invariant subspace attack, that is there exist a subspace U, a constant d and keys k such that*

$$E_k(U + d) = U + d.$$

Then, for those keys, there exist linear approximations with a bias ϵ such that

$$\epsilon \geq 2^{\dim(U)-n-1} - 2^{2(\dim(U)-n)-1}.$$

Proof. With the notation as in Section 4.6.2, h_d is a constant function. Thus $\mathrm{Cap}(h_d) = 2^r - 1$ and furthermore

$$\sum_{y \in \mathbb{F}_2^r} \mathrm{Cap}(h_y) \geq \mathrm{Cap}(h_d) = 2^r - 1.$$

Considering Equation (4.5) it follows that

$$\sum_{\substack{a \in \mathbb{F}_2^r \{0\} \\ b \in \mathbb{F}_2^t \{0\}, b \neq 0}} (\widehat{e}(a, b))^2 \geq 2^{2n}(1 - 2^{-r})$$

Lower bounding the maximal value by the average (and recalling that $r = t$), we compute

$$\max_{a,b \neq 0} (\widehat{e}(a, b))^2 \quad \geq \quad 2^{-2r} \sum_{\substack{a \in \mathbb{F}_2^r \{0\} \\ b \in \mathbb{F}_2^t \{0\}, b \neq 0}} (\widehat{e}(a, b))^2 \geq 2^{2n-2r}(1 - 2^{-r})$$

Thus there exists at least one Fourier coefficient such that

$$|\widehat{e}(a,b)| \geq 2^{n-r}\sqrt{1-2^{-r}} \geq 2^{n-r} - 2^{n-2r}$$

Applying identity (4.4) and remembering that $r = n - \dim U$, the theorem follows. □ □

Clearly, this Theorem is only interesting for the case where $\dim(U) > n/2$ as the existence of the stated approximations otherwise is trivial. For the case of PRINTcipher-48 we summarize the findings below

Corollary 4.6.3. *Given a weak key for any round $r \leq 48$ there exists at least one linear approximation for* PRINTcipher-*48 with bias at least $2^{-17} - 2^{-33}$.*

4.7 Related Work

When it comes to attacks on PRINTcipher, one of the earliest results are the differential attack on 22-round PRINTcipher in [157]. The attack uses the full code book to investigate single-bit differentials to gain information about the bit permutation through the cipher. Then they find the r-th root of the permutation to get the single round permutation and then the key.

In [15], 29 rounds of PRINTcipher cipher was attacked using 4.54% or the key and 31 rounds using 0.36% of the keys. The method rely on using a combination of differential and linear cryptanalysis approaches. Linear attacks on PRINTcipher were used on 28 for 50% of the keys, and 29 rounds for 3.125% of the keys.

As already mentioned, This attack can be seen as a weak key variant of statistical saturation attacks [68, 69]. As the statistical saturation attack itself is a special case of partitioning cryptanalysis [99], so is our attack. Again, the main difference is that we make use of weak keys and for those keys the bias is maximal. Moreover, our attack can also be interpreted as an extreme case of a dynamic cube attack [3]. Here, the algebraic normal form of certain ciphertext bits becomes a constant when a weak key is used and certain message bits are fixed correctly.

The work in [11] explains the reason behind having a large correlations as an effect of the invariant subspace. They present the linear hull of the linear approximations with large correlations that lies in the orthogonal subspace with two different sums. These sums represents the linear trails with intermediate masks inside and outside the orthogonal subspace for at least one round. Then, they construct a correlation matrix for PRINTcipher and end up having the same submatrix for each round M^r. This is due to the fact that the submatrix of this correlation matrix does not converge to all zero matrix when raised to the power r. Hence, in the case of unique eigenvector with eigenvalue of norm 1 the M^r will converge to all non-zero constant. They stated that there is an equivalence between a submatrix of the correlation matrix that has an eigenvector with eigenvalue one and a round function with invariant subspace.

In [57], 64 classes of weak keys were found for PRINTcipher-48 beside the two classes mentioned in this chapter that contains $2^5 1$ keys. They have noted that the presented classes in our results contains $2^4 5$ joint weak keys. In addition to $115,669$ classes of weak keys for PRINTcipher-96

beside the two classes in this chapter. This achieved by finding an invariant subspace by finding the specific subsets in $\mathbb{Z}_n$. Then finding the permutation key with an invariant subspace. It alternatively, proposes to have a second approach for large number of keys where certain polytope in $\mathbb{Z}_n$ will be used to achieve using Mixed Integer Linear Programming (MILP)

The details of the presented results in this section will be left to the reader.

4.8 Conclusion

In this chapter, a new attack against iterative block ciphers named *invariant subspace attack* was presented and its validity was demonstrated by breaking PRINTCIPHER for a significant fraction of its keys. The presented *invariant subspace attack* verifies that there exist 2^{52} weak keys of the 2^{80} possible keys for PRINTCIPHER-48 and 2^{102} weak keys of the 2^{160} possible keys of PRINTCIPHER-96. Furthermore, the relationship between the *invariant subspace attack* and other classes of attacks as in multi-dimensional attack linear attack and statistical saturation attack was explored.

Moreover, for PRINTCIPHER there are strongly biased linear approximations for any number of rounds, if a weak key is chosen with an absolute correlation of at least 2^{-16} . For example, there is at least one linear approxiamtion for PRINTCIPHER-48 with bias at least 2^{-17}.

Chapter 5

Cryptanalysis of SIMON

Recently, the U.S National Security Agency has published the specifications of two families of lightweight block ciphers, SIMON and SPECK, on ePrint [26]. The ciphers are developed with optimization towards both hardware and software in mind. While the specification paper discusses design requirements and performance of the presented lightweight ciphers thoroughly, no security assessment was given.

This chapter is a move towards filling that cryptanalysis gap for the SIMON family of ciphers. We present a series of observations on the presented construction that, in some cases, yield attacks, while in other cases may provide basis of further analysis by the cryptographic community.

Specifically, we obtain attacks using classical- as well as truncated differentials. In the former case, we show how the smallest version of SIMON, Simon32/64, exhibits a strong differential effect. In addition to that we investigate the security of SIMON against different variants of linear cryptanalysis techniques, i.e. classical and and linear hulls. We present a connection between linear- and differential characteristics as well as differentials and linear hulls in SIMON. We employ it to adapt the current known results on differential cryptanalysis of SIMON into the linear setting. In addition to finding a linear approximation with a single characteristic, we show the effect of the linear hulls in SIMON by finding better approximations that enable us to improve the previous results.

Our best linear cryptanalysis employs average squared correlation of the linear hull of SIMON based on correlation matrices. The result covers 21 out of 32 rounds of SIMON32/64 with time and data complexity $2^{54.56}$ and $2^{30.56}$ respectively. We have implemented our attacks for small scale variants of SIMON and our experiments confirm the theoretical biases and correlation presented in this work. So far, our results are the best known with respect to linear cryptanalysis for any variant of SIMON.

5.1 Our Contribution

This paper is a move toward providing an initial-in some respects- cryptanalytic research and results for the SIMON family of ciphers. A series of observations on the presented SIMON construction are made, some utilized into attacks, while others may provide grounds for more improved analysis. The main contributions of this chapter are the following:

- For differential cryptanalysis, we have determined iterative differentials for Simon32/64, and general differentials for all variants of SIMON, that yield differential attacks on

reduced versions with at least half the total rounds of the cipher in all cases. An interesting observation in Section 5.3.4 is that Simon32/64 exhibits a strong differential effect. This suggests that bounding the expected differential probability (EDP) by the expected maximum characteristic probability is not well-founded in this case.

- Furthermore, we considered using truncated differentials to construct impossible differentials over a number of rounds, which yielded a distinguisher on reduced versions of most of the cipher variants, however it can not be to launch a practical attack.
- We analyze the security of SIMON against variants of linear cryptanalysis.
  - Using Algorithm 2 of Matsui, we extend attack of [16] to 17, 20, 23, 34 and 43 rounds for the respective block sizes of 32, 48, 64, 96 and 128 bits respectively. We also present a generalized algorithm based on the connection given by Alizadeh et al. in [16] to convert any given differential characteristic to a linear characteristic for SIMON.
  - We also establish a connection between capacity of a linear hull and differential for SIMON and use the known results on differential cryptanalysis of SIMON to attack 21, 21, 29, 36, and 50 rounds of the respective block/key sizes of 32/64, 48/96, 64/128, 96/144, and 128/256 bits. Our focus on improving the linear cryptanalysis results on SIMON by estimating the average squared correlation of linear hulls. We show the linear hull effect by finding better approximations that enable us have better data complexity. A brief summary of our results are presented in Table 5.16.

The presented results are published in total of three papers: one conference paper in RFIDSEC2014 [16] and two e-Print papers ([18], [158]) papers which one of them was peer-reviewed [18] and the other is under submission for IEEE Transactions on Information Theory.

5.2 General Description of SIMON

5.2.1 Structure and Variants

SIMON is a family of lightweight block ciphers designed by the NSA with the aim of providing a cipher of an optimal hardware performance [26]. The design of SIMON is a classical Feistel scheme, operating on two n-bit halves in each round, thus the general round block size is $2n$ bits. In the remainder of this paper, we use n to refer to half the block size of the cipher, i.e. the size of the left and right branches, respectively. Each round of SIMON applies a non-linear, non-bijective hence non-invertible function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ to the left half of the state. The output of F is added using XOR to the right half along with a round key, and the two halves are swapped. The function F is defined as

$$F(x) = ((x \lll 8) \odot (x \lll 1)) \oplus (x \lll 2)$$

where $x \lll j$ denotes left rotation of x by j positions and $\odot$ is binary AND. A single round of SIMON is depicted in Figure 5.1.

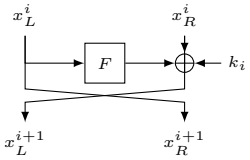


Figure 5.1: The SIMON round function

Variants of SIMON exist for different parameters of key size, block size and number of rounds. The name of each SIMON variant with its parameters are presented in Table 5.1.

Table 5.1: Members of the SIMON family with their parameters

Cipher	Block size $2n$	Key words m	Key size mn	Rounds T	Index to z j
Simon32/64	32	4	64	32	0
Simon48/72	48	3	72	36	0
Simon48/96	48	4	96	36	1
Simon64/96	64	3	96	42	2
Simon64/128	64	4	128	44	3
Simon96/92	96	2	92	52	2
Simon96/144	96	3	144	54	3
Simon128/128	128	2	128	68	2
Simon128/192	128	3	192	69	3
Simon128/256	128	4	256	72	4

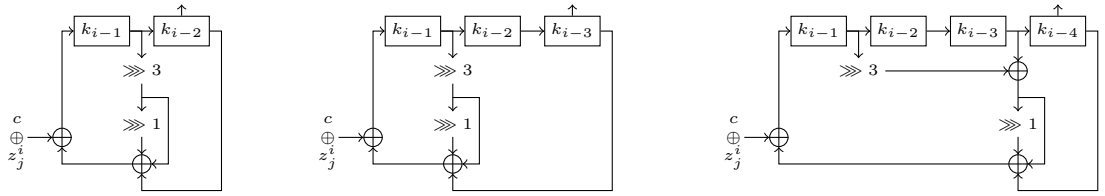
5.2.2 Key Schedule

The key schedule of SIMON is described as a function that will operate on two, three or four n-bit word registers, depending on the size of the master key. It performs two rotations to the right by $x \ggg 3$ and $x \ggg 1$ and XOR the results together with a fixed constant c and five constant sequences z_j^i which are version-dependent. These constant sequences are obtained by using three 55 matrices over $\mathbb{F}_2$, and a linear feedback shift register where the first two are of period 31 and the last three are of period 62. The specification rationalizes the use of these constants as a mean of eliminating sliding properties and circular shift symmetries between the different rounds keys. Furthermore, they are used to provide cryptographic separation between different variants of SIMON that have the same block size, but with different key sizes.

Figure 5.2 describes the general function of SIMON key scheduling. The m master key words, each of n bits where $m \in \{2, 3, 4\}$, are used at the first iterations of key scheduling, and hence the first mn round key bits equal the master key.

Depending on m, the key schedule varies slightly, c.f. Figure 5.2. The value c is a constant equal to $(2^n - 1) \oplus 3$, i.e. a string of $n - 2$ ones and two zeroes on the least significant two

bits. The value z_j^i is the ith bit (from most significant to least significant, where i is computed modulo n) of z_j, where z_j is from Table 5.2 and j is a parameter of the cipher, c.f. Table 5.1.



(a) $m = 2$ key words (b) $m = 3$ key words (c) $m = 4$ key words

Figure 5.2: The SIMON key schedule for cases $m \in \{2, 3, 4\}$. The computation on round key k_i depends on k_{i-1} and k_{i-m}, and also k_{i-m+1} in the case of $m = 4$.

Table 5.2: The z_j vectors used in the SIMON key schedule

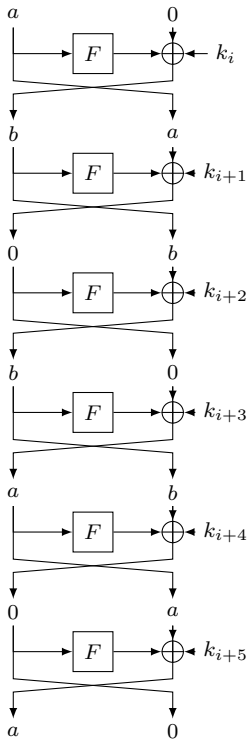
j	z_j
0	11111010001001010110000111001101111101000100101011000011100110
1	10001110111110010011000010110101000111011111001001100001011010
2	10101111011100000011010010011000101000010001111110010110110011
3	11011011101011000110010111100000010010001010011100110100001111
4	11010001111001101011011000100000010111000011001010010011101111

5.3 Differential Cryptanalysis

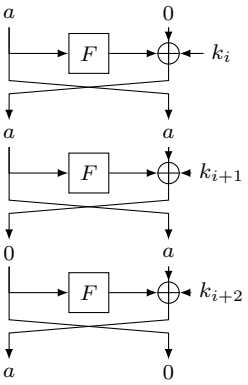
Let us first recall the motivation behind classical differential cryptanalysis. The key goal of the analysis is to trace the input/output difference propagation through the cipher structure, for a specific number of rounds, and detect the non-random behaviour exhibited in the final output, with a certain success probability. The differential property can be utilized to recover the (parts of) a sub-key, typically the first or the last, in a reduced r-round version of the cipher. Several chosen plaintext pairs are used, in a combination with trying all candidates for the sub-key under attack, and the expected net result is that the correct sub-key is suggested more frequently than the wrong ones, allowing the attacker to tell which is correct.

First, we discuss iterated differentials, i.e. differentials using the same input/output difference. For the SIMON family of block ciphers, we are interested in one of two properties of F for constructing the iterated differentials. Firstly, we consider pairs of n-bit differences (a, b), for which the combined probability $\Pr(a \to b) \cdot \Pr(b \to a)$ is maximized. Here, $\Pr(a \to b)$ denotes the probability that a difference a goes to a difference b over the function F, taken over all inputs. We refer to this as a *type-1* iterated characteristic. Secondly, we may consider looking for a characteristic using a single difference a, for which $\Pr(a \to a)$ is maximized. We refer to this as a *type-2* iterated characteristic.

For type-1 characteristics, we can construct a 6-round iterative characteristic, while for type-2 we get a similar 3-round characteristic. Both are shown in Figure 5.3.



(a) A 6-round iterated characteristic using two input/output relations



(b) A 3-round iterated characteristic using a single input/output relation

Figure 5.3: Type-1 and type-2 iterated differential characteristics for SIMON

5.3.1 Difference Distribution Table

For block ciphers using a Substitution Permutation Network (SPN) design structure, a common method for obtaining a non-linearity is to use parallel applications of small b-bit S-Boxes. In this case, the output difference on b consecutive bits depends solely on the input difference on the corresponding b bits. As such, a difference distribution table for the whole non-linear component can be derived directly from the corresponding table for the S-Box. For the function F used in SIMON there is no S-Box, and in general a single bit of the output difference Δy depends on 2 bits of the input x and 3 bits of the input difference Δx, by the relation

$$\Delta y_i = x_{i-1} \cdot \Delta x_{i-8} \oplus \Delta x_{i-1} \cdot x_{i-8} \oplus \Delta x_{i-1} \cdot \Delta x_{i-8} \oplus \Delta x_{i-2},$$

where all indices are computed modulo n. As such, constructing the difference distribution table requires $O(2^{2n})$ memory and has the same complexity. Thus, for $n = 16$, this requires 8 GB of memory using an unsigned 16-bit data type for the entries.

For $n = 16$, we construct the table exhaustively and determine the best pairs (a, b) as above for the type-1 characteristic. The best pairs (a, b) yield a probability

$$\begin{aligned} \Pr(a \to b) \cdot \Pr(b \to a) &= \frac{256}{2^{16}} \cdot \frac{2048}{2^{16}} \\ &= 2^{-13}. \end{aligned}$$

If we square this probability, we find that 2^{-26} is the probability of the 6-round type-1 characteristic shown in Figure 5.3, using those (a, b) pairs. The pairs are listed in Table 5.3.

Table 5.3: Best possible (a, b) pairs for type-1 differential characteristics obtained for Simon32/64

a	b	$\log_2(\Pr(a \to b))$	$\log_2(\Pr(b \to a))$
0045	051e	-5	-8
008a	0a3c	-5	-8
0114	1478	-5	-8
0228	28f0	-5	-8
028f	8022	-8	-5
0450	51e0	-5	-8
08a0	a3c0	-5	-8
1140	4781	-5	-8
1401	7814	-5	-8
1e05	4500	-8	-5
2280	8f02	-5	-8
2802	f028	-5	-8
3c0a	8a00	-8	-5
4011	8147	-5	-8
5004	e051	-5	-8
a008	c0a3	-5	-8

As the type-1 characteristic uses only the difference a in the input/output, we may instead think of it as a 6-round differential, where the difference b can take on any possible value. As such, we can search for the best difference a, s.t.

$$\sum_{b \in \mathbb{F}_2^n} \Pr(a \to b) \cdot \Pr(b \to a)$$

is maximized. Doing so, we find that for $n = 16$ there are four best such differences, $a \in \{1111, 2222, 4444, 8888\}$.

These represent 3-round differentials of probability $2^{-11.19}$, where we do not care about the intermediate differences, i.e. the type-2 characteristics considered as differentials. When putting two such differentials together, we get a 6-round differential of probability at least $2^{-2 \cdot 11.19} = 2^{-22.38}$, which is similar to the type-1 characteristic considered as a differential, except that after 3 rounds we know the difference is $(a \parallel 0)$.

For $n > 16$, the memory and complexity required renders constructing the difference distribution table infeasible. However, a method by Dinur et al. which was presented at the Eurocrypt 2013 rump session [82] computes the diagonal of the difference distribution table using $O(2^n)$ memory and complexity. Thus, we can use this method to obtain results for $n = 24$ as well. The diagonal entries of the difference distribution table represent the iterative characteristics $a \to a$.

The algorithm uses a hash table M which maps values $x \oplus F(x)$ to a list holding the x values giving this difference. M is constructed by iterating over all $x \in \mathbb{F}_2^n$. After this, any pair of distinct x, x' in the list associated with the same key in M, are values s.t. $x \oplus F(x) = x' \oplus F(x')$, or in other words, $\Delta = x \oplus x'$ is the diagonal entry under consideration. However, to compute the actual differential probability, we must again iterate over all $x \in \mathbb{F}_2^n$ and check how many times $F(x) \oplus F(x \oplus \Delta) = \Delta$.

For $n = 16$ and $n = 24$, we obtain a list of best diagonal differential probabilities, presented in Table 5.4.

Table 5.4: Best diagonal entries of the difference distribution table for $n \in \{16, 24\}$

n	p	Differences
16	2^{-8}	`5555, aaaa, ac0e, 1d58, ab03, 581d, 3ab0, 6075, 5607, 0eac, b03a, 7560, c0ea, 03ab, eac0, 81d5, 0756, d581`
24	2^{-12}	`555555, aaaaaa, 0e22ac, 1c4558, 388ab0, 711560, c45581, e22ac0, 88ab03, 115607, 22ac0e, 45581c, ab0388, b0388a, 560711, 8ab038...`

It is evident from Table 5.4 that already for $n = 16$, $\Pr(\Delta \to \Delta)$, for some difference Δ, is very low, and will not lead to any good differential characteristic using this method. The table *suggests* that the best probability for a diagonal entry is $2^{-n/2}$. Thus, the probability paid for such characteristic would be too low, even for two iterations of the type-2 characteristic, as the number of plaintext pairs needed for the attack would exceed the possible number of plaintext pairs, 2^{2n}.

5.3.2 Input/Output Differences over F

For SIMON, consider an n-bit input difference $\alpha = x \oplus x'$ to F of Hamming weight one. As the $\oplus$ operation is invariant with respect to rotation, say w.l.o.g. that $\alpha = (0 \cdots 01)$. Recall that $F(x)$ left rotates x by eight and one positions respectively, applies binary AND to those two, and to the result of that XORs the left rotation of x by two positions. Due to the rotation by two and the XOR, the output difference $F(x) \oplus F(x')$ will, for this particular α, have a '1' on position 2. Also, on positions 1 and 8. There *may* be a '1' in the output difference (in fact each case occurs, on both bits independently, with probability $\frac{1}{2}$). As the $\odot$ operation is non-linear with respect to differences, this depends on the actual inputs x and x'. We may describe the output difference in truncated form as $(0 \cdots 0 * 000001 * 0)$. Here, an asterisk denotes an unknown bit.

This approach of determining a truncated mask captures all possible output differences can be generalized to arbitrary input differences, and each time we put an asterisk on a position we lose certainty about that particular bit of the output difference. Note, that this also provides

a means of determining all possible output difference, given some input difference, which in general is very useful for differential analysis. We will use this observation in the following section, and when we consider impossible differentials in Section 5.4.

5.3.3 Branch-and-Bound Approach to Differentials

Given a way of determining the possible output differences, along with their probabilities, when using a fixed input difference α, one can think of a tree where each difference at reach round spawns several possible output differences.

Besides fixing an input difference α, we fix a number of rounds to r for which we search for differentials. Starting with α, we progress in a depth-first manner, searching through characteristics until we reach round r. At that point, we add the characteristic probability to the output difference β in a lookup table. At the same time, we keep running score of the best seen output difference, for the fixed α, in terms of differential probability.

Using this approach gives us the best results on differential probabilities. Naturally, one can not hope to exhaustively try all input differences and still look through much of the tree. To that end, we maintain an array containing the best characteristic probability seen, for each level of tree, corresponding to each number of rounds $1, \ldots, r$. We bound the search at round i by allowing it only to go to round $i+1$ if the computed characteristic probability for level $i+1$ is within some fraction away from the best observed probability, which is stored in the array. Otherwise, we cut off that part of the tree and backtrack to the previous round. The constant fraction used in the bounding, giving the best results, is determined experimentally for each variant of SIMON. Note, that this method of cutting off sub-trees helps keep the Hamming weight of the differences low. Furthermore, we considered only input differences of low Hamming weight, as these intuitively have less possible output differences in the beginning, which are also of low Hamming weight.

As such, we can not claim to have found the best differentials for any of the variants, but our results certainly do provide lower bounds. A summary of the attack parameters and complexities can be found in Table 5.9.

5.3.4 Differential Effect

Using the branch-and-bound method described in Section 5.3.3, we are able, due to the small block size of Simon32/64, for a given number of rounds of the cipher to determine lower bounds on the *Expected Differential Probability* (EDP), which is defined in the following way, c.f. [74, 77]

$$\mathrm{EDP}(\alpha, \beta) = 2^{-n} \sum_{k \in \mathbb{F}_2^n} \mathrm{DP}_k(\alpha, \beta), \tag{5.1}$$

where $\mathrm{DP}_k(\alpha, \beta)$ is the differential probability for input difference α and output difference β using key k. Due to the computational complexity involved, our method is not directly applicable to test for the differential effect for larger block sizes.

The 12-round differential leading to our 16-round differential attack on Simon32/64, as described in Table 5.9, is

$$\alpha \to \beta = (0001 \parallel 0000) \to (0100 \parallel 0000),$$

for which we found that $\mathrm{EDP}(\alpha, \beta) > 2^{-29.481}$. The reason that the bound is not tight is twofold:

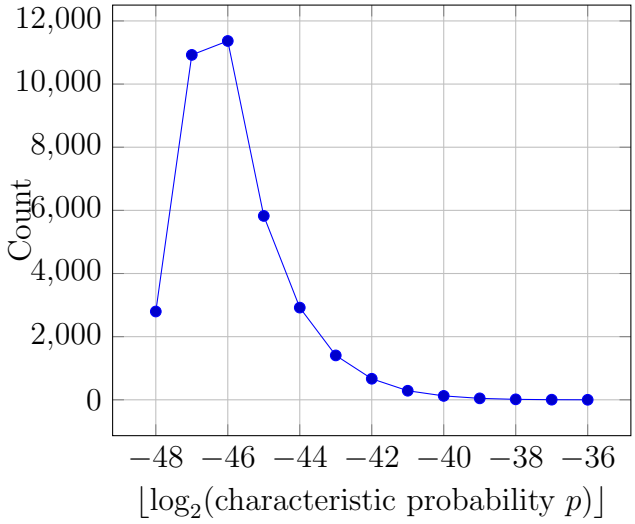
1. Firstly, due to the pruning of branches during the search, we never consider a large portion of characteristics belonging to some differential
2. Secondly, the search was, in some cases, stopped before considering all characteristics, even when using the pruning as just described, due to time limits.

An interesting question we are able to answer using the presented search method, for this small version of SIMON, is how strong the *differential effect* is. That is, we can determine if the EDP is due to the contribution of a few (or even a single) characteristics of high probability, or rather is the result of clustering of many characteristics of lower probability. For the differential $\alpha \to \beta$ of Equation (5.1), we keep track of the number of characteristics of probability $]p; 2p]$ in this differential by mapping $\lfloor \log_2 p \rfloor$ to a counter. We note that the search, and hence the characteristic counting, is stopped at the same point as for differential search, i.e. when obtain the bound $\mathrm{EDP}(\alpha, \beta) > 2^{-29.481}$.

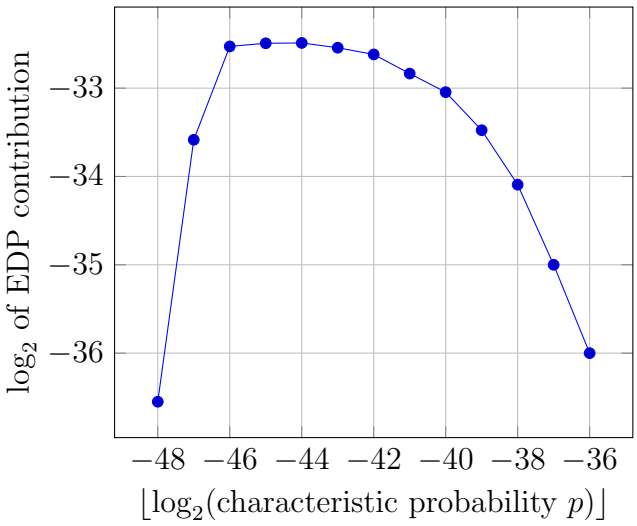
The resulting distribution of the number of characteristics and their probabilities are shown in Figure 5.4a. Figure 5.4b shows a division of the characteristics of probability $]p; 2p]$ on the first axis, and their total contribution to the EDP as the plotted value.

Figure 5.4a shows a low frequency of characteristics of probability 2^{-43} to 2^{-36}. In fact, we find just one characteristic of $\lfloor \log_2 p \rfloor = -36$ and four characteristics of $\lfloor \log_2 p \rfloor = -37$. While these few characteristics do provide an accumulated probability of $\approx 2^{-36} + 4 \cdot 2^{-37} \approx 2^{-34.42}$, the majority of the $\mathrm{EDP}(\alpha, \beta) > 2^{-29.481}$, is due to the vast number of characteristics of probability p s.t. $\lfloor \log_2 p \rfloor \in [-47; -39]$. Note that there is only one characteristic of probability 2^{-36}, which is a factor of $\approx 2^{6.5}$ from the bound on $\mathrm{EDP}(\alpha, \beta)$. This might give us an indication, that the theoretical bound on the EDP, chosen initially by the designers, is based on a provable bound on the characteristic probability, which is close to the 2^{-36} for 12 rounds, as seen above. In Figure 5.5, the same experiment is performed. However, characteristic probability frequencies for *all* differentials of $\mathrm{EDP}(\alpha, \beta) > 2^{-33}$ that we observe during our search, are collected. A total of 53 differentials were found, and in Figure 5.5, we clearly see the same large differential effect for all 53 cases.

Based on this observation, we conclude that, at least for Simon32/64, there is a prominent clustering of characteristics of lower probability, i.e. a strong differential effect. This might lead to a better understanding of the constraints imposed by the designers of SIMON, especially for smaller block sizes, when considering security bounds against certain attacks such as a differential attack. As mentioned, due to the computational complexity involved, we were not able to verify the observation on larger block sizes using our method. Whether this is so, poses an interesting problem for further research in this direction.



(a) Number of characteristics with a given characteristic probability p



(b) Total contribution to the EDP by characteristics of probability in $]p; 2p]$

Figure 5.4: Account of the number of characteristics of a certain probability p (left) and their accumulated probability (right). The first axis is determined as $\lfloor \log_2 p \rfloor$.

5.3.5 Generic Extension by Two Rounds on Top

Consider an $(r-2)$-round differential property, where the desired input difference is of the form $(\alpha \parallel 0)$, i.e. an arbitrary non-zero difference on the left half of the input, and a zero difference on the right half.

As the difference is zero on the right half of the input, the corresponding input difference to F in the previous round is zero, and consequently the output difference of F is too. As such, we can extend the $(r-2)$-round property to an $(r-1)$-round property by using the input difference $(0 \parallel \alpha)$ instead.

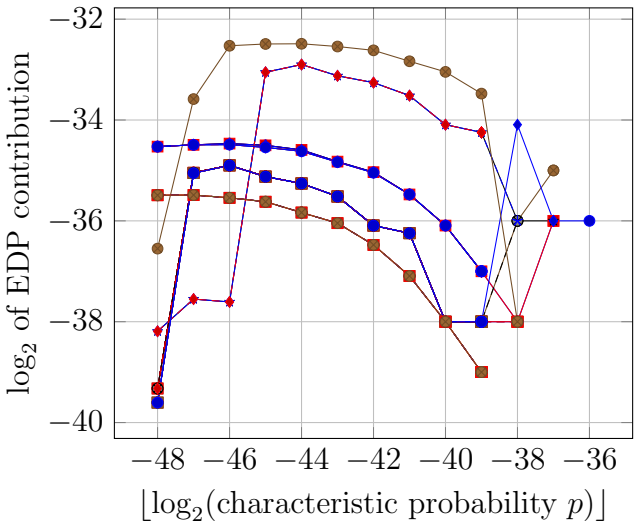


Figure 5.5: Total contribution to the EDP by characteristics of probability in $]p; 2p]$, for every 12-round Simon32/64 differential found with $\mathrm{EDP}(\alpha, \beta) > 2^{-33}$. Each plot represents a single differential. Note that the plots for some differentials overlap, due to identical counts for the characteristic occurences.

Moreover, if we choose a plaintext $(x \parallel y)$, and set $x' = x \oplus \alpha$, then we will suffer an overhead of two applications of F. As a result, we determine the second plaintext $(x' \parallel y') = (x \oplus \alpha \parallel y \oplus F(x) \oplus F(x \oplus \alpha))$, such that the difference after one round becomes $(0 \parallel \alpha)$. Thus, after two rounds the difference is $(\alpha \parallel 0)$. This extends the $(r-2)$-round property to an r-round property without reducing the differential probability, but with the overhead of just two applications of F.

5.3.6 Key Recovery

When using a differential for key recovery, one would normally attack a reduced r-round version of the cipher using an $(r-1)$-round differential. However, as the round key addition is performed after the application of F in each round for SIMON, we will in fact do key recovery on an r-round version of SIMON by using an $(r-2)$-round differential. We refer to Figure 5.6 in our explanation of the key recovery.

The key recovery works as follows. We assume that the output difference of the $(r-2)$-round differential is $(\alpha \parallel 0)$. Furthermore, let an output ciphertext pair be $(c_L \parallel c_R)$ and $(c'_L \parallel c'_R)$, for which the corresponding input plaintext pair have a chosen difference dictated by the differential. We initialize a counter for each possible key guess v to zeroes.

As we can compute F, we may determine

$$u_R \oplus u'_R = F(c_R) \oplus F(c'_R) \oplus c_L \oplus c'_L,$$

and check if this difference matches the difference α dictated by the differential. If this is the case, then the plaintext/ciphertext pair is assumed to follow the $(r-2)$-round differential.

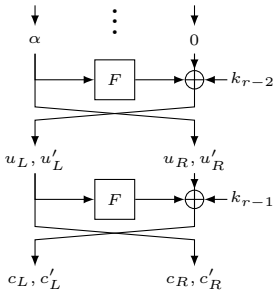


Figure 5.6: Differential Key Recovery Attack on SIMON

By trying all possible values v for the last round key, we may partially decrypt to obtain the actual pairs $(u_L \parallel u_R), (u'_L \parallel u'_R)$. Again, as we can evaluate F, we can check if

$$F(u_R) \oplus F(u'_R) \oplus u_L \oplus u'_L$$

equals zero. If this is the case, then the current guess for v was considered a candidate, and a counter for the key guess v is incremented.

The process above is repeated with about $\frac{c}{p}$ chosen plaintext pairs, for some small constant c, where p is the probability of the $(r-2)$-round differential. In the end, a ranking of key candidates by their counter values provides the attacker with the most probable key guesses for the attacked last round key.

The general complexity of the differential key recovery attacks can be expressed in terms of the following:

- Data complexity which can be defined as the number of chosen plaintexts used in the attack
- Time complexity is determined as the work effort, spent in partially decrypting the last round(s), in terms of *encryption queries*, i.e. the equivalence of r rounds of encryption
- Memory used on average for given time complexity

The data complexity of the classical differential attack can be expressed as $\frac{c}{P}$, where P is the differential probability for $r-1$ rounds, and c is a small constant. For the presented attack, it will be $2^{29.481}$ chosen pairs for 16 rounds of Simon32/64, as shown in Table 5.9. As for time complexity, it is defined by the number of total number of encryption queries achieved for all filtered pairs, using all possible key values:

$$\frac{c}{P}\gamma 2^k \frac{2}{r},$$

where r is the number of rounds, k is the number of key bits to be guessed, which are equal to n for SIMON, and γ is the probability that a pair survives the filtering which is 2^{-n}. This will yield a time complexity of $\frac{2c}{rP}$ encryption query equivalents for SIMON variants. As for the memory needed for the key recovery attack in the presented cases, it will be the number of key guesses which is 2^n words of memory. Refer to Chapter 3 for more information on data complexity of a differential attack. Summary of the obtained results on SIMON variants when it comes to differential cryptanalysis is presented in the Table 5.9

Table 5.5: Summary of our classical differential cryptanalytic results on SIMON.

Cipher	Rounds		Data	Memory	Time
	Total	Attacked			
Simon32/64	32	16	$2^{29.5}$	2^{16}	$2^{26.5}$
Simon48/72	36	18	$2^{46.4}$	2^{24}	$2^{43.3}$
Simon48/96	36	18	$2^{46.4}$	2^{24}	$2^{43.3}$
Simon64/96	42	24	$2^{62.0}$	2^{32}	$2^{58.4}$
Simon64/128	44	24	$2^{62.0}$	2^{32}	$2^{58.4}$
Simon96/92	52	29	$2^{87.5}$	2^{48}	$2^{83.7}$
Simon96/144	54	29	$2^{87.5}$	2^{48}	$2^{83.7}$
Simon128/128	68	40	$2^{124.8}$	2^{64}	$2^{120.5}$
Simon128/192	69	40	$2^{124.8}$	2^{64}	$2^{120.5}$
Simon128/256	72	40	$2^{124.8}$	2^{64}	$2^{120.5}$

5.4 Impossible Differential Cryptanalysis

Let us start by recalling what is introduced in Chapter 3 when it comes to impossible differential cryptanalysis. The approach combines two certain properties (two differentials with probability 1), one in the forward direction and one in the backward direction, and uses a resulting conflict when both directions are joined. This miss-in-the-middle approach is used to obtain an impossibility result. This can be utilized in a chosen-plaintext attack by requesting encryptions of plaintext pairs with a fixed difference, guessing key material and checking for the impossibility property to discard wrong guesses. In our case, the forward and backward differentials are truncated.

Some impossible differentials rely on the round function F being a permutation, a prominent example being the general 5-round property on Feistel schemes presented in [126]. However, the F function of SIMON is not a bijection, and indeed the impossible differentials we present in the following do not rely on it being so.

In Section 5.3.2, we saw how one can determine the possible output differences of the F function of SIMON, using a fixed input difference, in the sense that we can determine the truncated output difference. We also saw, that all possible output differences are equiprobable. We are interested in investigating for how many rounds a particular input difference can go before we are uncertain about all output difference bits, i.e. before we have asterisks on all positions. Intuitively, using an input difference of Hamming weight one will be the best approach, as each active bit in the input difference gives rise to 1, 2 or 3 active bits in the output difference, ignoring the possibility of cancellations, which is less predictable. For $n \in \{16, 24, 32\}$, we exhaustively tried all possible input differences and saw that this was indeed the case. For $n = 16$ and $n = 32$, there was another pattern of Hamming weight two, namely $(0 \cdots 00101)$ and any rotation of it, that covered equally many rounds in one direction. However, as there was no occurrence of both 0'es and 1's in the last truncated difference, the resulting impossible differential would cover less rounds than when using a Hamming weight one input difference.

Table 5.6 shows how the truncated differences progress over the rounds of SIMON for some

Table 5.6: Truncated differential pattern propagation for SIMON using word sizes $n \in \{16, 24, 32\}$, with an input difference $(0 \cdots 01 \parallel 0 \cdots 0)$

Rounds	32-bit block Left	Right
0	0000000000000001	0000000000000000
1	0000000*000001*0	0000000000000001
2	00000**00001**0*	0000000*000001*0
3	000***0*01*****0	00000**00001**0*
4	0******1******0*	000***0*01*****0
5	****************	0******1******0*

(a) For $n = 16$

Rounds	48-bit block Left	Right
0	000000000000000000000001	000000000000000000000000
1	000000000000000*000001*0	000000000000000000000001
2	0000000*00000**00001**01	0000000000000000*000001*0
3	00000**0000***0*01***0**	0000000*00000**00001**01
4	000***0*0**************1	00000**0000***0*01***0**
5	0**********************	000***0*0**************1
6	**********************	0**********************

(b) For $n = 24$

Rounds	64-bit block Left	Right
0	00000000000000000000000000000001	00000000000000000000000000000000
1	00000000000000000000000*000001*0	00000000000000000000000000000001
2	000000000000000*00000**00001**01	00000000000000000000000*000001*0
3	0000000*00000**0000***0*01***0*0	0000000000000000*00000**00001**01
4	00000**0000***0*0******1******0*	0000000*00000**0000***0*01***0*0
5	000***0*0********************0	00000**0000***0*0******1******0*
6	0****************************0*	000***0*0********************0
7	********************************	0****************************0*

(c) For $n = 32$

block sizes. We refer to Appendix **??** for the rest of the cases. All progressions use the same input difference $(0 \cdots 01 \parallel 0 \cdots 0)$. Other Hamming weight one input differences would yield a progression of truncated differences that are rotated correspondingly.

Taking the $n = 16$ case as an example, we see that after 5 rounds of SIMON, we have with probability 1 the truncated output difference

$$(* * * * * * * * * * * * * * * * \parallel 0 * * * * * * 1 * * * * * * 0 *).$$

By left rotating this right truncated difference by 7 or 9 positions, one of the 0's will be shifted to the position of the 1. Due to the symmetry of decryption and encryption of the Feistel scheme, we find that this provides us with two impossibility properties:

$$\Pr\left((0001 \parallel 0000) \rightarrow (0001 \lll 7 \parallel 0000)\right) = 0 \qquad \text{and}$$
$$\Pr\left((0001 \parallel 0000) \rightarrow (0001 \lll 9 \parallel 0000)\right) = 0,$$

where the impossible differential is over 10 rounds of SIMON. With this, we find two impossibility properties for each input difference of Hamming weight one, i.e. $2n$ in total. This property for the rotation by $q = 7$ is depicted in Figure **??** of Appendix **??**. In the further description of the attack, we denote by Q the set of indices for such rotations of the output difference, relative to the input difference, and hence $|Q|$ is the number of impossible differentials using one input difference. For example, for Simon32/64, $Q = \{7, 9\}$.

Note that the attack described so far uses an input difference of the form $(\alpha \parallel 0)$. Thus, the impossible differentials described in this section can trivially be extended by two rounds on top of probability 1, as described in Section 5.3.5, yielding an extra 2 rounds attacked.

Referring to Table 5.6, we see that for other values of n, we do not have both a 0 and 1 in the last truncated difference. Thus, we can not use this for obtaining an impossibility property, because we need to make a 0 overlap with a 1. We can, however, trace back to the last

round where the truncated output difference on the right half contains a 1, and match this up with the last truncated output difference containing a 0. This sacrifice means the impossible differential covers less rounds.

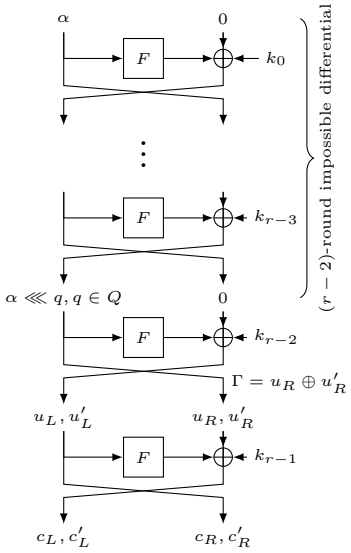


Figure 5.7: Key recovery attack with impossible differentials on SIMON

5.4.1 Key Recovery

As it was described for key recovery using the standard differentials, we again encrypt for two rounds more than the property covers. Consider a pair of output ciphertexts $(c_L \parallel c_R)$ and $(c'_L \parallel c'_R)$. The first filter in the recovery we can apply, is to test if

$$\Gamma := F(c_R) \oplus F(c'_R) \oplus c_L \oplus c'_L \tag{5.2}$$

equals the right half of one of the $|Q|$ impossible differentials, i.e. if it equals some $\alpha \lll q, q \in Q$.

If it does, we try all values v of the last round key and partially decrypt for one round to obtain the 1-round decrypted pair $(u_L \parallel u_R)$ and $(u'_L \parallel u'_R)$. We may now test if

$$F(u_R) \oplus F(u'_R) \oplus u_L \oplus u'_L \tag{5.3}$$

equals 0. If it does, then v can be discarded forever as a possible last round key. The attack procedure is presented as Algorithm 1 and we refer to Figure 5.7 for an illustration of the attack.

Algorithm 1 Impossible differential key recovery pseudo-code for SIMON

Data: Q : set of rotation indices relative to input difference α giving impossible differentials
Result: $\mathcal{K}$: set of remaining key candidates for last round key

1 $\mathcal{K} \leftarrow \mathbb{F}_2^n$
Construct a "basis" of plaintexts $\mathcal{M}$ of size 2^ℓ
foreach $\alpha = (\mathtt{0}\cdots\mathtt{01}) \lll j, j = 0, \ldots, n-1$ **do**
2 **foreach** $m \in \mathcal{M}$ **do**
3 $m' = (m'_L \parallel m'_R) \leftarrow (m_L \oplus \alpha \parallel m_R \oplus F(m_L) \oplus F(m_L \oplus \alpha))$
  Look up $c = (c_L \parallel c_R)$ and query $c' = (c'_L \parallel c'_R) = E_K(m')$
  $\Gamma \leftarrow F(c_R) \oplus F(c'_R) \oplus c_L \oplus c'_L$
  if $\Gamma \in \{\alpha \lll q \mid q \in Q\}$ **then**
4 **foreach** $v \in \mathcal{K}$ **do**
5 A $(u_L \parallel u_R) \leftarrow (c_R \parallel F(c_R) \oplus c_L \oplus v)$
  $(u'_L \parallel u'_R) \leftarrow (c'_R \parallel F(c'_R) \oplus c'_L \oplus v)$
  if $F(u_R) \oplus F(u'_R) \oplus u_L \oplus u'_L = 0$ **then**
6 $\mathcal{K} \leftarrow \mathcal{K} \backslash \{v\}$
7 **end**
8 **end**
9 **end**
10 **end**
11 **end**
12 **return** $\mathcal{K}$

5.4.2 Complexity

In the following, we give our analysis of the key recovery complexity for the impossible differential attack, in terms of data (which we define as the number of encryption oracle queries), memory and computational (time) complexity, given in terms of equivalent number of r-round encryption queries. During our analysis, we refer to the line numbers of Algorithm 1, as well as Equations (5.2) and (5.3).

As the plaintexts of the basis $\mathcal{M}$ of size 2^ℓ are queried once and stored in memory, the data and memory complexity for line 2 is 2^ℓ data and 2^ℓ memory. By choosing $\mathcal{M}$ in a way that we avoid using a particular pair twice in the form of (m, m') and (m', m), the total number of plaintext pairs used for the attack is

$$n \cdot 2^\ell,$$

where the factor n comes from the possible rotations of the input difference $\alpha = (\mathtt{0}\cdots\mathtt{01}) \lll j, j = 0, \ldots, n-1$.

As the number of input differences we iterate over in line 3 is n, and $|\mathcal{M}| = 2^\ell$, the number of m' constructed and queried in lines 5 and 6 is $n \cdot 2^\ell$. These m' are used once and not stored in memory, hence the total memory complexity of the attack is 2^ℓ for storing $\mathcal{M}$, and the total data complexity is $2^\ell + n \cdot 2^\ell = (n+1)2^\ell$.

5.4.2.1 Expected Size of $\mathcal{K}$

When using a particular plaintext pair (m, m') with corresponding ciphertext pair (c, c') in lines 5 through 16, we first check if the difference Γ matches one of the right halves of the

$|Q|$ impossible differences. Assuming that Γ is uniformly distributed with probability mass function 2^{-n}, the probability of entering the **if** statement of line 8 is

$$\frac{|Q|}{2^n},$$

and as such, the expected number of pairs passing the filtering of Equation (5.2) is

$$n2^{\ell} \cdot \frac{|Q|}{2^n}.$$

Consider now a wrong guess v for the key under attack. We know already that for the correct key, the probability of the *if statement* of line 12 being true is zero, due to the miss-in-the-middle property of the impossible differential attack. However, under the assumption that for a wrong key guess v, the difference of Equation (5.3) is uniformly distributed, the probability of discarding a wrong key, using a single pair, is 2^{-n}, and thus the probability of not discarding it is

$$(1 - 2^{-n}).$$

Assuming independency of the probabilities of discarding a wrong key, for each of the $n2^{\ell}$ pairs, the expected number of remaining keys $|\mathcal{K}|$ after using all pairs is

$$\mathbb{E}[|\mathcal{K}|] = 2^n \left(1 - 2^{-n}\right)^{n2^{\ell}|Q|2^{-n}}.$$

5.4.2.2 Time Complexity

For every pair used in lines 9 through 15, i.e. those pairs satisfying $\Gamma \in \{\alpha \lll q \mid q \in Q\}$, we must try as many keys as there are currently in $\mathcal{K}$. The fraction of the set $\mathcal{K}$ which is not discarded by using a single such pair equals the probability that some pair does not discard some wrong key. This probability is computed as

$$\begin{aligned}
&1 - \Pr\left(\text{wrong key } v \text{ discarded by some pair}\right) \\
&= 1 - \Pr\left(\text{pair discards } v \mid \Gamma \in \{\alpha \lll q \mid q \in Q\}\right) \cdot \Pr\left(\Gamma \in \{\alpha \lll q \mid q \in Q\}\right) \\
&= 1 - 2^{-n} \cdot \frac{|Q|}{2^n} \\
&= 1 - \frac{|Q|}{2^{2n}}.
\end{aligned}$$

As such, the expected number of 1-round partial decryptions we will do during the course of the attack, using $n2^{\ell}$ pairs, is determined as

$$\begin{aligned}
&2^n + 2^n \cdot \left(1 - \frac{|Q|}{2^{2n}}\right) + 2^n \cdot \left(1 - \frac{|Q|}{2^{2n}}\right)^2 + \cdots + 2^n \cdot \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^{\ell}-1} \\
&= 2^n \sum_{i=0}^{n2^{\ell}-1} \left(1 - \frac{|Q|}{2^{2n}}\right)^i \\
&= 2^n \cdot \frac{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^{\ell}}}{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)} \\
&= 2^{3n} \cdot \frac{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^{\ell}}}{|Q|}
\end{aligned} \tag{5.4}$$

Evaluating this expression numerically is very computationally intensive for larger values of ℓ and n. For the numerator of Equation (5.4), we can use the fact that $\lim_{x\to\pm\infty} \left(1 - \frac{k}{x}\right)^x = e^{-k}$. We write 2^{ℓ} as $2^{\ell} = c2^{2n}$ for some constant c. Then

$$\begin{aligned}
\lim_{x\to\pm\infty} 2^{3n} \cdot \frac{1 - \left(1 - \frac{|Q|}{2^{2n}}\right)^{n2^{\ell}}}{|Q|} &= 2^{3n} \cdot \frac{1 - e^{-|Q|nc}}{|Q|} \\
&= 2^{3n} \cdot \frac{1 - e^{-|Q|n2^{\ell-2n}}}{|Q|}.
\end{aligned} \tag{5.5}$$

We use the approximation of Equation (5.5), when computing Equation (5.4) is too intensive. For the attack, the time complexity is determined as the total effort spent in the 1-round partial decryption phase, converted to the equivalents of r-round encryption queries. This is done, since 2^n r-round encryption queries would be the effort required to brute-force the key. As such, the total complexity in terms of r-round encryptions equals the expression from either Equation (5.4) or (5.5), multiplied by $\frac{2}{r}$. In Table 5.7 we present our results on key recovery attacks using impossible differentials, for all variants of SIMON, such that the expected number of remaining subkeys is 1% of the whole key space. We note that the complexities for some of the variants of SIMON are higher than brute-force effort, and hence is not considered an attack. However, we include the analysis here such that it assist in reflecting the current security margins when it comes to studying impossible differentials.

5.4.3 Practical Tests

For the case $n = 16$, the block size is small enough that we may actually implement and verify the attack. Thus, we provide in [141] among other cryptanalytic functionalities, our C++ implementation of the key-recovery attack on 14 rounds of Simon32/64, using the 12-round impossible differential.

In Table 5.8, we present the results of 10 experimental runs, the time for each run and the size of the output $|\mathcal{K}|$, with its corresponding percentage of the full round key space. Figure 5.8

Table 5.7: Results on key recovery attack on SIMON using $|Q| \cdot n$ impossible differentials. The number of pairs used, $n2^{\ell}$ is determined such that the expected size of $\mathcal{K}$, i.e. the remaining key candidates, is 1% of the total subkey space 2^n. The complexities indicated with a † are computed using the approximation of Equation (5.5).

Cipher	Rounds		$\lvert Q\rvert$	Pairs	Data	Memory	Time
	Total	Attacked		$n2^{\ell}$	$2^{\ell} + n2^{\ell}$	2^{ℓ}	
Simon32/64	32	14	2	$2^{33.2}$	$2^{33.3}$	$2^{29.2}$	$2^{44.2}$
Simon48/72	36	15	1	$2^{50.2}$	$2^{50.3}$	$2^{45.6}$	$2^{69.1\dagger}$
Simon48/96	36	15	1	$2^{50.2}$	$2^{50.3}$	$2^{45.6}$	$2^{69.1\dagger}$
Simon64/96	42	16	2	$2^{65.2}$	$2^{65.2}$	$2^{60.2}$	$2^{92.0\dagger}$
Simon64/128	44	16	2	$2^{65.2}$	$2^{65.2}$	$2^{60.2}$	$2^{92.0\dagger}$
Simon96/92	52	19	2	$2^{97.2}$	$2^{97.2}$	$2^{92.0}$	$2^{139.7\dagger}$
Simon96/144	54	19	2	$2^{97.2}$	$2^{97.2}$	$2^{91.6}$	$2^{139.7\dagger}$
Simon128/128	68	22	2	$2^{129.2}$	$2^{129.2}$	$2^{123.2}$	$2^{187.5\dagger}$
Simon128/192	69	22	2	$2^{129.2}$	$2^{129.2}$	$2^{123.2}$	$2^{187.5\dagger}$
Simon128/256	72	22	2	$2^{129.2}$	$2^{129.2}$	$2^{123.2}$	$2^{187.5\dagger}$

Table 5.8: Results from key recovery experiments on Simon32/64, using the parameters of Table 5.7. Note, that half the tests were run during the night, where the server was under less load, hence the difference in the runtimes.

Size of $\mathcal{K}$	Time (sec.)	% of 2^n
3805	1619	5.81
789	1636	1.20
2455	1655	3.75
607	1615	0.93
1600	1634	2.44
344	1152	0.52
1536	1190	2.34
2937	1172	4.48
3170	1268	4.84
5259	1207	8.02

shows how the size of $|\mathcal{K}|$ progressed over the course of the attack, when using difference rotation amounts on the input difference.

The summary of the results obtained in this section can be summarised in the following.

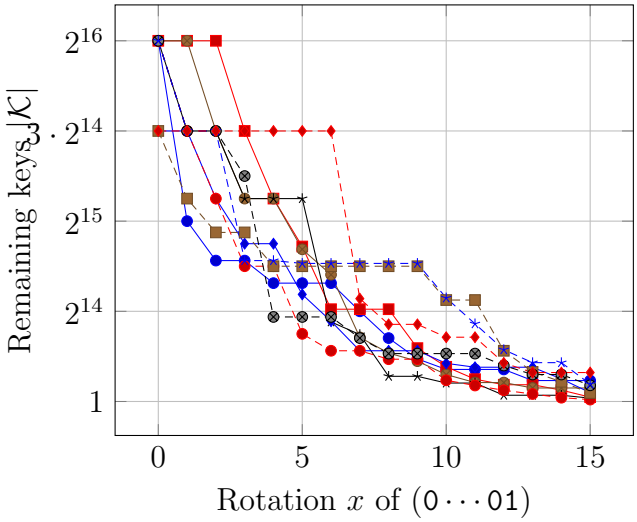


Figure 5.8: Progression of the size of $|\mathcal{K}|$ for the key recovery attack on Simon32/64 using the parameters of Table 5.7, as a function of the rotation amount on the input difference (input difference used is $\alpha = (0 \cdots 01) \lll x, x = 0, \ldots, n-1$. The progressions are from the experimental results of Table 5.8.

5.5 Linear Cryptanalysis

In this chapter we investigate the security of SIMON against different variants of linear cryptanalysis approaches, i.e. classical and linear hulls. We present a connection between linear- and differential characteristics as well as differentials and linear hulls in SIMON. We employ it to adapt the current known results on differential cryptanalysis of SIMON into the linear setting. In addition to finding a linear approximation with a single characteristic, we show the effect of the linear hulls in SIMON by finding better approximations that enable us to improve the previous results. Our best linear cryptanalysis employs average squared correlation of the linear hull of SIMON based on correlation matrices. The result covers 21 out of 32 rounds of SIMON32/64 with time and data complexity $2^{54.56}$ and $2^{30.56}$ respectively. We have implemented our attacks for small scale variants of SIMON and our experiments confirm the theoretical biases and correlation presented in this work. So far, our results are the best known with respect to linear cryptanalysis for any variant of SIMON.

5.5.1 Preliminaries

In order to go through the different details presented in this section we will revisit the concepts presentde in Chapter 3 to give an intuition on how correlation matrices are constructed (which we will use later on in the section).

Linear cryptanalysis finds a linear relation between some plaintexts bits, ciphertexts bits and some secret key bits and then exploits the bias or the correlation of this linear relation. In other words, the adversary finds an input mask α and an output mask β that yield a higher

Table 5.9: Summary of our differential and Impossible cryptanalytic results on SIMON. Note, that entries with a † in the complexity column indicate results which are worse than brute-force search. The parameters for impossible differentials are such, that the expected fraction of remaining keys after the attack is 1%.

Cryptanalysis	Cipher	Rounds		Data	Memory	Time
		Total	Attacked			
Differential	Simon32/64	32	16	$2^{29.5}$	2^{16}	$2^{26.5}$
	Simon48/72	36	18	$2^{46.4}$	2^{24}	$2^{43.3}$
	Simon48/96	36	18	$2^{46.4}$	2^{24}	$2^{43.3}$
	Simon64/96	42	24	$2^{62.0}$	2^{32}	$2^{58.4}$
	Simon64/128	44	24	$2^{62.0}$	2^{32}	$2^{58.4}$
	Simon96/92	52	29	$2^{87.5}$	2^{48}	$2^{83.7}$
	Simon96/144	54	29	$2^{87.5}$	2^{48}	$2^{83.7}$
	Simon128/128	68	40	$2^{124.8}$	2^{64}	$2^{120.5}$
	Simon128/192	69	40	$2^{124.8}$	2^{64}	$2^{120.5}$
	Simon128/256	72	40	$2^{124.8}$	2^{64}	$2^{120.5}$
Impossible Differential	Simon32/64	32	14	$2^{33.3}$	$2^{29.2}$	$2^{44.2}$
	Simon48/72	36	15	$2^{50.3}$	$2^{45.6}$	$2^{69.1}$
	Simon48/96	36	15	$2^{50.3}$	$2^{45.6}$	$2^{69.1}$
	Simon64/96	42	16	$2^{65.2}$	$2^{60.2}$	$2^{92.0}$
	Simon64/128	44	16	$2^{65.2}$	$2^{60.2}$	$2^{92.0}$
	Simon96/92	52	19	$2^{97.2}$	$2^{912.0}$	$2^{139.7\dagger}$
	Simon96/144	54	19	$2^{97.2}$	$2^{91.6}$	$2^{139.738}$
	Simon128/128	68	22	$2^{129.2}$	$2^{123.2}$	$2^{187.527\dagger}$
	Simon128/192	69	22	$2^{129.2}$	$2^{123.2}$	$2^{187.5}$
	Simon128/256	72	22	$2^{129.2}$	$2^{123.2}$	$2^{187.5}$

absolute *bias* $\epsilon_F(\alpha, \beta) \in [-\frac{1}{2}, \frac{1}{2}]$. In other words

$$Pr[\langle \alpha, X \rangle + \langle \beta, F_K(X) \rangle = \langle \gamma, K \rangle] = \frac{1}{2} + \epsilon_F(\alpha, \beta)$$

deviates from $\frac{1}{2}$, where $\langle \cdot, \cdot \rangle$ denotes an inner product. The correlation of a linear approximation is defined as

$$C_F(\alpha, \beta) := 2\epsilon_F(\alpha, \beta)$$

Another definition of the correlation which we will use later is

$$C_F(\alpha, \beta) := \hat{F}(\alpha, \beta)/2^n$$

where n is the block size of F in bits and $\hat{F}(\alpha, \beta)$ is the Walsh transform of F defined as follows

$$\hat{F}(\alpha, \beta) := \sum_{x \in \{0,1\}^n} (-1)^{\beta \cdot F(x) \oplus \alpha \cdot x}$$

For a given output mask β, the Fast Walsh Transform algorithm computes the Walsh transforms of an n-bit block size function F for all possible input masks α with output mask β using $n2^n$ arithmetic operations.

In order to find good linear approximations, one can construct a correlation matrix (or a squared correlation matrix). In the following, we state the definition the correlation matrix and show how the average squared correlation over all the keys is estimated.

Given a composite function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that

$$F = F_r \circ \cdots \circ F_2 \circ F_1,$$

We estimate the correlation of an r-round linear approximation (α_0, α_r) by considering the correlation of each linear characteristic between α_0 and α_r. The correlation of ith linear characteristic $(\alpha_0 = \alpha_{0i}, \alpha_{1i}, \cdots, \alpha_{(r-1)i}, \alpha_r = \alpha_{ri})$ is

$$C_i = \prod_{j=1}^{r} C_{F_j}(\alpha_{(j-1)i}, \alpha_{ji})$$

It is well known, see e.g. [76], that the correlation of a linear approximation is the sum of all correlations of linear trails starting with the same input mask α and ending with the same output mask β, i.e. $C_F(\alpha_0, \alpha_r) = \sum_{i=1}^{N_l} C_i$, where N_l is the number of all the possible linear characteristics between (α_0, α_r).

When considering the round keys which affects the sign of the correlation of a linear trail, the correlation of the linear hull (α, β) is

$$C_F(\alpha, \beta) = \sum_{i=1}^{N_l} (-1)^{d_i} C_i,$$

where $d_i \in \mathbb{F}_2$ refers to the sign of the addition of the subkey bits on the ith linear trail. In order to estimate the data complexity of a linear attack, one uses the average squared correlation over all the keys which is equivalent to the sum of the squares of the correlations of all trails, $\sum_i C_i^2$, assuming independent round keys [76].

Let C denote the correlation matrix of an n-bit key-alternating cipher. C has size $2^n 2^n$ and $C_{i,j}$ corresponds to the correlation of an input mask, say α_i, and output mask, say β_j. Now the correlation matrix for the keyed round function is obtained by changing the signs of each row in C according to the round subkey bits or the round constant bits involved. Squaring each entry of the correlation matrix gives us the squared correlation matrix M. Computing M^r gives us the squared correlations after r number of rounds. This can not be used for real block ciphers that have block sizes of at least 32-bit as in the case of Simon32/64. So in order to find linear approximations one can construct a submatrix of the correlation (or the squared correlation) matrix [10], [49]. In Section 5.5.3, we construct a squared correlation submatrix for Simon in order to find good linear approximations.

5.5.2 Connections and Linear Cryptanalysis of SIMON

In this section we will investigate the possibility to use connections between differential and linear cryptanalysis and its variants in order to provide better results on SIMON. In some

cases as linear hulls, this yields a better time and data complexity for more rounds. We should note that our results on classical linear cryptanalysis can be seen in [16], as the best results is achieved for 13 rounds for linear bias of 2^{-16} and a data complexity of $2^{3}2$ (the full codebook) for SIMON32/64. The details of this attack will be left to the reader.

5.5.2.1 Connections between Linear and Differential Characteristics for SIMON

In this section, we explain the connections described in [16] pertaining to the connection between linear- and differential characteristics for SIMON, and its application to SIMON variants other than SIMON32/64.

In the round function of SIMON, the only non-linear operation is the bitwise AND. Note that, given single bits A and B, the output of $(A\&B)$ is 0 with probability $\frac{3}{4}$. Hence, we can extract the following highly biased linear expressions for the F-function:

$$\left.\begin{array}{l} \text{Approximation 1}: \ Pr[(F(X))_i = (X)_{i-2}] = \frac{3}{4} \\ \text{Approximation 2}: \ Pr[(F(X))_i = (X)_{i-2} \oplus (X)_{i-1}] = \frac{3}{4} \\ \text{Approximation 3}: \ Pr[(F(X))_i = (X)_{i-2} \oplus (X)_{i-8}] = \frac{3}{4} \\ \text{Approximation 4}: \ Pr[(F(X))_i = (X)_{i-2} \oplus ((X)_{i-1} \oplus (X)_{i-8})] = \frac{1}{4} \end{array}\right\} \tag{5.6}$$

Similarly, differential cryptanalysis [37] is a widely used chosen plaintext/ciphertext cryptanalytic attack technique. In a differential attack we look for an input pair with difference ΔX that propagates to an output pair with difference ΔY with a high probability p. This differential characteristic is denoted by $\Delta X \xrightarrow{p} \Delta Y$.

There are many works which discuss connection between differential and linear characteristics [45, 66]. We observe that there is an explicit connection between linear characteristic and differential characteristic for SIMON. This observation is explained as follows. We can also extract the following highly probable differential expressions for the F-function:

$$\left.\begin{array}{l} \text{Differential Characteristic 1}: \ (\Delta X)_i \xrightarrow{\frac{1}{4}} (\Delta F(X))_{i+2} \\ \text{Differential Characteristic 2}: \ (\Delta X)_i \xrightarrow{\frac{1}{4}} (\Delta F(X))_{i+2,i+1} \\ \text{Differential Characteristic 3}: (\Delta X)_i \xrightarrow{\frac{1}{4}} (\Delta F(X))_{i+2,i+8} \\ \text{Differential Characteristic 4}: (\Delta X)_i \xrightarrow{\frac{1}{4}} (\Delta F(X))_{i+2,i+1,i+8} \end{array}\right\}, \tag{5.7}$$

where $(\Delta F(X))_{i+1,i+8}$ denotes differences in $(i+1)$-th and $(i+8)$-th bits for $\Delta F(X)$ to be 1 and remaining bit positions of $\Delta F(X)$ are 0 (and similarly for the other expressions). Given Equations 5.7 and comparing it with the related equation for a linear approximation of the function F, i.e. Equations 5.6, and the fact that for linear characteristic we approximate bits from output of F by bits from its input and for a differential characteristic we propagate differences in bits of input to the bits of output of F, we see a unique connection between Equations 5.6 and Equations 5.7. In other words, each approximation in Equation 5.6 can be mapped to a differential characteristic in Equation 5.7. Based on this observation, Algorithm 2 represents an approach to convert an r-round differential characteristic to an equivalent r-round linear characteristic. For example, consider a 3-round DC for SIMON 32/64 for which

the sequence of active bits in the input of different rounds are as follows:

$$\{((\Delta X)^0_L;(\Delta X)^0_R),((\Delta X)^1_L;(\Delta X)^1_R),((\Delta X)^2_L;(\Delta X)^2_R),((\Delta X)^3_L;(\Delta X)^3_R)\} =$$
$$\{(-;3),(3;-),(5;3),(3,7;5)\}$$

In addition, assume that while converting this DC to an LC we also want to map bit 3, as an active bit in the first round which is denoted by x in Algorithm 2, to bit 10 which is denoted by y in Algorithm 2. Hence, following Step 13 of the algorithm, the active bits in the left side of the DC are moved to the right side of the equivalent LC and the active bits in the right side of the DC are moved to the left side of the equivalent LC. Hence, at the end of Step 13 we have:

$$\{(X^0_L;X^0_R),(X^1_L;X^1_R),(X^2L;X^2_R),(X^3_L;X^3_R)\} = \{(3;-),(-;3),(3;5),(5;3,7)\}$$

Given that $x = 3$ and $y = 10$, when the active bits (denoted by z) of LC are corrected in Step 18, in the input of the first round the only active bit is bit 3 which should be corrected as $y-(z-x) = 10-(3-3) = 10$. Similarly, position 5 is corrected as $y-(z-x) = 10-(5-3) = 8$ and position 7 is corrected as $y-(z-x) = 10-(7-3) = 6$. Hence, Algorithm 2 returns the following LC as the output:

$$\{(X^0_L;X^0_R),(X^1_L;X^1_R),(X^2L;X^2_R),(X^3_L;X^3_R)\} = \{(10;-),(-;10),(10;8),(8;10,6)\}$$

Now we investigate the strength of different variants of SIMON against linear attack, given the above observation and the known results on differential cryptanalysis of variants of SIMON from [13]. For SIMON32/64 reduced to 11 rounds, a linear characteristics based on the Abed *et. al.* [13] approach will have bias of 2^{-17}. However, we considered the propagation of number of approximations for this variant of SIMON on more rounds and found the following pattern

$$\ldots,1,2,1,3,2,3,1,2,1,1,0,1,1,2,1,3,2,3,1,2,1,1,0,1,1,2,1,3,2,3,\ldots$$

Based on this pattern, it is possible to generate a pattern that has bias of 2^{-16} for 11-round, as

$$2,3,1,2,1,1,0,1,1,2,1.$$

Based on a similar strategy, it is possible to present linear characteristics for other variants of SIMON. We summarize the parameters of our linear attacks for the different variants of SIMON in Table 5.10. On the other hand, to use an approximation with the bias of ϵ to mount a linear attack the expected complexity is $O(\epsilon^{-2})$ [153]. Hence, we consider a case where $\epsilon \geq 2^{-n+2}$, where $N = 2n$ and for the complexity of $8\epsilon^{-2}$ the success probability of key recovery attack would be 0.997 [13, 153]. Our results for different variants of SIMON when $\epsilon \geq 2^{-n+2}$ have been represented in Table 5.11.

Letting $(X)[i_1,...,i_m] = (X)_{i_1} \oplus \ldots \oplus (X)_{i_m}$, it is possible to extract the linear expression related to each variant of SIMON that include only input, output and key bits. For example, the 11-round linear expression for SIMON32/64 is

$$\begin{pmatrix} (P_R)[0,8] \oplus (P_L)[2,10,14] \\ \oplus (C_R)[6,10] \oplus (C_L)_4 \end{pmatrix} = \begin{pmatrix} (K^1)[0,8] \oplus (K^2)[2,6,10] \oplus (K^3)_4 \oplus \\ (K^4)[6,10] \oplus (K^5)_8 \oplus (K^6)_{10} \oplus (K^8)_{10} \\ \oplus (K^9)_8 \oplus (K^{10})[6,10] \oplus (K^{11})_4 \end{pmatrix}. \quad (5.8)$$

Algorithm 2 A general algorithm to convert an r-round differential characteristic (DC) for SIMON N/K to an equivalent r-round linear characteristic (LC) for SIMON N/K.

Input:

- An r-round DC for SIMON N/K, where
  - $(\Delta X)^i_L; (\Delta X)^i_R$ for $0 \leq i \leq r$ are the position of active bits in the input of round i,
  - $N = 2n$,
  - and DC is given as a sequence of the location of active bits for each round in the

    left/right side.

$X^i_L \leftarrow (\Delta X)^i_R$ and $X^i_R \leftarrow (\Delta X)^i_L$, for $0 \leq i \leq r$
// $(\Delta X)^0_R$ is the sequence of active bits in the right side of round i of the given DC
if $(\Delta X)^i_R \neq \phi$ **then** // $(\Delta X)^0_R \neq \phi$ means there is no active bit in $(\Delta X)^0_R$
  select $x \in (\Delta X)^0_R$ // x is a location of an active bit in $(\Delta X)^0_R$
else
  select $x \in (\Delta X)^0_L$ // x is a location of an active bit in $(\Delta X)^0_L$
select $y \leq \frac{N}{2}$ // y is a position in LC which corresponds to the position x of the DC, note that DC/LC are rotation invariant [13] for $0 \leq i \leq r$ and for any $z \in \{X^i_L; X^i_R\}$: $z \leftarrow y - (z - x) \ mod \ \frac{N}{2}$ // z is the position of an active bit(any active bit in the DC has an equivalent active bit in the generated LC) **return** $X^i_L; X^i_R$ for $0 \leq i \leq r$

5.5.2.2 A Key Recovery Attack on SIMON Using the Matsui's Algorithm 2

Given an 11-round linear characteristic such as Equation 5.8, we can add another one round to the beginning and one round to the end of the characteristic to extend the attack up to 13-rounds free of any extra approximation [16]. To extend the 11-round linear characteristic to more rounds we use Algorithm 2 of Matsui to recover the key, where we guess subkyes of rounds at the beginning and the end of the cipher and determine the correlation of the following linear relation to filter the wrong subkeys:

$$(X^i_R)[0, 8] \oplus (X^i_L)[2, 10, 14] \oplus (X^{i+11}_R)[6, 10] \oplus (X^{i+11}_L)_4. \tag{5.9}$$

With respect to Figure 5.9, for the current 11-round linear hull, we evaluate,

$$(X^i_R)[0, 8] \oplus (X^i_L)[2, 10, 14] \oplus (X^{i+11}_R)[6, 10] \oplus (X^{i+11}_L)_4.$$

If we add a round in the backwards direction, i.e. round $i-1$, we can determine $(X^i_L)[2, 10, 14]$ as a function of $F(X^{i-1}_L)[2, 10, 14] \oplus (K^i)[2, 10, 14] \oplus X^i_R)[2, 10, 14]$, where we know X^{i-1}_R and

Table 5.10: Summary of linear analysis for the different variants of SIMON [16]. In this table **KR** denotes a linear characteristic that can be used trough a key recovery attack, **Dis** denotes a linear characteristic that can be used trough a distinguishing attack and **App.** denotes approximation.

	Linear Expression							
	Start		End					
SIMON	Active bits in the left side	Active bits in the right side	Active bits in the left side	Active bits in the right side	# Rounds	# App.	Bias	Attack
32/64	$10, 6, 2, 6, 14$	$8, 0$	$2, 10, 6, 2$	4	11	15	2^{-16}	KR
32/64	$4, 8, 4, 0$	$10, 6, 2$	$2, 14, 10$	12	22	31	2^{-32}	Dis
48/96	$2, 18, 14, 10$	12	$20, 0, 20, 16$	$2, 22, 18$	14	22	2^{-23}	KR
48/96	$2, 18, 14, 10$	12	$10, 22, 6, 6$	8	23	46	2^{-47}	Dis
64/128	$2, 26, 22, 18$	20	$2, 26, 22, 18$	20	17	28	2^{-29}	KR
64/128	$2, 26, 18, 28, 14, 28, 62, 24, 10$	$30, 0, 26, 12$	$2, 26, 18, 28, 14, 28, 62, 24, 10$	$30, 0, 26, 12$	25	60	2^{-61}	Dis
96/144	$2, 46, 42, 46, 38$	$0, 40$	$2, 46, 42$	44	27	46	2^{-47}	KR
96/144	$2, 42, 38, 34, 46, 38, 30$	$0, 40, 32$	$36, 0, 40, 36, 32$	$2, 42, 38, 34$	36	70	2^{-71}	Dis
128/256	$52, 0, 56, 52, 48$	$2, 58, 54, 50$	$2, 58, 54, 50$	52	34	63	2^{-64}	KR
128/256	$36, 0, 48, 40, 36, 32$	$2, 50, 42, 38, 34$	$2, 50, 42, 38, 34, 62, 46, 38, 30$	$0, 48, 40, 32$	52	127	2^{-128}	Dis

Table 5.11: Summary of linear analysis for the different variants of SIMON such that one can mount a linear attack with the success probability of 0.997 [16]. In this table **App.** denotes approximation.

	Linear Expression						
	Start		End				
SIMON	Active bits in the left side	Active bits in the right side	Active bits in the left side	Active bits in the right side	# Rounds	# App.	Bias
32/64	$10, 6, 2$	4	$0, 8, 0, 8, 4$	$2, 10, 6$	10	13	2^{-14}
48/96	$2, 18, 14, 10$	12	$2, 22, 18$	20	13	19	2^{-20}
64/128	$2, 26, 22, 18$	20	$2, 26, 22, 18$	20	17	28	2^{-29}
96/144	$2, 46, 42, 46, 38$	$0, 40$	$0, 0, 4$	$2, 46$	26	45	2^{-46}
128/256	$2, 58, 54, 50$	52	$2, 58, 54, 50$	52	33	59	2^{-60}

X_L^{i-1}. Hence, it is possible to use the correlation of the following linear relation to filter the wrong subkeys:

$$(X_R^i)[0, 8] \oplus (X_L^i)[2, 10, 14] \oplus (X_R^{i+11})[6, 10] \oplus (X_L^{i+11})_4 \oplus (K^i)[2, 10, 14].$$

We can continue our method to add more rounds to the beginning of linear hull in the cost of guessing some bits of subkeys. To add more rounds in backward, for example we must guess the bit $(F(X_L^{i-1}))_2 = (X_L^{i-1})_0 \oplus ((X_L^{i-1})_1 \& (X_L^{i-1})_{10})$. On the other hand, to determine $(F(X_L^{i-1}))_2$ one should guess $(X_L^{i-1})_0$ and $(X_L^{i-1})_1$ only if the guessed value for $(X_L^{i-1})_{10}$ is 1. So, in average we need one bit guess for $(X_L^{i-1})_1$ and $(X_L^{i-1})_{10}$ (in Figure 5.9 such bits are indicated in blue).

Figure 5.9 shows the bits of subkeys that should be guessed when we add 3 rounds to the beginning and 3 rounds to the end of the above 11-round characteristic (27.5 bits of subkeys).

Hence, we can attack 17 rounds of SIMON32/64 using Algorithm 2 of Matsui to recover the key. The time complexity for this attack is $2^{59.5}$ and the data complexity is 2^{32}.

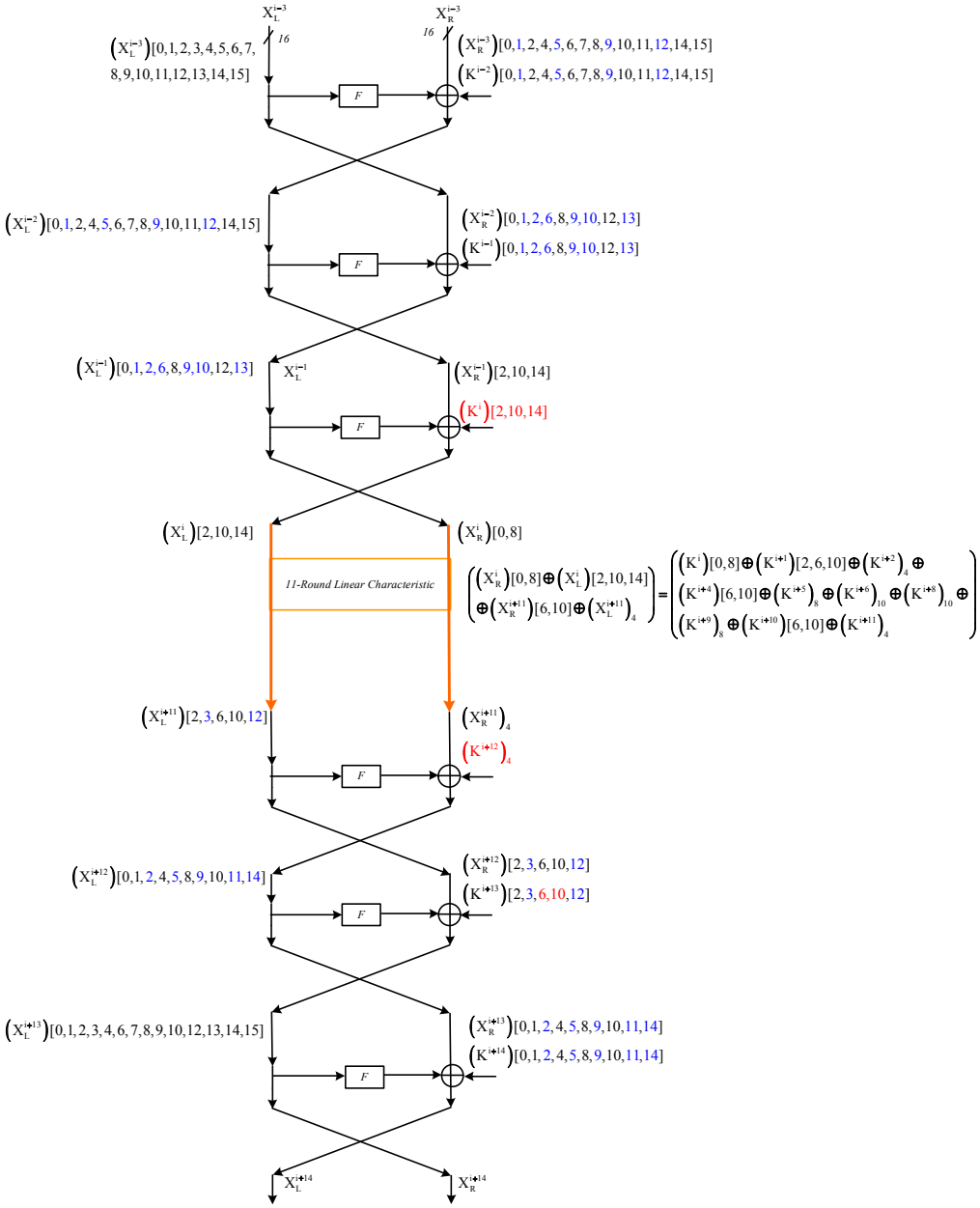


Figure 5.9: The keys (in *black*) that should be guessed to attack 17 rounds of SIMON32/64. The red bits are not required to be guessed and the blue bits cost guessing a half bit on average.

Similarly, in the Appendix of [158], we apply this technique to the variants SIMON48/K, SIMON64/K, SIMON96/K and SIMON128/K, to extend the linear characteristics to more rounds. In particular, we use Algorithm 2 of Matsui to recover the key, where we guess subkeys of rounds at the beginning and the end of each characteristic and determine the correlation

of the related linear relation between the input and the output of the characteristic to filter the wrong subkeys. The figures indicate the bits of subkeys that should be guessed when we add extra rounds to each variants of SIMON. The results using Matsui's Algorithm 2 are summarized in Table 5.16. The details of extension to the other variants will be left to the reader.

5.5.2.3 Linear Hulls of SIMON

Let us first revisit some necessary information by going over the concept of multiple linear cryptanalysis [115], [106], [39]. The technique of multiple linear cryptanalysis, an improved version of the linear cryptanalysis, is proposed by Biryukov et al. in 2004 [39]. This attack is applicable to (reduced-round) ciphers that have more than one approximation. Suppose that, there are m approximations on r rounds of a cipher as follows:

$$P^i_{p_j} \oplus C^i_{c_k} = K^i_{k_l} \quad (1 \leqslant i \leqslant m). \tag{5.10}$$

The goal is to recover bits of key or finding some informations about the key bits that appear in Equation 5.10. An explicit approach is that a counter t_i is associated with each approximation and increased when the corresponding linear approximation is verified for a particular pair of known plaintext and ciphertext. As for algorithm 1 of Matsui [153], the values of $K^i_{k_l}$ are determined from the experimental bias $(t_i - N/2)/N$ and the theoretical bias ϵ_i (bias of the approximation i) by means of a maximum likelihood rule [70,153]. In [39] the authors show that the theoretical data complexity of the generalized multiple linear cryptanalysis is decreased compared to the original attack. The data complexity of the attack is inversely proportional to the capacity of the system of m approximations used, which is given by

$$\overline{c}^2 = 4 \sum_{i=1}^{m} \epsilon_i^2. \tag{5.11}$$

In other words, by increasing the quantity of Equation 5.11, one can decrease the data complexity of the attack. Therefore, finding more approximations is the main task in multiple linear cryptanalysis.

Using the previous the definition of capacity in 5.11, one can defined a connection between capacity and Expected Differential Probability (EDP) for SIMON. A differential of SIMON with fixed input and output difference is composed of many differential characteristics of the cipher, with the same input and output difference. Suppose that there are m differential characteristics with input difference α and output difference β of probability $p_i(\alpha, \beta)$, $1 \leq i \leq m$. Then *Expected Differential Probability* for the differential with the same input and output difference is defined in the following way:

$$EDP(\alpha, \beta) = \sum_i p_i(\alpha, \beta). \tag{5.12}$$

In this section, we extend the given connection between a linear characteristic and differential characteristic in Section 5.5.2.1 to a connection between capacity of a system of approximations (in multiple linear cryptanalysis) and expected differential probability for SIMON as Theorem 5.5.1.

Theorem 5.5.1. *Suppose that there are m differential characteristics for SIMON reduced to r rounds that result a differential with probability p for the r rounds. Then there are m linear characteristics for SIMON reduced to r rounds that produce a system of approximations of capacity:*

$$\overline{c}^2 = p.$$

Proof. Suppose that differential characteristic i has probability p_i where $1 \leq i \leq m$. Then expected differential probability, p, for the m differential characteristics is:

$$p = \sum_{i=1}^{m} p_i.$$

On the other hand in Section 5.5.2.1, it is shown that for a differential characteristic of probability q, there is a linear characteristic of bias $2^{-1} \cdot q^{1/2}$ for SIMON. Therefore, using the m differential characteristics of probability p_i, m linear characteristics of bias ϵ_i can be found where $\epsilon_i = 2^{-1} \cdot p_i^{1/2}$ or equivalently $\epsilon_i^2 = 2^{-2} \cdot p_i$. Then

$$p = \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} 4\epsilon_i^2 = 4 \sum_{i=1}^{n} \epsilon_i^2 = \overline{c}^2. \tag{5.13}$$

□ □

Similarly to the connection between EDP of a differential and capacity of a system of linear equations (in the multiple linear cryptanalysis), one can show a relation between EDP of a differential and capacity of a system of linear hull for SIMON as Theorem 5.5.2.

Theorem 5.5.2. *Suppose that there are m differential characteristics for SIMON reduced to r rounds, with fixed input and output difference, that result a differential with probability p for the r rounds. Then there are m linear characteristics for SIMON reduced to r rounds, with fixed input and output mask, that produce a linear hull of capacity*

$$\overline{c}_{LH}^2 = 2^{-2} \cdot p.$$

Alkhzaimi and Lauridsen in [18] and Abed et al. in [14] found many differential characteristics for some variants of SIMON which yield the desirable differentials for the cipher. In addition, a maximum number of the differential characteristics for some variants of SIMON was investigated by Biryukov et al. [40]. Based on the connection between linear hulls and differentials of SIMON, one can use the differentials by Abed et al. in [14] or differentials by Biryukov et al. in [40] to find the corresponding linear hulls for variants of reduced-round SIMON. We find the linear characteristics for SIMON32/64, $48/K$, and $64/K$ reduced to 13, 15, and 21 rounds, respectively, based on the differential trails by Biryukov et al. For SIMON $96/K$ and $128/K$ reduced to 30 and 41 rounds, we use differential trails by Abed et al. Using those linear characteristics, we can find suitable linear hulls for each variant of SIMON. The summary of the results is presented in Table 5.12, In addition to more tables reflecting the same results for other variants in the Appendix of [158].

Table 5.12: Linear characteristics based on the differential trials by Biryukov et al. for SIMON32/64

	Differential		Linear		
r	$\triangle_L$	$\triangle_R$	X_L	X_R	Used App.
0	–	6	6	–	–
1	6	–	–	6	1
2	8	6	6	4	1
3	6, 10	8	4	2, 6	1; 1
4	12	6, 10	2, 6	0	1
5	6, 10, 14	12	0	2, 6, 14	1; 1; 1
6	0, 8	6, 10, 14	2, 6, 14	4, 12	1; 1
7	2, 6, 14	0, 8	4, 12	6, 10, 14	1; 1; 1
8	4	2, 6, 14	6, 10, 14	8	1
9	2, 14	4	8	10, 14	1; 1
10	0	2, 14	10, 14	12	1
11	14	0	12	14	1
12	–	14	14	–	–
13	14	–	–	14	–
	$\sum_r \log_2 pr = -36$ $\log_2 p_{diff} = -29.69$ # trails = 45083		$\log_2 \epsilon^2 = -38$ $\log_2 \bar{c}_{LH}^2 = -31.69$ # characteristics = 45083		

5.5.2.3.1 Extending Linear Hulls and Key Recovery Attack on SIMON32/64.

Similar to the approach we used to extend a linear characteristic when it is used in Algorithm 2 of Matsui (see Section 5.5.2.2), it is possible to extend a given linear hull for more rounds. For example, consider the linear hull based on the differential by Biryukov et al. for 13-round SIMON32/64. The input and output mask of the linear hull is $(\Gamma_6, -)$ and $(-, \Gamma_{14})$. We extend it by adding some rounds to the beginning and the end of the cipher, as follows.

the Backwards Direction With respect to Figure 5.10, to utilize a new 14-round linear hull using input mask $(-, \Gamma_6)$. We can continue our method to add more rounds to the beginning of linear hull in the cost of guessing some bits of subkeys similar to the approach presented in section 5.5.2.2.

In the Forward Direction We can use the same approach to add some rounds to the end of linear hull in the cost of guessing some bits of subkeys. More details are depicted in Figure 5.10.

We can extend the 13-round linear hull of SIMON32/64 by eight rounds (by adding four rounds at the beginning and four rounds to the end) in a key-recovery attack such that the total computational effort for collecting plaintext-ciphertext pairs and testing all subkey candidates for the appended rounds remains significantly smaller than for exhaustively searching the full key space.

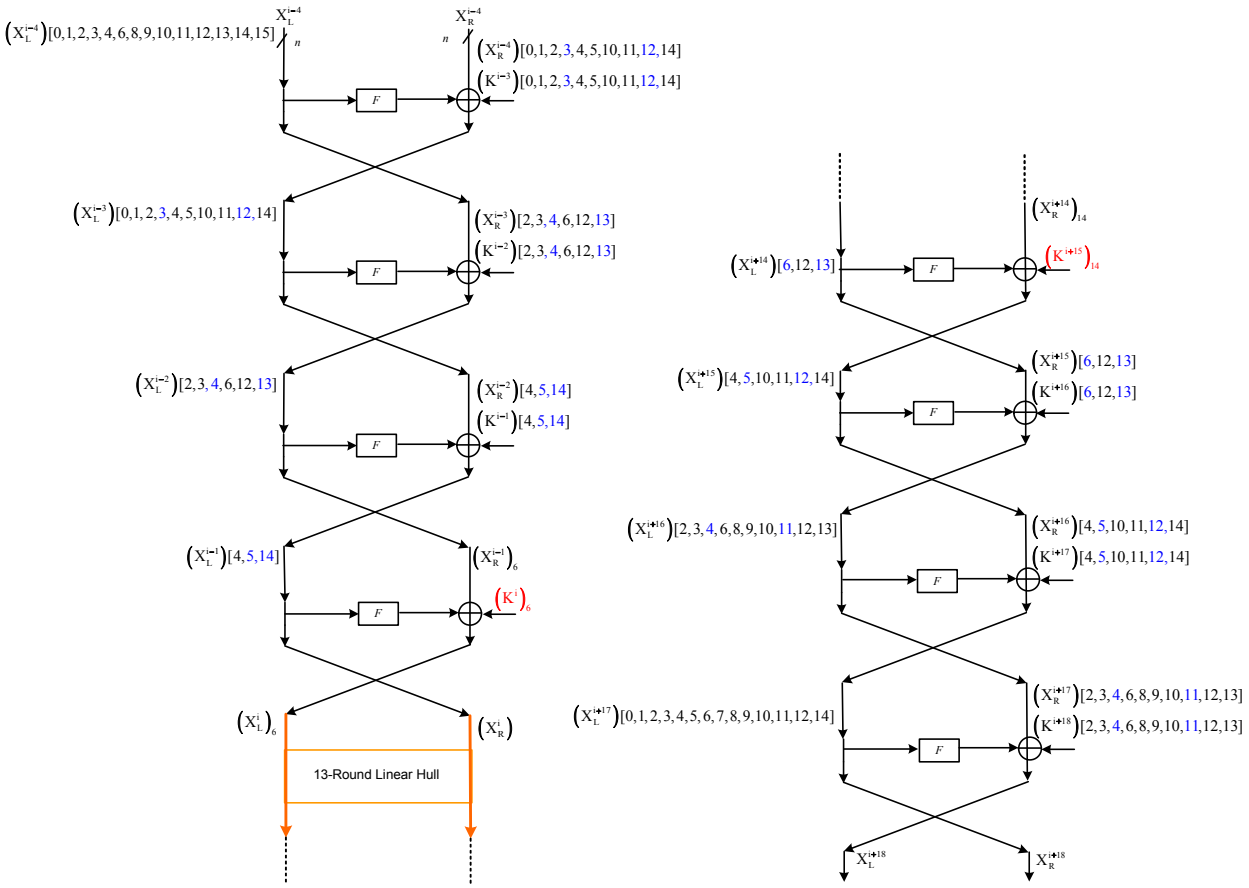


Figure 5.10: The subkey bits (in black) that should be guessed to attack 21 rounds of SIMON32/64. The red bits are not required to be guessed.

5.5.2.3.2 Attack Complexity

We require $2^{31.69}$ known plaintexts. We also need $2^{31.69}$ encryptions for producing the required known plaintexts and $2^{31.69}2^{32}$ encryptions to find the round-key bits on average. Therefore, the time complexity of the attack is $2^{31.69} + 2^{31.69}2^{32} \approx 2^{63.69}$.

5.5.2.3.3 Key Recovery Attack on Other Variants of SIMON

In the above, we explain a key recovery attack which uses a linear hull on SIMON32/64. The same procedure can be applied to other variants of SIMON, see Appendix **??** for more details. A summary of our results on the linear hull cryptanalysis of SIMON48/K, 64/K, 96/K, and 128/K is presented in Table 5.16. It must be noted that we use the linear hulls in obtaining the corresponding results of attacks on the variants which are in tables in the appendix of [158]

5.5.3 Linear Hull Effect in SIMON

In this section we will investigate the linear hull effect on SIMON using the correlation matrix method to compute the average squared correlation.

5.5.3.1 Correlation of the SIMON F Function

The section provides an analysis on some linear properties of the SIMON F function regarding the squared correlation. This will assist in providing an intuition around the design rationale when it comes to linear properties of SIMON round Function F. A general linear analysis was applied on the F function of SIMON, with regards to limits around the squared correlations for all possible Hamming weights on input masks α and output masks β, for SIMON32/64. The following observations were made based on results in Table 5.14.

- The best linear characteristics for a single application of F is obtained for input and output masks with Hamming weight as low as 1 and 2.
- The best squared correlation obtained is 2^{-2} and the lowest is 2^{-16} for all possible Hamming weights on the input and output masks of F.

5.5.3.2 Constructing Correlation Submatrix for SIMON

To construct a correlation submatrix for SIMON, we make use of the following proposition.

Proposition 5.5.3 (Correlation of a one-round linear approximation [32])**.** *Let $\alpha = (\alpha_L, \alpha_R)$ and $\beta = (\beta_L, \beta_R)$ be the input and output masks of a one-round linear approximation of SIMON. Let α_F and β_F be the input and output masks of the SIMON F function. Then the correlation of the linear approximation (α, β) is $C(\alpha, \beta) = C_F(\alpha_F, \beta_F)$ where $\alpha_F = \alpha_L \oplus \beta_R$ and $\beta_F = \beta_L = \alpha_R$.*

As our goal is to perform a linear attack on SIMON, we construct a squared correlation matrix in order to compute the average squared correlation (the sum of the squares of the correlations of all trails) in order to estimate the required data complexity. Algorithm 3 constructs a squared correlation submatrix whose input and output masks have Hamming weight less than a certain Hamming weight m. Algorithm 3 uses the Fast Walsh Transform algorithm to compute the correlations of a given input and output masks for the F function of SIMON.

Algorithm 3 Construction of SIMON's Correlation Submatrix

Require: Hamming weight m, bit size of SIMON's F function n and a *map* function.
Ensure: Squared Correlation Submatrix M

for all output masks β with Hamming weight $\leq m$ **do**
  Extract from β the left/right output masks β_L and β_R.
  $\alpha_R \leftarrow \beta_L$.
  Compute $\hat{F}(\alpha_F, \beta_L)$ to SIMON's F function for all possible α_F.
  for all input masks α_F to SIMON's F function **do**
    $c \leftarrow \hat{F}(\alpha_F, \beta_L)/2^n$.
    $\alpha_L \leftarrow \alpha_F \oplus \beta_R$.
    $\alpha = \alpha_L || \alpha_R$.
    if $c \neq 0$ **and** Hamming weight of $\alpha \leq m$ **then**
      $i \leftarrow map(\alpha)$. {map α to a row index i in the matrix M}
      $j \leftarrow map(\beta)$. {map α to a column index j in the matrix M}
      $M(i, j) = cc$.
    end if
  end for
end for

The size of the submatrix is $\sum_{i=0}^{m} \binom{2n}{i} \sum_{i=0}^{m} \binom{2n}{i}$ where n is the block size of SIMON's F function. One can see that the time complexity is in the order of $2^n \sum_{i=0}^{m} \binom{2n}{i}$ arithmetic operations. The submatrix size is large when $m > 5$, but most of its elements are zero and therefore it can easily fit in memory using a sparse matrix storage format. The table below shows the number of nonzero elements of the squared correlation submatrices of SIMON32/K when $1 \leq m \leq 9$. One can see that these matrices are very sparse (see Table 5.13). For instance, when $m \leq 8$, the density of the correlation matrix is very low, namely $\frac{133253381}{1503317315033173} \approx 2^{-20.7}$.

Table 5.13: SIMON32/K matrices using masks with Hamming weight $\leq m$, nnz = number of nonzero elements

m	Size of M	nnz
1	3333	17
2	529529	233
3	54895489	2835
4	4144941449	31381
5	242825242825	308805
6	11490171149017	2671829
7	45148734514873	20206757
8	1503317315033173	133253381
9	4308197343081973	763347577

5.5.3.3 Improved Linear Approximations

One can see that Algorithm 3 is embarrassingly parallelizable. Thus, the memory complexity rather than the time complexity is dominating. On a standard PC, we are able to construct a sparse squared correlation matrix of SIMON32/K with input and output masks that have Hamming weight ≤ 8. Using this matrix, we find new 14-round linear approximations with an average squared correlation $\leq 2^{-32}$ for SIMON32/K. We also get better estimations for the previously found linear approximations which were estimated before using only a single linear characteristic rather than considering many linear characteristics with the same input and output masks. For example, in [14], the squared correlation of the 9-round single linear characteristic with input mask `0x01110004` and output mask `0x00040111` is 2^{-20}. Using our matrix, we find that this same approximation has a squared correlation $\approx 2^{-18.4}$ with $11455 \approx 2^{13.5}$ trails, which gives us an improvement by a factor of $2^{1.5}$. Note that this approximation can be found using a smaller correlation matrix of Hamming weight ≤ 4 and we get an estimated squared correlation equal to $2^{-18.83}$ and only 9 trails. So the large number of other trails resulting covering Hamming weights ≥ 5 is insignificant as they only cause a factor of $2^{0.5}$ improvement.

Also, the 10-round linear characteristic in [17] with input mask `0x01014404` and output mask `0x10004404` has squared correlation 2^{-26}. Using our correlation matrix, we find that this same approximation has an estimated squared correlation $2^{-23.2}$ and the number of trails is $588173 \approx 2^{19.2}$. This gives an improvement by a factor of 2^3. Note also that this approximation can be found using a smaller correlation matrix with Hamming weight ≤ 5 and we get an estimated squared correlation equal to $2^{-23.66}$ and only 83 trails. So the large number of other trails resulting covering Hamming weights ≥ 5 is insignificant as they only cause a factor of $2^{0.4}$ improvement. Both of these approximations give us squared correlations less than 2^{-32} when considering more than 12 rounds.

In the following, we describe the new 14-round linear hulls found using a squared correlation matrix with Hamming weight ≤ 8.

5.5.3.3.1 New 14-round Linear Hulls.

Consider a squared correlation matrix M whose input and output masks have Hamming weight m. When $m \geq 6$, raising the matrix to the rth power, in order to estimate the average squared correlation, will not work as the resulting matrix will not be sparse even when r is small. For example, we are able only to compute M^6 where M is a squared correlation matrix whose masks have Hamming weight ≤ 6. Therefore, we use matrix-vector multiplication or row-vector matrix multiplications in order to estimate the squared correlations for any number of rounds r.

It is obvious that input and output masks with low Hamming weight gives us better estimations for the squared correlation. So we performed row-vector matrix multiplications using row vectors corresponding to Hamming weight one. We found that when the left part of the input mask has Hamming weight one and the right part of input mask is zero, we always get a 14-round squared correlation $\approx 2^{-30.9}$ for four different output masks. So in total we get 64 linear approximations with an estimated 14-round squared correlation $\approx 2^{-30.9}$.

We also constructed a correlation matrix with masks of Hamming weight ≤ 9 but we have only got a slight improvement for these 14-round approximations by a factor of $2^{0.3}$. We have found no 15-round approximation with squared correlation more than 2^{-32}. Table 5.15 shows the 14-round approximations with input and output masks written in hexadecimal notation.

Table 5.14: General analysis to the best and lowest squared correlations in SIMON32/64 for all possible Hamming weights entering the F function

Hamming weight	Best Sq. Corr	Lowest Sq. Corr
1	2^{-2}	2^{-2}
2	2^{-2}	2^{-4}
3	2^{-4}	2^{-6}
4	2^{-4}	2^{-8}
5	2^{-6}	2^{-10}
6	2^{-6}	2^{-12}
7	2^{-8}	2^{-14}
8	2^{-8}	2^{-16}
9	2^{-10}	2^{-16}
10	2^{-10}	2^{-16}
11	2^{-12}	2^{-16}
12	2^{-12}	2^{-16}
13	2^{-14}	2^{-16}
14	2^{-14}	2^{-16}
15	2^{-16}	2^{-16}
16	2^{-14}	2^{-14}

5.5.3.4 Key Recovery Attack using Linear Hulls

Similar to the approach we used in previous sections to add extra rounds to the given linear trail, we extend the given linear hull for 14 rounds of SIMON32/64 by adding some rounds to the beginning and the end of the cipher, as follows.

5.5.3.4.1 In the Backwards Direction

We start with the input mask of the 14-round linear hull (e.g. $(\Gamma_0, -)$) and go backwards to add some rounds to the beginning. More precisely, for the current 14-round linear hull, we evaluate $((X_L^i)_0 \oplus (X_R^{i+14})_8 \oplus (X_L^{i+14})_6)$. If we add a round in the backwards direction, i.e. round $i-1$, we know X_R^{i-1} and X_L^{i-1}, so

$$(X_L^{i-1})_{14} \oplus ((X_L^{i-1})_{15} \& (X_L^{i-1})_8) = (X_R^{i-1})_0 \oplus (K^i)_0 \oplus (X_L^i)_0.$$

Hence, we can consider $((X_L^i)_0 \oplus (X_R^{i+14})_8 \oplus (X_L^{i+14})_6) \oplus (K^i)_0$ as the new linear hull. We can continue our method to add more rounds to the beginning of linear hull at the cost of guessing some bits of subkeys.

Table 5.15: 14-round linear hulls for SIMON32/K found, using Hamming weight ≤ 9

α	β	$\log_2 c^2$	$\log_2 N_t$
80000000	00800020, 00800060, 00808020, 00808060	−30.5815	28.11
40000000	00400010, 00400030, 00404010, 00404030	−30.5815	28.11
20000000	00200008, 00200018, 00202008, 00202018	−30.5815	28.11
10000000	00100004, 0010000C, 00101004, 0010100C	−30.5815	28.11
08000000	00080002, 00080006, 00080802, 00080806	−30.5815	28.11
04000000	00040001, 00040003, 00040401, 00040403	−30.5816	28.11
02000000	00028000, 00028001, 00028200, 00028201	−30.5815	28.10
01000000	00014000, 00014100, 0001C000, 0001C100	−30.5815	28.10
00800000	80002000, 80002080, 80006000, 80006080	−30.5816	28.06
00400000	40001000, 40001040, 40003000, 40003040	−30.5815	28.11
00200000	20000800, 20000820, 20001800, 20001820	−30.5815	28.11
00100000	10000400, 10000410, 10000C00, 10000C10	−30.5815	28.11
00080000	08000200, 08000208, 08000600, 08000608	−30.5815	28.11
00040000	04000100, 04000104, 04000300, 04000304	−30.5816	28.10
00020000	02000080, 02000082, 02000180, 02000182	−30.5815	28.11
00010000	01000040, 01000041, 010000C0, 010000C1	−30.5814	28.11

To add more rounds in the backwards direction, we must guess the bit

$$(F(X_L^{i-1}))_0 = (X_L^{i-1})_{14} \oplus ((X_L^{i-1})_{15} \& (X_L^{i-1})_8).$$

On the other hand, to determine $(F(X_L^{i-1}))_0$ one should guess $(X_L^{i-1})_{14}$ and $(X_L^{i-1})_{15}$ only if the guessed value for $(X_L^{i-1})_8$ is 1. So, in average we need one bit guess for $(X_L^{i-1})_{15}$ and $(X_L^{i-1})_8$ (in Figure 5.11 such bits are indicated in blue).

5.5.3.4.2 In The Forward Direction

We can use the same approach to add some rounds to the end of linear hull in the cost of guessing some bits of subkeys. More details are depicted in Figure 5.11.

5.5.3.4.3 Attack Complexity

We require $2^{30.5593}$ known plaintexts. We also need $2^{30.5593}$ encryptions for producing the required known plaintexts and $2^{30.5593}2^{25}$ encryptions to find the related key bits of the extended rounds. Therefore, the time complexity of the attack is

$$2^{30.5593} + 2^{30.5593}2^{25} \approx 2^{55.56}.$$

The summary of the results provided in this section can be seen in the following table.

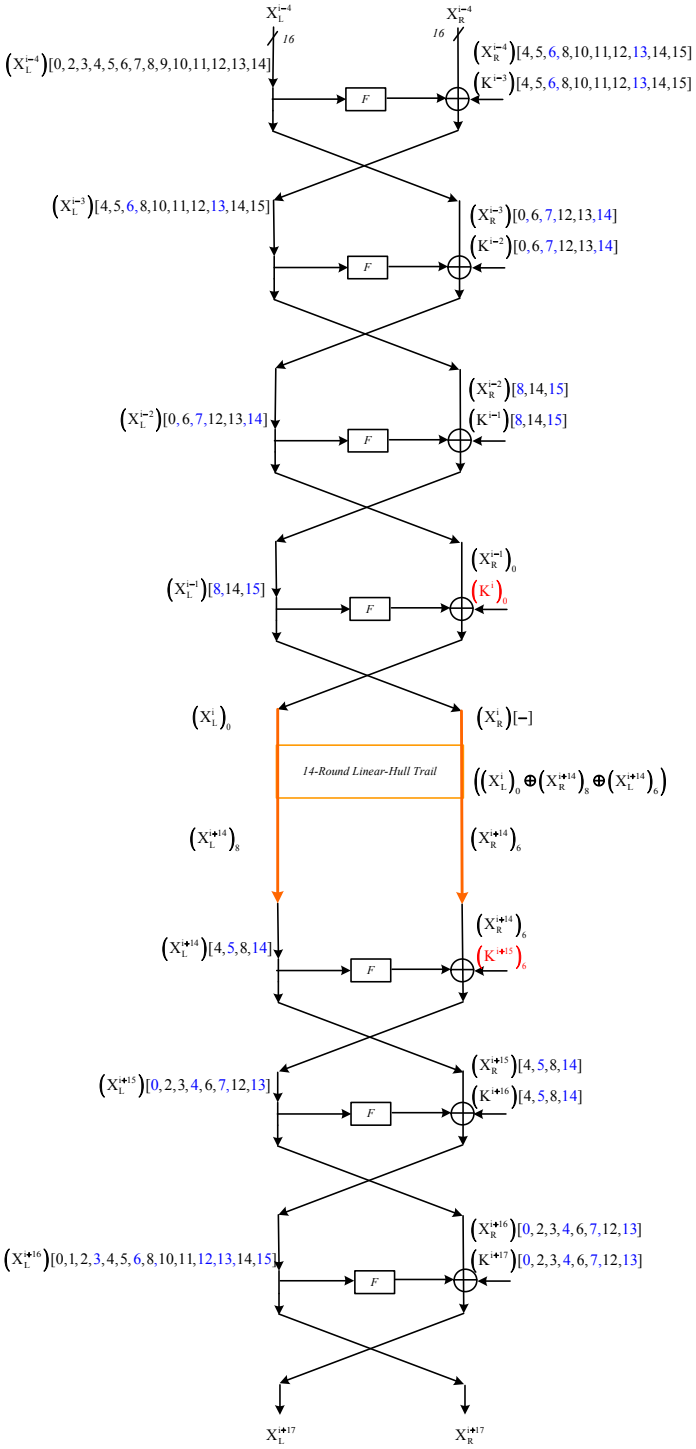


Figure 5.11: The subkey bits (in black) that should be guessed to attack 20 rounds of SIMON 32/64. The red bits are not required to be guessed and the blue bits cost guessing a half bit on average.

Table 5.16: Linear cryptanalysis of SIMON using Matsui's Algorithm 1 and 2, and linear hulls

	SIMON	# Rounds	Data	Time
Matsui's Algorithm 1 [16]	32/64	13	2^{32}	2^{32}
Matsui's Algorithm 2	32/64	17	2^{32}	$2^{59.5}$
	48/72	19	2^{46}	2^{70}
	48/96	20	2^{46}	$2^{86.5}$
	64/96	22	2^{58}	2^{91}
	64/128	23	2^{58}	2^{108}
	96/144	34	2^{94}	$2^{136.5}$
	128/192	40	2^{128}	$2^{176.5}$
	128/256	42	2^{128}	$2^{235.5}$
Linear Hull	32/64	21	$2^{31.69}$	$2^{63.69}$
	32/64	21	$2^{30.56}$	$2^{55.56}$
	48/72	20	$2^{44.11}$	$2^{70.61}$
	48/96	21	$2^{44.11}$	$2^{87.11}$
	64/96	27	$2^{62.53}$	$2^{88.53}$
	64/128	29	$2^{62.53}$	$2^{123.53}$
	96/144	36	$2^{94.2}$	$2^{135.2}$
	128/192	48	$2^{126.6}$	$2^{187.6}$
	128/256	50	$2^{126.6}$	$2^{242.6}$

5.6 Related Work

Since the publication of the specifications of SIMON a wave of cryptanalysis results were presented including what was presented in this Chapter.

In [13, 14], Abed et al. presented analysis of SIMON using various cryptanalytic techniques including linear-, differential-, impossible differential- and rectangular attacks. In the direction of differential cryptanalysis, the authors presented differential attacks on reduced-round versions of all SIMON variants. In the direction of impossible differential analysis, attacks are presented on 13 out of 32 rounds for SIMON 32/64 with data and time complexities 2^{30} respectively $2^{50.1}$, and up to 25 out of 72 rounds for SIMON 128/256 with data and time complexities 2^{119} respectively 2^{195}. With respect to linear cryptanalysis, [14] presented key-recovery attacks on variants of SIMON reduced to 11, 14, 16, 20 and 23 rounds for the respective block sizes of 32, 48, 64, 96 and 128 bits respectively.

Later, Alizadeh et al. [16] improved linear cryptanalysis of SIMON and presented attacks on 13-round SIMON32, 16-round SIMON48, 19-round SIMON64, 29-round SIMON96 and 36 round SIMON128/128.

In [40], Biryukov et al. presented a method for searching for differentials in ARX ciphers. The authors apply the method to SIMON and improve the previous differential characteristics to present attacks on 18 out of 32 rounds for SIMON 32/64 and up to 26 out of 44 rounds for SIMON 64/128.

In [54] the authors presented an improved generic version of impossible differential attacks on

SIMON that can attack all the variants of simon for different number of rounds for example 19 rounds for SIMON32/64 and 30 rounds for Simon128/256.

Most recently, Wang et al. [197] improved the known results on differential cryptanalysis of SIMON and presented attacks on 21-round SIMON32/64, 22-round SIMON48/72, 22-round SIMON48/96, 28-round SIMON64/96 and SIMON64/128. In [198], Wang *et al.* also improved results for SIMON32/64 for impossible differential cryptanalysis to 18 rounds for data and time complexities of 2^{32} and 2^{61} respectively. Other attack vectors are also presented; zero-correlation attacks are applied to 20 rounds with data and time complexities 2^{32} respectively $2^{56.9}$ and integral cryptanalysis techniques to 21 rounds with data and time complexities of 2^{31} respectively 2^{63}.

5.7 Conclusion

In this chapter we have analysed the security of SIMON against differential and variants of linear cryptanalysis, i.e. classical- as well as linear hull attacks.

For differential cryptanalysis, we have determined iterative differentials for Simon32/64, and general differentials for all variants of SIMON, that yield differential attacks on reduced versions with at least half the total rounds of the cipher in all cases. This analysis provided the grounds for our best results. An interesting observation in Section 5.3.4 is that Simon32/64 exhibits a strong differential effect. This suggests that bounding the expected differential probability (EDP) by the expected maximum characteristic probability is not well-founded in this case. Furthermore, we considered using truncated differentials to construct impossible differentials over a number of rounds, which yielded a distinguisher on reduced versions of most of the cipher variants, however it can not be to launch a practical attack as we have shown in the related sections that it yields high complexity. These results are summarized in Table 5.9. Our differential and impossible differentials use the simple assumption of a chosen plaintext setting, i.e. no chosen ciphertext oracle is required, nor is any known- chosen- or related-key settings used.

As for linear attacks, we mainly used a connection between linear- and differential characteristics and extended it to a connection between linear hulls and differentials. Given these connections, we used the known results on differential cryptanalysis on SIMON variants to present the best known results on SIMON using linear cryptanalysis.

Furthermore, we have investigated the linear hull effect on SIMON32/64 using the correlation matrix of the average squared correlations. Utilizing this technique, we achieve a lower time and data complexity than other attack variants by having a key recovery attack on 21-round SIMON32/64 with data complexity $2^{30.56}$ and time complexity $2^{55.6}$.

Chapter 6

Links among Integral, Impossible Differential and Zero-Correlation Linear Cryptanalysis

As we have established so far, block ciphers are considered vital elements in constructing many symmetric cryptographic schemes such as encryption algorithms, hash functions, authentication schemes and pseudo-random number generators. The core security of these schemes depends on the resistance of the underlying block ciphers to known cryptanalytic techniques and new dedicated ones. So far a variety of cryptanalytic techniques has been proposed such as differential cryptanalysis [37], linear cryptanalysis [152], differential-linear cryptanalysis [60,183], truncated differential cryptanalysis [130], impossible differential cryptanalysis [**?**,126], multidimensional linear cryptanalysis [103], zero-correlation linear cryptanalysis [7], integral cryptanalysis [128], statistical saturation cryptanalysis [68], interpolation attack [188] and so on. In this chapter, we will focus on cryptanalytic attacks of impossible, integral and zero-correlation cryptanalysis that we have briefly discussed in Chapter 3.

Along with the growing of the list of cryptanalytic tools, the question whether there are direct links or any connection between different tools has drawn much attention of the cryptographic research community, since such relations can be used to compare the effectiveness of different tools as well as improve cryptanalytic results of block ciphers.

Efforts to find and build the links among different cryptanalytic techniques were initiated by Chadaud and Vaudenay in [66], where a theoretical link between differential and linear cryptanalysis was presented. After that, many attempts have been made to establish further relations among various cryptanalytic tools. In [23], Sun *et al.* proved that from an algebraic view, integral cryptanalysis can be seen as a special case of the interpolation attack. In [142], Leander stated that statistical saturation distinguishers are averagely equivalent to multidimensional linear distinguishers. In [8], Bogdanov *et al.* showed that an integral implies a zero-correlation linear hull unconditionally, a zero-correlation linear hull indicates an integral distinguisher under certain conditions, and a zero-correlation linear hull is actually a special case of multidimensional linear distinguishers. In [45], Blondeau *et al.* further analyzed the link between differential and linear cryptanalysis and demonstrated some new insights on this link to make it more applicable in practice. This link was later applied in [60] to provide an exact expression of the bias of a differential-linear approximation. Moreover, they claimed that the existence of a zero-correlation linear hull is equivalent to the existence of an impossible differential in some specific cases.

However, as shown in [58], this link is usually not practical for most known impossible differential or zero correlation linear distinguishers, since the sum of the dimensions of inputs and outputs of these two distinguishers are always be block size of the cipher, which means if the dimension parameters for one type are small, they should be infeasible large for the other type. Blondeau *et al.* proposed a practical relation between these two distinguishers for Feistel-type and Skipjack-type ciphers and showed some equivalence between impossible differential and zero correlation linear cryptanalysis with respect to Feistel-type and Skipjack-type ciphers. In [59], Blondeau and Nyberg showed that statistical saturation cryptanalysis is indeed equivalent to truncated differential attack, which led to the link between multidimensional linear distinguisher and truncated differential.

6.1 Our Contributions.

Whilst of the results that have been presented on the different links of cryptanalytic methods are interesting, the links between impossible differential and integral cryptanalysis is still missing. In this chapter we aim to explore the relations between these two approaches. The main contribution can be summarized as follows:

- we propose the definition of *structure* $\mathcal{E}$, which is a set containing some ciphers that are "similar" with each other. Then, by introducing the *dual structure* $\mathcal{E}^{\perp}$, we prove that $a \to b$ is an impossible differential of $\mathcal{E}$ if and only if it is a zero correlation linear hull of $\mathcal{E}^{\perp}$. More specifically, let P^T and P^{-1} denote the transpose and inverse of P respectively. Then for a Feistel structure with SP-type round functions where P is invertible, denoted as $\mathcal{F}_{SP}$, constructing an r-round zero correlation linear hull is equivalent to constructing an impossible differential of $\mathcal{F}_{SP^T}$, which is the same structure as $\mathcal{F}_{SP}$ with P^T instead of P; Based on this result, we find 8-round zero correlation linear hulls of Camellia without FL/FL^{-1} layer.
- We find that a zero correlation linear hull always implies the existence of an integral distinguisher. This observation also provides a novel way for constructing integral distinguisher. Meanwhile, we observe that the statement "*integral unconditionally implies zero correlation linear hull*" in [**?**] is correct only under the definition that integral property is a balanced vectorial boolean function, while it does not hold for the general case. For example, denote by $\text{AES}^{(4)}$ the 4-round AES with MixColumns, up to date, we could not build any zero correlation linear hull of $\text{AES}^{(4)}$ from the integral distinguisher of $\text{AES}^{(4)}$ [7, 73].
- Then following the results given above, we build the link between impossible differential cryptanalysis and integral cryptanalysis, i.e., an r-round impossible differential of a structure $\mathcal{E}$ always implies the existence of an r-round integral distinguisher of $\mathcal{E}^{\perp}$. Moreover, in the case that $\mathcal{E}^{\perp} = A_2\mathcal{E}A_1$ where A_1 and A_2 are linear transformations, we could get direct links between impossible differential cryptanalysis and integral cryptanalysis of $\mathcal{E}$.
  - We improve the integral distinguishers of Feistel structures by 1 round. We propose an integral distinguishers of 5-round Feistel structure with bijective round functions and 3-round Feistel structure with round functions not necessarily being bijective.

– The best known integral distinguishers of Camellia so far, i.e., 7-round integral distinguishers of Camellia with FL/FL^{-1} layer and 8-round integral distinguishers of Camellia without FL/FL^{-1} layer.

It is worth mentioning that the results presented in this chapter is part of [38].

6.2 Preliminaries

6.2.1 Boolean Functions

This section recalls the notations and concepts [64] which will be used throughout this paper.

Let $\mathbb{F}_2$ denote the finite fields with two elements, and $\mathbb{F}_2^n$ be the vector space over $\mathbb{F}_2$ with dimension n. Let $a = (a_1, \ldots, a_n), b = (b_1, \ldots, b_n) \in \mathbb{F}_2^n$. Then

$$a \cdot b \triangleq a_1 b_1 \oplus \cdots \oplus a_n b_n$$

denotes the *inner product* of a and b. Note that the inner product of a and b can be written as ab^T where b^T stands for the *transpose* of b and the multiplication is defined as matrix multiplication. Given a function $G : \mathbb{F}_2^n \to \mathbb{F}_2$, the *correlation* of G is defined by

$$c(G(x)) \triangleq \frac{\#\{x \in \mathbb{F}_2^n | G(x) = 0\} - \#\{x \in \mathbb{F}_2^n | G(x) = 1\}}{2^n} = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{G(x)}.$$

Given a vectorial function $H : \mathbb{F}_2^n \to \mathbb{F}_2^k$, the *correlation* of the linear approximation for a k-bit output mask b and an n-bit input mask a is defined by

$$c(a \cdot x \oplus b \cdot H(x)) \triangleq \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x \oplus b \cdot H(x)}.$$

If $c(a \cdot x \oplus b \cdot H(x)) = 0$, then $a \to b$ is called a *zero correlation linear hull* of H [7]. This definition can be extended as follows: Let $A \subseteq \mathbb{F}_2^n$, $B \subseteq \mathbb{F}_2^k$. If for all $a \in A$, $b \in B$, $c(a \cdot x \oplus b \cdot H(x)) = 0$, then $A \to B$ is called a *zero correlation linear hull* of H. In the case that H is a permutation on $\mathbb{F}_2^n$, for any $b \neq 0$, $c(b \cdot H(x)) = 0$ and for any $a \neq 0$, $c(a \cdot x) = 0$, we call $0 \to b$ and $a \to 0$ trivial zero correlation linear hulls of H where $a \neq 0$ and $b \neq 0$. Let $A \subseteq \mathbb{F}_2^n$. If the size of the set

$$H_A^{-1}(y) \triangleq \{x \in A | H(x) = y\}$$

is independent of $y \in \mathbb{F}_2^k$, we say H is *balanced on* A. Specifically, if $A = \mathbb{F}_2^n$, we say H is a *balanced function*. If the sum of all images of H is 0, i.e.

$$\sum_{x \in \mathbb{F}_2^n} H(x) = 0,$$

we say H has an *integral-balanced (zero-sum)* property [**?**]. Let $\delta \in \mathbb{F}_2^n$ and $\Delta \in \mathbb{F}_2^k$. The differential probability of $\delta \to \Delta$ is defined as

$$p(\delta \to \Delta) \triangleq \frac{\#\{x \in \mathbb{F}_2^n | H(x) \oplus H(x \oplus \delta) = \Delta\}}{2^n}.$$

If $p(\delta \to \Delta) = 0$, then $\delta \to \Delta$ is called an *impossible differential* of H [**?**, **?**]. Let $A \subseteq \mathbb{F}_2^n$, $B \subseteq \mathbb{F}_2^k$. If for all $a \in A$ and $b \in B$, $p(a \to b) = 0$, $A \to B$ is called an *impossible differential* of H.

We recall the following property of balanced boolean functions: a function $G : \mathbb{F}_2^n \to \mathbb{F}_2$ is balanced if and only if $c(G(x)) = 0$.

6.2.2 Feistel Scheme Based Ciphers

Since In this chapter we will focus on providing the results for Feistel ciphers. We will briefly revisit the definition introduced in the Chapter 3. In this chapter, we will focus on the case that F_i's are SP-type functions which will be defined in the following.

An r-round Feistel cipher E is defined as follows:

Let $(L_0, R_0) \in \mathbb{F}_2^{2n}$ be the input of E. Iterate the following transformation r times:

$$\begin{cases} L_{i+1} = F_i(L_i) \oplus R_i \\ R_{i+1} = L_i \end{cases} \quad 0 \leq i \leq r-1,$$

where $L_i, R_i \in \mathbb{F}_2^n$. The output of the r-th iteration is defined as the output of E.

An SP-type round function will be defined as $f : \mathbb{F}_2^{sb} \to \mathbb{F}_2^{sb}$ and it details will be as the following:

Assume the input x is divided into b pieces $x = (x_0, \ldots, x_{b-1})$, and each of the x_i's is an s-bit word. Then apply the nonlinear transformation S_i which is the confusion layer usually an S-box or modular addition to x_i and let $y = (S_0(x_0), \ldots, S_{b-1}(x_{b-1})) \in \mathbb{F}_2^{sb}$. Then, a linear transformation or what we usually refer to as permutation P is applied to y. Hence, Py is the output of f.

The main approaches to design the non-linear permutation layer can be summarized as the following:

(1) P is a bit-wise permutation of $\mathbb{F}_2^{st}$ as in PRESENT [4].

(2) Each bit of Py is a sum of some bits of y as in PRINCE [52].

(3) Each word of Py is a sum of some words of y as in Camellia [119]. The transformation P could be written as follows:

$$P = \begin{pmatrix} E & 0 & E & E & 0 & E & E & E \\ E & E & 0 & E & E & 0 & E & E \\ E & E & E & 0 & E & E & 0 & E \\ 0 & E & E & E & E & E & E & 0 \\ E & E & 0 & 0 & 0 & E & E & E \\ 0 & E & E & 0 & E & 0 & E & E \\ 0 & 0 & E & E & E & E & 0 & E \\ E & 0 & 0 & E & E & E & E & 0 \end{pmatrix}$$

where E and 0 denote 88 identity and zero matrices, respectively.

(4) Each word of Py, seen as an element of some extension fields of $\mathbb{F}_2$, is a linear combination of some other words of y as in the AES.

It is note worthy that whatever linear transformation a cipher adopts, it is always linear over $\mathbb{F}_2$. Consequently, the permutation P can always be expressed as a multiplication by a matrix which leads to the following definition:

Definition 10. *Let P be a linear transformation over $\mathbb{F}_2^m$ for some positive integer m. The matrix representation of P over $\mathbb{F}_2$ is called the* primitive representation *of P.*

6.2.2.1 Camellia

The block cipher Camellia [119] was proposed by NTT and Mitsubishi Electric Corporations in 2000. It was recommended in the NESSIE block cipher portfolio in 2003 and adopted as a new international standard by ISO/IEC in 2005. Camellia is a 128-bit block cipher which adopts the Feistel structure with key-dependent functions FL/FL^{-1} inserted every 6 rounds. It supports variable key sizes and the number of rounds depends on the key size. The round function uses an SP-type structure, where the nonlinear transformation adopts S-boxes and the linear transformation can be defined as a binary matrix P. Please refer to [119] for the details of Camellia.

6.2.3 Structure and Dual Structure

In this section we will introduce three main definitions that we will use in establishing the links between integral, impossible differentials and zero-correlation linear hulls in the following sections.

Generally, when constructing a zero or impossible differentials we are interested in the fact that a difference exist on the function understudy (i.e. the substitution function S-box no matter what are the actual values. This should indicated that if the round functions used a different S-box the distinguishers shall hold as in AES in [7,95] and Camellia in [9,195]. Thus, the definitions of structure and dual structure will be as the following:

Definition 11. *Let $E : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher with bijective S-boxes as the basic non-linear components.*

(1) *A structure $\mathcal{E}^E$ on $\mathbb{F}_2^n$ is defined as a set of block ciphers E' which is exactly the same as E except that the S-boxes can take all possible bijective transformations on the corresponding domains.*

(2) *Let $a, b \in \mathbb{F}_2^n$. If for any $E' \in \mathcal{E}^E$, $a \to b$ is an impossible differential (zero correlation linear hull) of E', $a \to b$ is called an impossible differential (zero correlation linear hull) of $\mathcal{E}^E$.*

Nevertheless, to generalize this definition if S-boxes used in E are not necessarily bijective, then $\mathcal{E}^E$ could be defined as a set of block ciphers E' which is exactly the same as E except

that the S-boxes can take all possible transformations on the corresponding domains. As discussed above, the truncated impossible differentials and zero correlation linear hulls of AES and Camellia found so far are actually the impossible differentials and zero correlation linear hulls of $\mathcal{E}^{\text{AES}}$ and $\mathcal{E}^{\text{Camellia}}$.

Definition 12. *Let $\mathcal{F}_{SP}$ be a* Feistel structure *with SP-type round function, and let the primitive representation of the linear transformation be P. Let σ be the operation that exchanges the left and right halves of a state. Then the* dual structure *$\mathcal{F}_{SP}^{\perp}$ of $\mathcal{F}_{SP}$ is defined as $\sigma\mathcal{F}_{P^TS}\sigma$.*

6.3 Links among Integral, Impossible Differential and Zero-Correlation Linear Hulls

6.3.1 Links between Integral and Zero-Correlation Linear Hull

The main goal of this section is to prove that a zero-correlation linear hull always implies an integral distinguisher. Firstly, we will give a generalization of Lemma 1 in [8]. This statement will give a foundation for the link between integral and zero-correlation distinguisher as follows:

Theorem 6.3.1. *Let A be a subspace of $\mathbb{F}_2^n$, $A^{\perp} = \{x \in \mathbb{F}_2^n | a \cdot x = 0, a \in A\}$ be the dual space of A and $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a function on $\mathbb{F}_2^n$. For any $\lambda \in \mathbb{F}_2^n$, $T_\lambda : A^{\perp} \to \mathbb{F}_2^n$ is defined as $T_\lambda(x) = F(x \oplus \lambda)$, then for any $b \in \mathbb{F}_2^n$,*

$$\sum_{a \in A} (-1)^{a \cdot \lambda} c(a \cdot x \oplus b \cdot F(x)) = c(b \cdot T_\lambda(x)).$$

Proof.

$$\begin{aligned}
&\sum_{a \in A} (-1)^{a \cdot \lambda} c(a \cdot x \oplus b \cdot F(x)) = \sum_{a \in A} (-1)^{a \cdot \lambda} \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x \oplus b \cdot F(x)} \\
= \;& \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F(x)} \sum_{a \in A} (-1)^{a \cdot (\lambda \oplus x)} = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F(x)} |A| \delta_{A^{\perp}}(\lambda \oplus x) \\
= \;& \frac{1}{|A^{\perp}|} \sum_{y \in A^{\perp}} (-1)^{b \cdot T_\lambda(y)} = c(b \cdot T_\lambda(x)),
\end{aligned}$$

where $\delta_{A^{\perp}}(x) = \begin{cases} 1 & x \in A^{\perp} \\ 0 & x \notin A^{\perp} \end{cases}$ □

The second theorem follows :

Theorem 6.3.2. *Let A be a subspace of $\mathbb{F}_2^n$, $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $T_\lambda : A^{\perp} \to \mathbb{F}_2^n$ be defined as $T_\lambda(x) = F(x \oplus \lambda)$ where $\lambda \in \mathbb{F}_2^n$, then for any $b \in \mathbb{F}_2^n$,*

$$\frac{1}{2^n} \sum_{\lambda \in \mathbb{F}_2^n} (-1)^{b \cdot F(\lambda)} c(b \cdot T_\lambda(x)) = \sum_{a \in A} c^2(a \cdot x \oplus b \cdot F(x)).$$

This can be proved as follows:

$$\begin{aligned}
\sum_{a\in A} c^2(a\cdot x\oplus b\cdot F(x)) &= \sum_{a\in A}\frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n}(-1)^{a\cdot x\oplus b\cdot F(x)}\frac{1}{2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{a\cdot\lambda\oplus b\cdot F(\lambda)}\\
&= \frac{1}{2^{2n}}\sum_{x\in\mathbb{F}_2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(x)\oplus b\cdot F(\lambda)}\sum_{a\in A}(-1)^{a\cdot x\oplus a\cdot\lambda}\\
&= \frac{1}{2^{2n}}\sum_{x\in\mathbb{F}_2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(x)\oplus b\cdot F(\lambda)}|A|\delta_{A^\perp}(x\oplus\lambda)
\end{aligned}$$

Let $\theta = x\oplus\lambda$. Since $|A||A^\perp| = 2^n$, we have

$$\begin{aligned}
&\frac{1}{2^{2n}}\sum_{x\in\mathbb{F}_2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(x)\oplus b\cdot F(\lambda)}|A|\delta_{A^\perp}(x\oplus\lambda)\\
=\ &\frac{|A|}{2^{2n}}\sum_{\theta\oplus\lambda\in\mathbb{F}_2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(\theta\oplus\lambda)\oplus b\cdot F(\lambda)}\delta_{A^\perp}(\theta) = \frac{1}{2^n|A^\perp|}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(\lambda)}\sum_{\theta\oplus\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(\theta\oplus\lambda)}\delta_{A^\perp}(\theta)\\
=\ &\frac{1}{2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(\lambda)}\frac{1}{|A^\perp|}\sum_{\theta\in A^\perp}(-1)^{b\cdot F(\theta\oplus\lambda)} = \frac{1}{2^n}\sum_{\lambda\in\mathbb{F}_2^n}(-1)^{b\cdot F(\lambda)}c(b\cdot T_\lambda(x)).
\end{aligned}$$

For general integral distinguishers, $c(b\cdot T_\lambda(x))$ may not necessarily be 0, therefore the conclusion that integral unconditionally implies zero-correlation linear hull in [8] is correct only under their definition of integral while it may not hold for general ones.

From Theorem 6.3.1, we can deduce the following:

Corollary 6.3.3. *Let $F:\mathbb{F}_2^n\to\mathbb{F}_2^n$ be a function on $\mathbb{F}_2^n$, A be a subspace of $\mathbb{F}_2^n$ and $b\in\mathbb{F}_2^n\backslash\{0\}$. Suppose that $A\to b$ is a zero-correlation linear hull of F, then for any $\lambda\in\mathbb{F}_2^n$, $b\cdot F(x\oplus\lambda)$ is balanced on $A^\perp$.*

This Corollary states that if the input masks of a zero-correlation linear hull form a subspace, then a zero-correlation linear hull implies an integral distinguisher. Furthermore, the condition that input masks form a subspace can be removed, which leads to the following result:

Theorem 6.3.4. *A nontrivial zero-correlation linear hull of a block cipher always implies the existence of an integral distinguisher.*

Proof. Assume that $A\to B$ is a non-trivial zero-correlation linear hull of a block cipher E. Then we can choose $0\neq a\in A, 0\neq b\in B$, such that $\{0,a\}\to b$ is also a zero-correlation linear hull of E.

Since $V=\{0,a\}$ forms a subspace on $\mathbb{F}_2$, according to Corollary 6.3.3, $b\cdot E(x)$ is balanced on $V^\perp$. This implies an integral distinguisher of E. □

Moreover, in the proof of Theorem 6.3.4, we can always assume that $0\in A$. Then

(1) If A forms a subspace, an integral distinguisher can be constructed from $A \to b$;

(2) If A does not form a subspace, we can choose some $A_1 \subset A$ such that A_1 forms a subspace, then an integral distinguisher can be constructed from $A_1 \to b$.

It was stated in [8] that a zero-correlation linear hull indicates the existence of an integral distinguisher under certain conditions, while Theorem 6.3.4 shows that these conditions can be removed, which results in a more applicable and practical link between zero-correlation linear hull and integral distinguisher.

It can be seen that Theorem 6.3.4 also gives us a new approach to find integral distinguishers of block ciphers. More specifically, an r-round zero-correlation linear hull can be used to construct an r-round integral distinguisher.

6.3.2 Links between Impossible Differential and Zero-Correlation Linear Hull

In this section, we will show the equivalence between impossible differentials and zero correlation linear hulls of a structure, which will be used to establish the link between impossible differential and integral cryptanalysis in Sec.**??**.

Theorem 6.3.5. *$a \to b$ is an r-round impossible differential of $\mathcal{F}_{SP}$ if and only if it is an r-round zero correlation linear hull of $\mathcal{F}_{SP}^{\perp}$.*

Proof. The proof can be divided into the following two parts (See Fig.6.1):

Part (I) In this part, we are going to prove that for $(\delta_0, \delta_1) \to (\delta_r, \delta_{r+1})$, if one can find $E \in \mathcal{F}_{SP}^{\perp}$ such that $c((\delta_0, \delta_1) \cdot x \oplus (\delta_r, \delta_{r+1}) \cdot E(x)) \neq 0$, then one can find $E' \in \mathcal{F}_{SP}$ such that $p((\delta_1, \delta_0) \to (\delta_{r+1}, \delta_r)) > 0$.

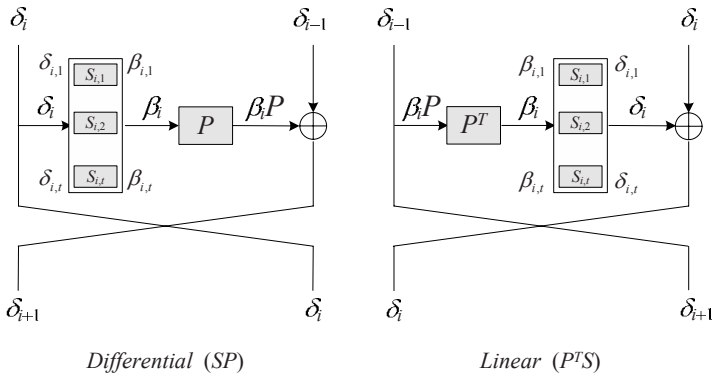


Figure 6.1: Differential Propagation of $\mathcal{F}_{SP}$ and Linear Propagation of $\mathcal{F}_{SP}^{\perp}$

Assume that $(\delta_0, \delta_1) \to (\delta_r, \delta_{r+1})$ is a linear hull with non-zero correlation for some $E \in \mathcal{F}_{SP}^{\perp}$, and the input to the round function could be divided into t pieces, each of which is an s-bit word. Then there exists a linear characteristic with non-zero correlation:

$$(\delta_0, \delta_1) \to \cdots (\delta_{i-1}, \delta_i) \to \cdots \to (\delta_r, \delta_{r+1}),$$

where $\delta_i \in (\mathbb{F}_2^s)^t$. In this characteristic, let the output mask of $S_i = (S_{i,1}, \ldots, S_{i,t})$ be $\delta_i = (\delta_{i,1}, \ldots, \delta_{i,t}) \in (\mathbb{F}_2^s)^t$, and let the input mask of S_i be $\beta_i = (\beta_{i,1}, \ldots, \beta_{i,t}) \in (\mathbb{F}_2^s)^t$. Since for $\gamma \neq \beta_i P$, $c(\gamma \cdot x \oplus \beta_i \cdot (xP^T)) = 0$, $\delta_{i+1} = \delta_{i-1} \oplus \beta_i P$.

In the following, for any $(x_L, x_R) = (x_{L,1}, \ldots, x_{L,t}, x_{R,1}, \ldots, x_{R,t}) \in (\mathbb{F}_2^s)^t (\mathbb{F}_2^s)^t$, we will construct an r-round cipher $E_r \in \mathcal{F}_{SP}$, such that $E_r(x_L, x_R) \oplus E_r(x_L \oplus \delta_1, x_R \oplus \delta_0) = (\delta_{r+1}, \delta_r)$.

If $r = 1$, for $j \in \{1, \ldots, t\}$: if $\delta_{1,j} = 0$, we can define $S_{1,j}$ as any possible transformation on $\mathbb{F}_2^s$, and if $\delta_{1,j} \neq 0$, we can define

$$S_{1,j}(x_{L,j}) = x_{L,j}, \quad S_{1,j}(x_{L,j} \oplus \delta_{1,j}) = x_{L,j} \oplus \beta_{1,j},$$

then for $E_1 \in \mathcal{F}_{SP}$ which adopts such S-boxes,

$$E_1(x_L, x_R) \oplus E_1(x_L \oplus \delta_1, x_R \oplus \delta_0) = (\delta_0 \oplus \beta_1 P, \delta_1) = (\delta_2, \delta_1).$$

Suppose that we have constructed E_{r-1} such that $E_{r-1}(x_L, x_R) \oplus E_{r-1}(x_L \oplus \delta_1, x_R \oplus \delta_0) = (\delta_r, \delta_{r-1})$. Denote by $(y_L, y_R) = (y_{L,1}, \ldots, y_{L,t}, y_{R,1}, \ldots, y_{R,t})$ the output of $E_{r-1}(x_L, x_R)$. Then in the r-th round, if $\delta_{r,j} = 0$, we can define $S_{r,j}$ as any possible transformation on $\mathbb{F}_2^s$, otherwise, define $S_{r,j}$ as follows:

$$S_{r,j}(y_{L,j}) = y_{L,j}, \quad S_{r,j}(y_{L,j} \oplus \delta_{r,j}) = y_{L,j} \oplus \beta_{r,j}.$$

Therefore $E_r(x_L, x_R) \oplus E_r(x_L \oplus \delta_1, x_R \oplus \delta_0) = (\delta_{r-1} \oplus \beta_r P, \delta_r) = (\delta_{r+1}, \delta_r)$.

Part (II) In this part, we will prove that for $(\delta_1, \delta_0) \to (\delta_{r+1}, \delta_r)$, if one can find some $E \in \mathcal{F}_{SP}$ such that $p((\delta_1, \delta_0) \to (\delta_{r+1}, \delta_r)) > 0$, one can find some $E' \in \mathcal{F}_{SP}^{\perp}$ such that $c((\delta_0, \delta_1) \cdot x \oplus (\delta_r, \delta_{r+1}) \cdot E'(x)) \neq 0$.

Assume that $(\delta_1, \delta_0) \to (\delta_{r+1}, \delta_r)$ is a differential of $E \in \mathcal{F}_{SP}$. Then there exists a differential characteristic with positive probability:

$$(\delta_1, \delta_0) \to \cdots (\delta_{i+1}, \delta_i) \to \cdots \to (\delta_{r+1}, \delta_r),$$

where $\delta_i \in (\mathbb{F}_2^s)^t$. In this characteristic, the input difference of $S_i = (S_{i,1}, \ldots, S_{i,t})$ is $\delta_i = (\delta_{i,1}, \ldots, \delta_{i,t}) \in (\mathbb{F}_2^s)^t$, and let the output difference of S_i be $\beta_i = (\beta_{i,1}, \ldots, \beta_{i,t}) \in (\mathbb{F}_2^s)^t$, then $\delta_{i+1} = \delta_{i-1} \oplus (\beta_i P)$.

Taking the following fact into consideration: for $(\delta_{i,j}, \beta_{i,j})$, where $\delta_{i,j} \neq 0$, there always exists an ss binary matrix $M_{i,j}$ such that $\beta_{i,j} = \delta_{i,j} M_{i,j}^T$, then for $S_{i,j}(x) = xM_{i,j}$, $c(\beta_{i,j} \cdot x \oplus \delta_{i,j} \cdot S_{i,j}(x)) = 1$.

Now we are going to construct an r-round cipher $E_r \in \mathcal{F}_{SP}^{\perp}$ such that $c((\delta_0, \delta_1) \cdot x \oplus (\delta_r, \delta_{r+1}) \cdot E_r(x)) \neq 0$.

If $r = 1$, let $S_{1,j}(x) = xM_{1,j}$ for $\delta_{1,j} \neq 0$ and any linear transformation on $\mathbb{F}_2^s$ otherwise. Then all operations in $E_1 \in \mathcal{F}_{SP}^{\perp}$ are linear over $\mathbb{F}_2$, which implies that there exists a $2st2st$ binary matrix M_1 such that $E_1(x) = xM_1$, and

$$c((\delta_0, \delta_1) \cdot x \oplus (\delta_1, \delta_2) \cdot E_1(x)) = 1.$$

Assume that we have constructed $E_{r-1}(x) = xM_{r-1}$ with M_{r-1} being a $2st2st$ binary matrix such that

$$c((\delta_0, \delta_1) \cdot x \oplus (\delta_{r-1}, \delta_r) \cdot E_{r-1}(x)) = 1,$$

and we can define $S_{r,j}(x)$ in the r-th round similarly, then $E_r(x) = xM_r$ for some $2st2st$ binary matrix M_r, and

$$c((\delta_0, \delta_1) \cdot x \oplus (\delta_r, \delta_{r+1}) \cdot E_r(x)) = 1,$$

which ends our proof. □ □

Similarly, we can prove the following theorem:

Theorem 6.3.6. *$a \to b$ is an r-round impossible differential of $\mathcal{E}_{SP}$ if and only if it is an r-round zero correlation linear hull of $\mathcal{E}_{SP}^{\perp}$.*

From the proof of Theorem 6.3.5, we can see that when finding impossible differential of a structure, the only contradiction is that for some differential characteristic $\delta_1 \to \delta_2$ of S-box, $\delta_1 = 0, \delta_2 \neq 0$ or $\delta_1 \neq 0, \delta_2 = 0$ for invertible S-boxes and $\delta_1 = 0, \delta_2 \neq 0$ for non-invertible ones. Since otherwise, we can always construct an S-box such that $\delta_1 \to \delta_2$ is a possible differential. Therefore, we have the following corollary:

Corollary 6.3.7. *The method presented in [176] could find all impossible differentials of $\mathcal{F}_{SP}$ and $\mathcal{E}_{SP}$.*

As a matter of fact, this Corollary can be used in the provable security of block ciphers against impossible differential cryptanalysis. Since by the help of this Corollary, the longest impossible differentials of a given structure could be given.

In case P is invertible, according to equivalent structures defined in [136], we have

$$\mathcal{F}_{P^T S} = \left((P^T)^{-1}, (P^T)^{-1}\right) \mathcal{F}_{SP^T} \left(P^T, P^T\right), \tag{6.1}$$

which indicates:

Corollary 6.3.8. *Let $\mathcal{F}_{SP}$ be a Feistel structure with SP-type round function, and let the primitive representation of the linear transformation be P. If P is invertible, finding zero correlation linear hulls of $\mathcal{F}_{SP}$ is equivalent to finding impossible differentials of $\mathcal{F}_{SP^T}$.*

Example 1. (8-Round Zero Correlation Linear Hull of Camellia Without FL/FL^{-1}) Let Camellia* denote the cipher which is exactly the same as Camellia without FL/FL^{-1} layer except that P^T is used instead of P. Then we find that, for example:

$$((0,0,0,0,0,0,0,0),(0,0,0,0,a,0,0,0)) \to \quad ((0,0,0,0,0,0,0,h),(0,0,0,0,0,0,0,0))$$

is an 8-round impossible differential of Camellia*, where a and h denote any non-zero values. Therefore, we could derive an 8-round zero correlation linear distinguisher of Camellia without FL/FL^{-1} layer as shown below:

$$((a,a,0,0,a,0,a,a),(0,0,0,0,0,0,0,0)) \rightarrow \quad ((0,0,0,0,0,0,0,0),(h,0,0,h,0,h,h,h)).$$

Furthermore, if $\mathcal{F}_{SP} = \mathcal{F}_{SP^T}$ and $\mathcal{E}_{SP} = \mathcal{E}_{S(P^{-1})^T}$, the following corollary holds:

Corollary 6.3.9. *For a Feistel structure $\mathcal{F}_{SP}$ with SP-type round function, if P is invertible and $P = P^T$, there is a one-to-one correspondence between impossible differentials and zero correlation linear hulls.*

In the proof of Theorem 6.3.5, the S-boxes we constructed are not necessarily bijective. If we add the bijective condition, Theorem 6.3.5 still holds. Since for a bijective S-box, if the correlation is non-zero, $\delta_{1,j} \neq 0$ implies $\beta_{1,j} \neq 0$. Therefore, in Part(I) of the proof, we can further define $S_{1,j}$ as

$$S_{1,j}(x) = \begin{cases} x_{L,j} \oplus \delta_{1,j} & x = x_{L,j} \oplus \beta_{1,j}, \\ x_{L,j} \oplus \beta_{1,j} & x = x_{L,j} \oplus \delta_{1,j}, \\ x & \text{others}, \end{cases}$$

and a similar definition can also be given to $S_{r,j}$. In this case, the S-boxes are invertible. Moreover, for a bijective S-box, if the differential probability is positive, $\delta_{i,j} \neq 0$ implies $\beta_{i,j} \neq 0$, thus in Part (II) of the proof, we can always find a non-singular binary matrix $M_{i,j}$ such that $\beta_{i,j} = \delta_{i,j} M_{i,j}^T$.

Theorem 6.3.5 and 6.3.6 show some links between impossible differential and zero correlation linear hull of a structure $\mathcal{E}$ and the corresponding dual structure $\mathcal{E}^{\perp}$. However, it doesn't mean, for example, an impossible differential of a cipher $E \in \mathcal{E}$ indicates a zero correlation linear hull of another cipher $E' \in \mathcal{E}^{\perp}$, which could be distinguished from the definitions of impossible differential and zero correlation linear hull of a cipher and a structure, respectively.

6.4 New Integral Distinguishers for Block Ciphers

Based on the links among integral, impossible differential and zero-correlation linear hull proposed in Theorems 6.3.4 and 6.3.5, we can construct new integral distinguishers from some zero-correlation linear hulls of block ciphers. Following are some intriguing examples.

6.4.1 New Integral Distinguishers for a Feistel Structure

So far the longest integral distinguisher known for a Feistel structure with invertible round functions is 4-round integral distinguisher. However, we can derive a 5-round integral distinguisher of this structure in terms of Theorem 6.3.4.

Proposition 6.4.1. *Let $\mathcal{E}_r$ be an r-round Feistel structure defined as:*

$$\begin{cases} L_{i+1} = F_i(L_i) \oplus R_i \\ R_{i+1} = L_i \end{cases} \quad 0 \le i \le r-1,$$

where $L_i, R_i \in \mathbb{F}_2^n$. Suppose that F_i's are bijective, then for any $c \in \mathbb{F}_2^n$, $c \neq 0$, $c \cdot R_5$ is balanced on $\{(0,0),(c,0)\}^\perp$ with respect to $\mathcal{E}_5$.

As a matter of fact, for any $c \in \mathbb{F}_2^n$, $c \neq 0$, $(c,0) \to (0,c)$ is a zero-correlation linear hull of $\mathcal{E}_5$. Thus according to Theorem 6.3.4, we can construct an integral distinguisher of $\mathcal{E}_5$, i.e., let (L_0, R_0) take all values in $\{(0,0),(c,0)\}^\perp$, then $c \cdot R_5$ is balanced.

Specifically, let $c = (1,1,\ldots,1) \in \mathbb{F}_2^n$, then we have

$$\{(0,0),(c,0)\}^\perp = \{((x_1,\ldots,x_n),(x_{n+1},\ldots,x_{2n}))|x_i \in \mathbb{F}_2, \sum_{i=1}^{n} x_i = 0\}.$$

Let $R_5 = (R_{5,1},\ldots,R_{5,n})$, then we can derive that $\sum_{i=1}^{n} R_{5,i}$ is balanced on $\{(0,0),(c,0)\}^\perp$.

Similarly, we can construct 3-round integral distinguisher for Feistel structure with round functions not necessarily being bijective according to Example 1 and Theorem 6.3.4, while the previously best integral distinguisher of such structure only covers two rounds.

Proposition 6.4.2. *Let $\mathcal{E}_r$ be a Feistel structure as defined in Proposition 6.4.1, $c = (0,\ldots,0,1) \in \mathbb{F}_2^n$ and $V = \{((0,\ldots,0,x_n),(x_{n+1},\ldots,x_{2n}))|x_i \in \mathbb{F}_2\}$. For any F_i, $c \cdot R_3$ is balanced on V with respect to $\mathcal{E}_3$.*

6.4.2 New Integral Distinguishers for Camellia

In the following, we will present a series of 8-round integral distinguishers of Camellia without FL/FL^{-1} layer as well as 7-round integral distinguishers of Camellia with FL/FL^{-1} layer, which are the best integral distinguishers of Camellia found so far.

Proposition 6.4.3. *Let V be defined as*

$$V = \{((x_1,\ldots,x_8),(x_9,\ldots,x_{16}))|x_1 \oplus x_2 \oplus x_5 \oplus x_7 \oplus x_8 = 0, x_i \in \mathbb{F}_2^8\}.$$

For any $h \in \mathbb{F}_2^8$, $h \neq 0$, $(h,0,0,h,0,h,h,h) \cdot R_{i+8}$ is balanced on V with respect to 8-round Camellia without FL/FL^{-1} layer.

This can be demonstrated as follows. Firstly, Camellia adopts the Feistel structure $\mathcal{E}_{SP}$ with invertible round function as defined in Theorem 6.3.5 if not taking into account the FL/FL^{-1} layer. Let E denote the cipher which is exactly the same as Camellia except that P^T is used instead of P. We find that

$$\begin{array}{rl} & ((0,0,0,0,0,0,0,0),(0,0,0,0,a,0,0,0)) \\ \to & ((0,0,0,0,0,0,0,h),(0,0,0,0,0,0,0,0)) \end{array}$$

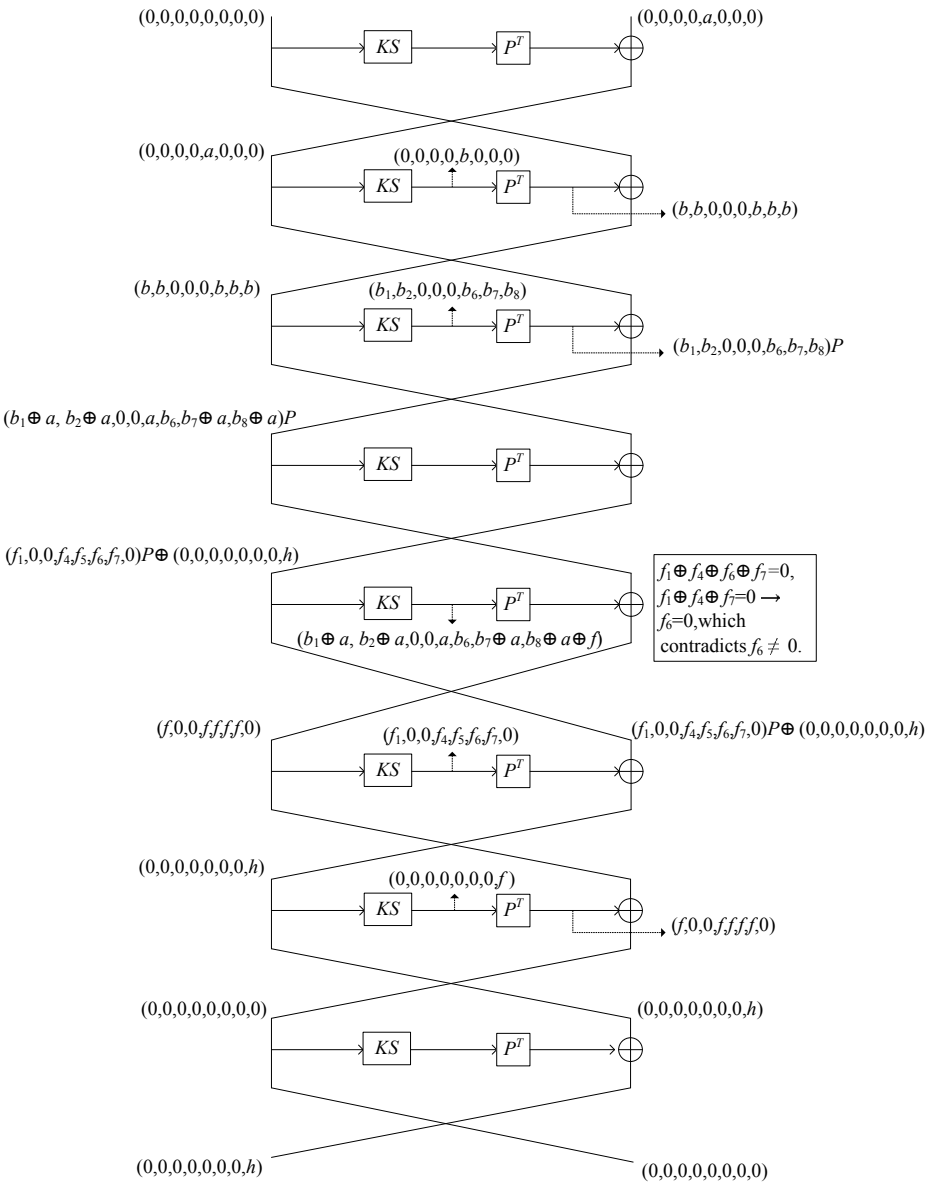


Figure 6.2: 8-round impossible differential of E

is an 8-round impossible differential of E (See Fig.6.2), where h denotes any non-zero value. Thus according to Theorem 6.3.5, we can derive an 8-round zero-correlation linear distinguisher of Camellia without FL/FL^{-1} layer as shown below:

$$\begin{aligned} & ((a,a,0,0,a,0,a,a),(0,0,0,0,0,0,0,0)) \\ \rightarrow \quad & ((0,0,0,0,0,0,0,0),(h,0,0,h,0,h,h,h)). \end{aligned}$$

Furthermore, let V denote the set

$$\{((a,a,0,0,a,0,a,a),(0,0,0,0,0,0,0,0))|a \in \mathbb{F}_2^8\},$$

then we have

$$V^{\perp} = \{((x_1,\ldots,x_8),(x_9,\ldots,x_{16}))|x_1 \oplus x_2 \oplus x_5 \oplus x_7 \oplus x_8 = 0, x_i \in \mathbb{F}_2^8\}.$$

Following Theorem 6.3.4 we conclude that for Camellia without FL/FL^{-1} layer, if (L_i, R_i) takes all values in the set $V^{\perp}$, $(h,0,0,h,0,h,h,h) \cdot R_{i+8}$ is balanced.

Proposition 6.4.4. *Let V be defined as*

$$\{((x_1,\ldots,x_8),(x_9,\ldots,x_{16}))|x_1 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_8 = 0, x_i \in \mathbb{F}_2^8\}.$$

Assume that the FL/FL^{-1} layer lies between the $(i+3)$-th round and the $(i+4)$-th round of Camellia. For any $h \in \mathbb{F}_2^8$, $h \neq 0$, $(h,0,0,h,0,h,h,h) \cdot R_{i+7}$ is balanced on V with respect to 7-round Camellia with FL/FL^{-1} layer.

In fact, Proposition 6.4.4 follows from the zero-correlation linear hull

$$\begin{aligned} & ((b,0,0,b,0,b,b,b),(0,0,0,0,0,0,0,0)) \\ \rightarrow \quad & ((0,0,0,0,0,0,0,0),(h,0,0,h,0,h,h,h)) \end{aligned}$$

proposed in [9] and Theorem 6.3.4, where h denotes any nonzero value in $\mathbb{F}_2^8$.

6.5 Conclusion

In this chapter we have provided an intuition into the links among impossible differential, integrals and zero-correlations. Such analysis is step toward providing a classification of such attacks in terms of how effective is a certain attack to push results further for another. In this chapter we presented the concept of structure and dual structure between zero-correlation and impossible differential distiguishers. We used these concepts to prove that an impossible differential distinguisher on r rounds only exists if and only if there is a zero-correlation distinguisher on the same number of rounds. Then we found that a zero-correlation linear hull always implies an integral distinguisher on the same number of rounds. Finally we constructed a link between impossible differentials and integral cryptanalysis where r-round impossible differential distinguisher implies an integral distinguisher on the same number of rounds. This was used to push the results of integral distinguishers to be applicable on 5-round and 3-rounds structures with bijective and non-bijective components respectively. The links were mainly applied to yield a distinguisher on 7 and 8 rounds of Camellia.

Chapter 7

Epilogue and Final Remarks

Finally, for any block cipher design to withstand the public scrutiny, it should maintain the claimed security margin regardless of the mounted cryptanalytic techniques on the design or implementation over the passing of time. Which is a concept that is practically difficult to obtain as computing and cryptanalytic powers are constantly improving. In this thesis we have presented a brief on block cipher design and we introduced an analysis on selected block ciphers namely PRINTcipher, SIMON, Camellia and Cast-256. This chapter is meant to provide a brief on what we have discussed in each chapter and some final remarks around potential future work.

In Chapter 4, we have presented the we have presented an *invariant subspace attack* against iterative block ciphers which was presented and its validity was demonstrated by breaking PRINTcipher for a significant fraction of its keys. The attack finds that there are 2^{52} weak keys of the 2^{80} possible keys for PRINTcipher-48 and 2^{102} weak keys of the 2^{160} possible keys of PRINTcipher-96. The attack was linked to other classes of attacks as in multi-dimensional attack linear attack and statistical saturation attack. The attack showed that a very analytical consideration to the key classes within the design is essential especially when having a key dependant permutation layer and lightweight parameters.

In Chapter 5, we studied the security of the family of SIMON lightweight variants against differential cryptanalysis, impossible differentials and variants of linear cryptanalysis, i.e. classical- as well as linear hull attacks. For differential cryptanalysis, we have determined iterative differentials for Simon32/64, and general differentials for all variants of SIMON, that yield differential attacks on reduced versions with at least half the total rounds of the cipher in all cases for our attack. This analysis provided the grounds for our best results. An interesting observation in Section 5.3.4 is that Simon32/64 exhibits a strong differential effect. This suggests that bounding the expected differential probability (EDP) by the expected maximum characteristic probability is not well-founded in this case. Furthermore, we considered using truncated differentials to construct impossible differentials over a number of rounds, which yielded a distinguisher on reduced versions of most of the cipher variants, however it can not be to launch a practical attack as we have shown in the related sections that it yields high complexity. As for linear attacks, we mainly used a connection between linear- and differential characteristics and extended it to a connection between linear hulls and differentials. Given these connections, we used the known results on differential cryptanalysis on SIMON variants to present the best known results on SIMON using linear cryptanalysis. Furthermore, we have investigated the linear hull effect on SIMON32/64 using the correlation matrix of the average squared correlations. Utilizing this technique, we achieve a lower time and data complexity than other attack variants by having a key recovery attack on 21-round SIMON32/64 with

data complexity $2^{30.56}$ and time complexity $2^{55.6}$. A future analysis would be to investigate how far can we push the connections (within practical limits) among different cryptanalytic methods when it comes to SIMON-like ciphers as in impossible differentials and potentially zero-correlations. In addition to studying the concentration of linear trails with different hamming weights and how to efficiently use them to evaluate the linear hull effect in SIMON variant. Furthermore, understanding the security of the key scheduling of SIMON might be interesting, as based our observations when applying certain rotational difference then there is a partial repetition in the bits of the round keys. This can be used to define a possible classes of keys that can be used further to understand the minimum number of rounds that can be used to diminish such behaviour giving an intuition on the security claims or possibly finding a certain way to exploit the design. whether one can obtain related-key properties that can be exploited in a combination with rotational cryptanalysis, is an interesting open question.

In Chapter 6, In this chapter we have provided a view on the links among impossible differential, integrals and zero-correlations. Such analysis is step toward providing a classification of such attacks in terms of how effective is a certain attack to push results further for another. In this chapter we presented the concept of structure and dual structure between zero-correlation and impossible differential distiguishers. We used these concepts to prove that an impossible differential distinguisher on r rounds only exists if and only if there is a zero-correlation distinguisher on the same number of rounds. Then we found that a zero-correlation linear hull always implies an integral distinguisher on the same number of rounds. Finally we constructed a link between impossible differentials and integral cryptanalysis where r-round impossible differential distinguisher implies an integral distinguisher on the same number of rounds. This was used to push the results of integral distinguishers to be applicable on 5-round and 3-rounds structures with bijective and non-bijective components respectively. It is still an open question on how practically effective such links are for different design structures. It is interesting to seek the possibility of actually having a transparent framework of links for the most commonly used cryptanalytic attacks, for different design structures in order to visualize the collective impact of these methods on a certain structure.

Bibliography

[1] 5, 11, 13

[2] 54

[3] 69

[4] 113

[5] European Network of Excellence in Cryptology II , 2012. 13

[6] Cryptographic competitions, 2014. 14

[7] A. Bogdanov and V. Rijmen. Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers., 2014. 110, 111, 112, 114

[8] A. Bogdanov, G. Leander, K. Nyberg and M. Wang. Integral and Multidimensional Linear Distinguishers with Correlation Zero., 2012. 110, 115, 116, 117

[9] A. Bogdanov, H. Geng, M. Wang, L. Wen and B. Collard. Zero-Correlation Linear Cryptanalysis with FFT and Improved Attacks on ISO Standards Camellia and CLEFIA. 114, 123

[10] Mohamed Ahmed Abdelraheem. Estimating the Probabilities of Low-Weight Differential and Linear Approximations on PRESENT-Like Ciphers. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*, volume 7839 of *Lecture Notes in Computer Science*, pages 368–382. Springer, 2012. 92

[11] Mohamed Ahmed Abdelraheem, Martin Ågren, Peter Beelen, and Gregor Leander. On the distribution of linear biases: Three instructive examples. In *CRYPTO*, pages 50–67, 2012. 69

[12] Mohamed Ahmed Awadelkareem Mohamed Ahmed Abdelraheem. *Cryptanalysis of Some Lightweight Symmetric Ciphers*. PhD thesis, Denmark Technical University, 2013. 38

[13] Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential Cryptanalysis of Reduced-Round Simon. *IACR Cryptology ePrint Archive*, 2013:526, 2013. 94, 94, 94, 95, 108

[14] Farzaneh Abed, Eik List, Jakob Wenzel, and Stefan Lucks. Differential Cryptanalysis of round-reduced Simon and Speck. In *FSE (to appear)*, 2014. 99, 99, 104, 108, 108

[15] Martin Ågren and Thomas Johansson. Linear cryptanalysis of printcipher - trails and samples everywhere. In *INDOCRYPT*, pages 114–133, 2011. 69

[16] Javad Alizadeh, Hoda A. Alkhzaimi, Mohammad Reza Aref, Nasour Bagheri, Praveen Gauravaram, Abhishek Kumar, Martin M. Lauridsen, and Somitra Kumar Sanadhya. Cryptanalysis of SIMON Variants with Connections. In *RFIDSec'14*, volume 8651 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014. 22, 23, 72, 72, 72, 93, 93, 95, 96, 96, 108, 108

[17] Javad Alizadeh, Nasour Bagheri, Praveen Gauravaram, Abhishek Kumar, and Somitra Kumar Sanadhya. Linear Cryptanalysis of Round Reduced SIMON. *IACR Cryptology ePrint Archive*, 2013:663, 2013. 104

[18] Hoda AlKhzaimi and Martin M. Lauridsen. Cryptanalysis of the SIMON Family of Block Ciphers. *IACR Cryptology ePrint Archive*, 2013:543, 2013. 72, 72, 99

[19] Ross J. Anderson and Eli Biham. Two practical and provably secure block ciphers: BEARS and LION. In *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings*, pages 113–120, 1996. 20

[20] Kara O. Karakoc F. Atalay, A. and C. Manap. Shamata hash function algorithm specifications. *Submission to NIST*, 2008. 20

[21] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube testers and key recovery attacks on reduced-round md6 and trivium. In Orr Dunkelman, editor, *Fast Software Encryption*, volume 5665 of *Lecture Notes in Computer Science*, pages 1–22. Springer Berlin Heidelberg, 2009. 57

[22] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A Lightweight Hash. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2010. 32

[23] B. Sun, R. Li, L. Qu and C. Li. SQUARE Attack on Block Ciphers with Low Algebraic Degree, 2010. 110

[24] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004. 55

[25] Paulo S. L. M. Barreto and Vincent Rijmen. The WHIRLPOOL Hashing Function, 2001. `http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html` (accessed 2010-09-07). 29

[26] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. `http://eprint.iacr.org/`. 71, 72

[27] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Message authentication using hash functions- the hmac construction. *CryptoBytes*, 2, 1996. 18

[28] Steven M. Bellovin. Frank miller:inventor of the one-time pad. *Cryptologia*, 35(3):203–222, 2011. 23

[29] Daniel J. Bernstein. The salsa20 family of stream ciphers. In *in [38] (2008). URL: http://cr.yp.to/papers.html#salsafamily. Citations in this document: §2*, 2007. 30

[30] Daniel J. Bernstein. Chacha, a variant of salsa20, 2008. 30

[31] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak specifications, 2009. 20, 29, 29

[32] Eli Biham. On Matsui's Linear Cryptanalysis. In Santis [177], pages 341–355. 102

[33] Eli Biham, Ross J. Anderson, and Lars R. Knudsen. Serpent: A new block cipher proposal. In Serge Vaudenay, editor, *FSE*, volume 1372 of *Lecture Notes in Computer Science*, pages 222–238. Springer, 1998. 29

[34] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999. 51

[35] Eli Biham, Alex Biryukov, and Adi Shamir. Miss in the Middle Attacks on IDEA and Khufu. In *FSE*, pages 124–138, 1999. 51

[36] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990. 30, 49

[37] Eli Biham and Adi Shamir. Differential Cryptanalysis of the Full 16-Round DES. In Ernest F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 487–496. Springer, 1992. 9, 33, 44, 49, 93, 110

[38] Vincent Rijmen Ruilin Li Lei Cheng Qingju Wang Hoda Alkhzaimi Chao Li Bing Sun, Zhiqiang Liu. 112

[39] Alex Biryukov, Christophe De Cannière, and Micha el Quisquater. On multiple linear approximations. In Franklin [89], pages 1–22. 55, 98, 98, 98

[40] Alex Biryukov, Arnab Roy, and Vesselin Velichkov. Differential analysis of block ciphers SIMON and SPECK. *FSE (to appear)*, 2014. 99, 99, 108

[41] Alex Biryukov and Vesselin Velichkov. Automatic search for differential trails in arx ciphers. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 227–250. Springer International Publishing, 2014. 48

[42] Alex Biryukov and David Wagner. Slide attacks. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1999. 56

[43] Céline Blondeau and Benoît Gérard. Multiple differential cryptanalysis: Theory and practice. In Joux [114], pages 35–54. 50

[44] Céline Blondeau, Benoît Gérard, and Jean-Pierre Tillich. Accurate estimates of the data complexity and success probability for various cryptanalysis. *Designs Codes and Cryptography / Designs Codes and Cryptography An International Journal*, 59(1-3):3–34, 2011. 36 pages. 33, 50

[45] Céline Blondeau and Kaisa Nyberg. New Links between Differential and Linear Cryptanalysis. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 388–404. Springer, 2013. 93, 110

[46] Céline Blondeau and Benoît Gérard. On the data complexity of statistical attacks against block ciphers (full version). *IACR Cryptology ePrint Archive*, 2009:64, 2009. 50

[47] Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen, and Jingyuan Zhao. Key difference invariant bias in block ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8269 of *Lecture Notes in Computer Science*, pages 357–376. Springer Berlin Heidelberg, 2013. 56

[48] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full aes. In *Proceedings of the 17th International Conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'11, pages 344–371, Berlin, Heidelberg, 2011. Springer-Verlag. 42

[49] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. Spongent: A Lightweight Hash Function. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011. 92

[50] Andrey Bogdanov and Vincent Rijmen. Zero-correlation linear cryptanalysis of block ciphers. *IACR Cryptology ePrint Archive*, 2011:123, 2011. 56

[51] Andrey Bogdanov and Meiqin Wang. Zero correlation linear cryptanalysis with reduced data complexity. In Canteaut [62], pages 29–48. 56

[52] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Lars R. Knudsen, Gregor Le, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S, and Tolga Yalçın. Prince – a low-latency block cipher for pervasive computing applications full version, 2012. 32, 113

[53] Johan Borst, LarsR. Knudsen, and Vincent Rijmen. Two attacks on reduced idea. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 1997. 49

[54] Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon (full version). Cryptology ePrint Archive, Report 2014/699, 2014. http://eprint.iacr.org/. 108

[55] Canteaut A. Chevallier-Mames B. Clavier C. Fuhr T. Gouget A. Icart T. Misarsky J.-F. Naya-Plasencia M. Paillier P. Pornin T. Reinhard J.-R. Thuillet C. Bresson, E. and M. Videau. Shabal. *Submission to NIST (Round 1)*, 2008. 20

[56] Lyle D Broemeling. *An Account of Early Statistical Inference in Arab Cryptology*. The American Statistician 65, 2011. 4

[57] Stanislav Bulygin and Michael Walter. Study of the invariant coset attack on printcipher: more weak keys with practical key recovery. *IACR Cryptology ePrint Archive*, 2012:85, 2012. 69

[58] C. Blondeau, A. Bogdanov and M. Wang. On the (In)Equivalence of Impossible Differential and zero-correlation Distinguishers for Feistel- and Skipjack-type Ciphers, to appear. 111

[59] C. Blondeau and K. Nyberg. Links Between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities, to appear. 111

[60] C. Blondeau, G. Leander and K. Nyberg. Differential-Linear Cryptanalysis Revisited, to appear. 110, 110

[61] Christophe De Cannière and Bart Preneel. Trivium. In Robshaw and Billet [173], pages 244–266. 25, 32, 57

[62] Anne Canteaut, editor. *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*. Springer, 2012. 130, 140

[63] Anne Canteaut and Marion Videau. Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 518–533. Springer, 2002. 51

[64] Claude Carlet. Boolean functions for cryptography and error correcting codes. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, 134:257, 2010. 112

[65] Claude Carlet. Vectorial boolean functions for cryptography. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, (134):398–471, 2010. 27, 54

[66] Florent Chabaud and Serge Vaudenay. Links Between Differential and Linear Cryptoanalysis. In Santis [177], pages 356–365. 93, 110

[67] Joo Yeon Cho, Miia Hermelin, and Kaisa Nyberg. A New Technique for Multidimensional Linear Cryptanalysis with Applications on Reduced Round Serpent. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, volume 5461 of *Lecture Notes in Computer Science*, pages 383–398. Springer, 2008. 52

[68] Baudoin Collard and François-Xavier Standaert. A statistical saturation attack against the block cipher PRESENT. 60, 66, 67, 69, 110

[69] Baudoin Collard and François-Xavier Standaert. Multi-trail statistical saturation attacks. vol. 6123 of LNCS, 2010. 60, 69

[70] Baudoin Collard, François-Xavier Standaert, and Jean-Jacques Quisquater. Improved and Multiple Linear Cryptanalysis of Reduced Round Serpent. In Dingyi Pei, Moti Yung, Dongdai Lin, and Chuankun Wu, editors, *Information Security and Cryptology, Third SKLOIS Conference, Inscrypt 2007, Xining, China, August 31 - September 5, 2007, Revised Selected Papers*, volume 4990 of *Lecture Notes in Computer Science*, pages 51–65. Springer, 2007. 98

[71] W. Sullivan D. Malone. Guesswork is not a substitute for entropy. In *Proceedings of the Information Technology and Telecommunications Conference.*, 2005. 7

[72] Joan Daemen, Joan Daemen, Joan Daemen, Vincent Rijmen, and Vincent Rijmen. Aes proposal:rijndael, 1998. 13, 29

[73] Joan Daemen, Lars Knudsen, and Vincent Rijmen. The block cipher square, 1997. 29, 43, 57, 111

[74] Joan Daemen, Mario Lamberger, Norbert Pramstaller, Vincent Rijmen, and Frederik Vercauteren. Computational aspects of the expected differential probability of 4-round aes and aes-like ciphers. *Computing*, 85(1-2):85–104, 2009. 78

[75] Joan Daemen and Vincent Rijmen. The wide trail design strategy. In *IMA Int. Conf.*, pages 222–238, 2001. 29, 29, 58

[76] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002. 48, 54, 54, 92, 92

[77] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *IACR Cryptology ePrint Archive*, 2005:212, 2005. 45, 53, 78

[78] Ivan Damgård. A design principle for hash functions. In *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '89, pages 416–427, London, UK, UK, 1990. Springer-Verlag. 18, 20

[79] M. H. Dawson and S. E. Tavares. An expanded set of s-box design criteria based on information theory and its relation to differential-like attacks. In *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'91, pages 352–367, Berlin, Heidelberg, 1991. Springer-Verlag. 28

[80] DES. Data encryption standard. In *In FIPS PUB 46, Federal Information Processing Standards Publication*, pages 46–2, 1977. 9, 44

[81] W. Diffie and M. E. Hellman. Special feature exhaustive cryptanalysis of the nbs data encryption standard. *Computer*, 10(6):74–84, June 1977. 40

[82] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. How to Efficiently and Simultaneously Compute the Probabilities of All Iterative Characteristics. Eurocrypt 2013 Rump Session, 2013. 76

[83] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009. 56

[84] Patrik Ekdahl and Thomas Johansson. A new version of the stream cipher snow. In Kaisa Nyberg and Howard Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 47–61. Springer Berlin Heidelberg, 2003. 25

[85] ENISA. Algorithms, key sizes and parameters report. Technical report, 2013. 22, 20, 38

[86] H. Feistel. Block cipher cryptographic system, 1974. US Patent 3,798,359. 9

[87] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The skein hask function family, 2010. 30, 30

[88] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research,Wiretap Politics & Chip Design*. pub-ORA, 1998. 9

[89] Matthew K. Franklin, editor. *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*. Springer, 2004. 129

[90] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A"paradoxical" solution to the signature problem (extended abstract). In *FOCS*, pages 441–448. IEEE Computer Society, 1984. 34

[91] Zheng Gong, Svetla Nikova, and YeeWei Law. Klein: A new family of lightweight block ciphers. In Ari Juels and Christof Paar, editors, *RFID. Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2012. 32

[92] Mark Goresky and Andrew Klapper. Review of algebraic shift register sequences. *Cryptologia*, 37:175–183, 2013. 24

[93] Michaël Peeters Guido Bertoni, Joan Daemen and Gilles Van Assche. Sponge functions. In *Ecrypt Hash Workshop*, 2007. 20

[94] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON Family of Lightweight Hash Functions. In Rogaway [174], pages 222–239. 32

[95] H. Mala, M. Dakhilalian,V. Rijmen and M. Modarres-Hashemi. Improved Impossible Differential Cryptanalysis of 7-Round AES-128, 2010. 114

[96] Risto M. Hakala and Kaisa Nyberg. Linear Distinguishing Attack on Shannon. In Mu et al. [159], pages 297–305. 52

[97] Shai Halevi, Don Coppersmith, and Charanjit Jutla. Scream: A software-efficient stream cipher. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption*, volume 2365 of *Lecture Notes in Computer Science*, pages 195–209. Springer Berlin Heidelberg, 2002. 25

[98] Helena Handschuh, Helena H, and David Naccache. Shacal (- submission to nessie -), 2000. 20

[99] Carlo Harpes and booktitle =Proc. FSE 1997 volume= vol. 1267 of LNCS pages=p. 13–27 publisher=Springer year=1997 James L. Massey. title=Partitioning cryptanalysis, editor=In Eli Biham. 69

[100] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. The grain family of stream ciphers. In Robshaw and Billet [173], pages 179–190. 25, 32

[101] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980. 39, 40

[102] Luca Henzen, Willi Meier, and Raphael C. w. Phan. Sha-3 proposal blake, 2008. 30

[103] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. Multidimensional linear cryptanalysis of reduced round serpent. In Mu et al. [159], pages 203–215. 55, 55, 110

[104] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. Multidimensional extension of matsui's algorithm 2. In Orr Dunkelman, editor, *FSE*, volume 5665 of *Lecture Notes in Computer Science*, pages 209–227. Springer, 2009. 55, 55

[105] Miia Hermelin and Kaisa Nyberg. Correlation properties of the bluetooth combiner. In JooSeok Song, editor, *Information Security and Cryptology - ICISC'99*, volume 1787 of *Lecture Notes in Computer Science*, pages 17–29. Springer Berlin Heidelberg, 2000. 25

[106] Miia Hermelin and Kaisa Nyberg. Linear cryptanalysis using multiple linear approximations. Cryptology ePrint Archive, Report 2011/093, 2011. 98

[107] VietTung Hoang and Phillip Rogaway. On generalized feistel networks. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 613–630. Springer Berlin Heidelberg, 2010. 27

[108] Horapollo. *Hieroglyphica*. Translated by George Boas, 1950. 2

[109] Kota Ideguchi and Dai Watanabe. Second preimage attack on shamata-512. In Bimal K. Roy and Nicolas Sendrier, editors, *INDOCRYPT*, volume 5922 of *Lecture Notes in Computer Science*. Springer, 2009. 20

[110] Ecrypt II. Ecrypt ii yearly report on algorithms and keysizes. 2011-2012. 38

[111] Takanori Isobe and Taizo Shirai. Low-weight pseudo collision attack on shabal and preimage attack on reduced shabal-512, 2010. 20

[112] W. Stanley Jevons. *The principles of science : a treatise on logic and scientific method*, volume 2. London :Macmillan„ 1874. http://www.biodiversitylibrary.org/bibliography/31805. 21

[113] Gilles Van Assche Joan Daemen, Michael Peeters and Vincent Rijmen. block nessie proposal: NOEKEON. 2000. 65

[114] Antoine Joux, editor. *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*. Springer, 2011. 129

[115] Burton S. Kaliski Jr. and Matthew J. B. Robshaw. Linear cryptanalysis using multiple approximations. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 26–39. Springer, 1994. 55, 98

[116] Jorge Nakahara Jr., Bart Preneel, and Joos Vandewalle. Linear cryptanalysis of reduced-round versions of the SAFER block cipher family. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 244–261. Springer, 2000. 52

[117] Pascal JUNOD. *Statistical Cryptanalysis of Block Ciphers*. PhD thesis, ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE, 2005. 34

[118] Pascal Junod and Serge Vaudenay. Fox:a new family of block ciphers. In Helena Handschuh and M.Anwar Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 114–129. Springer Berlin Heidelberg, 2005. 30

[119] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima and T. Tokita. Camellia: A 128–Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis, 2000. 113, 114, 114

[120] David Kahn. *The codebreakers : the story of secret writing*. Scribner, New York, 1996. 9

[121] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC cryptography and network security. 2008. 26

[122] Auguste Kerckhoffs. La cryptographie militaire. 5

[123] Dmitry Khovratovich and Ivica Nikolić. Rotational cryptanalysis of arx. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption*, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer Berlin Heidelberg, 2010. 48

[124] Jongsung Kim, Seokhie Hong, Sangjin Lee, Junghwan Song, and Hyungjin Yang. Truncated differential attacks on 8-round crypton. In Jong-In Lim and Dong-Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 446–456. Springer Berlin Heidelberg, 2004. 50

[125] Kwangjo Kim, Tsutomu Matsumoto, and Hideki Imai. A recursive construction method of s-boxes satisfying strict avalanche criterion. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '90, pages 564–574, London, UK, UK, 1991. Springer-Verlag. 28

[126] Lars Knudsen. Deal - a 128-bit block cipher. In *NIST AES Proposal*, 1998. 44, 83, 110

[127] Lars Knudsen and Vincent Rijmen. Truncated differentials of idea. Technical report, 1997. 50

[128] Lars Knudsen and David Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer Berlin Heidelberg, 2002. 58, 110

[129] Lars R. Knudsen. *Block Ciphers - Analysis, Design and Applications*. PhD thesis, Denmark, Aarhus University, 1994. 34

[130] Lars R. Knudsen. Truncated and higher order differentials. In Preneel [167], pages 196–211. 44, 44, 50, 51, 110

[131] Lars R. Knudsen and Matthew Robshaw. *The Block Cipher Companion*. Information security and cryptography. Springer, 2011. 22, 27, 33, 34, 38, 38, 40, 44, 47, 49

[132] LarsR. Knudsen and ThomasA. Berson. Truncated differentials of safer. In Dieter Gollmann, editor, *Fast Software Encryption*, volume 1039 of *Lecture Notes in Computer Science*, pages 15–26. Springer Berlin Heidelberg, 1996. 50

[133] LarsR. Knudsen and JohnErik Mathiassen. A chosen-plaintext linear attack on des. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Bruce Schneier, editors, *Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 262–272. Springer Berlin Heidelberg, 2001. 55

[134] LarsR. Knudsen, M.J.B. Robshaw, and David Wagner. Truncated differentials and skipjack. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, volume 1666 of *Lecture Notes in Computer Science*, pages 165–180. Springer Berlin Heidelberg, 1999. 50

[135] Donald E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms,statistical tests for non-randomness*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997. 11

[136] L. Duo, C. Li and K. Feng. New Observation on Camellia. 119

[137] X. Lai. Higher order derivatives and differential cryptanalysis. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, pages 227–227, 1994. 51

[138] Xuejia Lai. *On the Design and Security of Block Ciphers*. ETH Series in Information Processing. Hartung Gorre Verlag, v.1 edition, 1992. 29

[139] Xuejia Lai and James L. Massey. Markov ciphers and differentail cryptanalysis. In Donald W. Davies, editor, *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38. Springer, 1991. 44, 46, 47, 49

[140] Axel Poschmann Lars R. Knudsen, Gregor Leander and Matthew J. B. Robshaw. Printcipher: A block cipher for IC-Printing. 61, 61, 68

[141] Martin M. Lauridsen and Hoda A. Alkhzaimi. SIMON and SPECK cryptanalysis code repository, 2013. 88

[142] Gregor Leander. On linear hulls, statistical saturation attacks, present and a cryptanalysis of puffin. vol. 6632 of LNCS:p. 303–322, 2011. 66, 67, 67, 110

[143] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A cryptanalysis of printcipher: The invariant subspace attack. In Rogaway [174], pages 206–221. 60

[144] Gaëtan Leurent. Construction of differential characteristics in arx designs application to skein. In Ran Canetti and JuanA. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 241–258. Springer Berlin Heidelberg, 2013. 48

[145] R. Lidl and H. Niederreiter. Finite fields, 1997. 24

[146] Chae Hoon Lim and Tymur Korkishko. mcrypton - a lightweight block cipher for security of low-cost rfid tags and sensors. In JooSeok Song, Taekyoung Kwon, and Moti Yung, editors, *WISA*, volume 3786 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2005. 32

[147] Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In Mitsuru Matsui, editor, *Fast Software Encryption*, volume 2355 of *Lecture Notes in Computer Science*, pages 336–350. Springer Berlin Heidelberg, 2002. 48

[148] Helger Lipmaa, Johan Wallén, and Philippe Dumas. On the additive differential probability of exclusive-or. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 317–331. Springer Berlin Heidelberg, 2004. 48

[149] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, pages 373–386, 1988. 12

[150] Stefan Lucks. The saturation attack - a bait for twofish, 2000. preprint lucks@th.informatik.uni-mannheim.de 11214 received 14 Sep 2000. 57

[151] James L. Massey. Guessing and entropy. In *In Proceedings of the 1994 IEEE International Symposium on Information Theory*, page 204, 1994. 7

[152] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993. 30, 53, 53, 110

[153] Mitsuru Matsui. Linear Cryptoanalysis Method for DES Cipher. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1994. 9, 52, 52, 94, 94, 98, 98

[154] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001. 9, 11, 11, 12, 13, 18, 19, 20, 20, 24, 41, 42

[155] Ralph C. Merkle. One way hash functions and des. In *Proceedings on Advances in Cryptology*, CRYPTO '89, pages 428–446, New York, NY, USA, 1989. Springer-Verlag New York, Inc. 19

[156] Ralph C. Merkle. A fast software one-way hash function. *Journal of Cryptology*, (3):43–58, 1990. 19

[157] Gregor Leander Mohamed Ahmed Abdelraheem and Erik Zenner. Differential cryptanalysis of round-reduced PRINTcipher: Computing roots of permutations. 6733 of LNCS, 2011. 69

[158] Hoda A. Alkhzaimi Mohammad Reza Aref Nasour Bagheri Praveen Gauravaram Mohamed Ahmed Abdelraheem, Javad Alizadeh and Martin M. Lauridsen. Improved linear cryptanalysis of reduced-round simon. Cryptology ePrint Archive, Report 2014/681, 2014. http://eprint.iacr.org/. 72, 97, 99, 101

[159] Yi Mu, Willy Susilo, and Jennifer Seberry, editors. *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings*, volume 5107 of *Lecture Notes in Computer Science*. Springer, 2008. 133, 134

[160] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 33–43, New York, NY, USA, 1989. ACM. 19

[161] Biryukov A. Nikolic, I. and D. Khovratovich. Hash family lux - algorithm specifications and supporting documentation. *Submission to NIST (Round 1)*, 2008. 20

[162] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer, 2003. 40

[163] U.S.Department of Commerce/National Institute of Standards and Technology. FIPS PUB 197, Advanced Encryption Standard (AES), 2001. 13

[164] Herodotus of Halicarnassus. *The Histories*. Perseus Digital Library, 2014. 2

[165] National Institute of Standards and Technology. *FIPS PUB 46-3*. 9

[166] Kenji Ohkuma, Hideo Shimizu, Fumihiko Sano, and Shin ichi Kawamura. Security assessment of hierocrypt and rijndael against the differential and linear cryptanalysis (extended abstract). *IACR Cryptology ePrint Archive*, 2001:70, 2001. 48

[167] Bart Preneel, editor. *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*. Springer, 1995. 136

[168] Bart Preneel. Cryptographic primitives for information authentication — state of the art. In *State of the Art in Applied Cryptography*, Lecture Notes in Computer Science, pages 49–104. Springer Berlin Heidelberg, 1998. 11

[169] Bart Preneel. The state of cryptographic hash functions. In *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, pages 158–182, London, UK, UK, 1999. Springer-Verlag. 18, 20

[170] Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, and Erik De Win. The cipher shark. In *FAST SOFTWARE ENCRYPTION, THIRD INTERNATIONAL WORKSHOP*, pages 99–112. Springer-Verlag, 1996. 29

[171] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. pages 120–126, 1978. 10, 21

[172] R.L. Rivest. *Cryptographic Competitions*, volume volume A. MIT Press/Elsevier, 1990. 17

[173] Matthew J. B. Robshaw and Olivier Billet, editors. *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*. Springer, 2008. 131, 134

[174] Phillip Rogaway, editor. *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*. Springer, 2011. 133, 136

[175] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer Berlin Heidelberg, 2004. 19

[176] S. Wu, M. Wang. Automatic Search of Truncated Impossible Differentials for Word-Oriented Block Ciphers, 2012. 119

[177] Alfredo De Santis, editor. *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*. Springer, 1995. 129, 131

[178] AliAydın Selçuk. On probability of success in linear and differential cryptanalysis. *Journal of Cryptology*, 21(1):131–147, 2008. 33, 49, 55

[179] Claude E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423,623–656, 1948. 1, 6

[180] Claude E. Shannon. Communication Theory of Secrecy Systems. *Bell Systems Technical Journal*, 28:656–715, 1949. 6, 26

[181] Akihiro Shimizu and Shoji Miyaguchi. Fast data encipherment algorithm feal. In David Chaum and Wyn L. Price, editors, *EUROCRYPT*, volume 304 of *Lecture Notes in Computer Science*, pages 267–278. Springer, 1987. 30, 52

[182] Simon Singh. *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, New York, NY, USA, 1st edition, 1999. 9

[183] S.K. Langford and M.E. Hellman. Differential-Linear Cryptanalysis, 1994. 110

[184] Arthur Sorkin. LUCIFER, a cryptographic algorithm. *j-CRYPTOLOGIA*, 8(1):22–42, jan 1984. 9

[185] Othmar Staffelbach and Willi Meier. Cryptographic significance of the carry for ciphers based on integer addition. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '90, pages 601–614, London, UK, UK, 1991. Springer-Verlag. 48

[186] Paul Stankovsk. *Cryptanalysis of Selected Stream Ciphers*. PhD thesis, Lund University, 2013. 34

[187] Douglas R. Stinson. Some observations on the theory of cryptographic hash functions. *Des. Codes Cryptography*, 38(2):259–277, 2006. 19

[188] T. Jacobsen and L.R. Knudsen. The Interpolation Attack on Block Ciphers, 1997. 110

[189] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, pages 230–265, 1936. 9

[190] Henk C. A. van Tilborg and Sushil Jajodia, editors. *Encyclopedia of Cryptography and Security, 2nd Ed.* Springer, 2011. 9, 18

[191] Serge Vaudenay. On the lai-massey scheme. In Kwok-Yan Lam, Eiji Okamoto, and Chaoping Xing, editors, *Advances in Cryptology - ASIACRYPT'99*, volume 1716 of *Lecture Notes in Computer Science*, pages 8–19. Springer Berlin Heidelberg, 1999. 30

[192] Serge Vaudenay. *A Classical Introduction to Cryptography : Applications for Communications Security*. Computer Science Springer-11645; ZDB-2-SCS. Springer Science+Business Media, Inc. Springer e-books, Boston, MA, 2006. 2, 3

[193] Vesselin Velichkov, Nicky Mouha, Christophe De Cannière, and Bart Preneel. UNAF: A special set of additive differences with application to the differential analysis of ARX. In Canteaut [62], pages 287–305. 44

[194] G.S. Vernam. Secret signaling system, 1919. US Patent 1,310,719. 23

[195] W. Wu, W. Zhang and D. Feng. Impossible Differential Cryptanalysis of Round-Reduced ARIA and Camellia, 2007. 114

[196] David Wagner. The boomerang attack. In Lars Knudsen, editor, *Fast Software Encryption*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer Berlin Heidelberg, 1999. 44

[197] Ning Wang, Xiaoyun Wang, Keting Jia, and Jingyuan Zhao. Improved Differential Attacks on Reduced SIMON Versions, 2014. 109

[198] Qingju Wang, Zhiqiang Liu, Kerem Varici, Yu Sasaki, Vincent Rijmen, and Yosuke Todo. Cryptanalysis of Reduced-Round SIMON32 and SIMON48, 2014. 109

[199] R.P. Weinmann. Axr-crypto made from modular additions, xors and word rotations. Dagstuhl Seminar 09031, 2009),. 30

[200] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography, 1976. 10, 19, 21

[201] WIKIPEDIA. Global surveillance disclosures (2013–present), 2014. 12

[202] F. W.Kasiski. 4

[203] Shuang Wu, Dengguo Feng, and Wenling Wu. Cryptanalysis of the hash function lux-256, 2008. 20

[204] Muhammad Reza Z'Aba, Håvard Raddum, Matt Henricksen, and Ed Dawson. Fast software encryption. chapter Bit-Pattern Based Integral Attack, pages 363–381. Springer-Verlag, Berlin, Heidelberg, 2008. 58