

École doctorale n°432 : Sciences des Métiers de l'ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers
Spécialité Informatique

présentée et soutenue publiquement par

Arnaud BANNIER

le 1 octobre 2017

Combinatorial Analysis of Block Ciphers With Trapdoors

Directeur de thèse : **Éric FILIOL**

Jury

M. Kenneth Paterson	Professeur	Royal Holloway, University of London	Rapporteur
M. Massimiliano Sala	Professeur	University of Trento	Rapporteur
Mme. Anne Canteaut	Professeur	Inria Paris	Examineur
M. Jean-Marc Steyaert	Professeur	École Polytechnique	Examineur
M. Alexei Zhukov	Professeur	Bauman Moscow State Technical University	Examineur

THÈSE

Trapdoors are a two-face key concept in modern cryptography. They are primarily related to the concept of *trapdoor function* used in asymmetric cryptography. A trapdoor function is a one-to-one mapping that is easy to compute, but for which its inverse function is difficult to compute without special information, called the *trapdoor*. Such a trapdoor is essential in public key encryption algorithms and digital signatures as it ensures that only the person who knows the secret information can decrypt or sign messages. In this case, the trapdoor mechanism is always public and fully detailed.

The second concept of trapdoor considers *by-design mathematical backdoors* and is a key issue in symmetric cryptography. In this case, the aim is to insert hidden mathematical weaknesses which enable one who knows them to break the cipher. The existence of a backdoor is hence a strongly undesirable property. While the term of *trapdoor* has been already used in the very few literature covering this issue, we suggest however to use the term of *backdoor* to describe hidden mathematical weaknesses in symmetric cryptography in order to avoid ambiguity. This thesis is focused on backdoors in block ciphers or, more specifically, on Substitution-Permutation Networks (SPN).

Inserting a backdoor in an encryption algorithm gives an effective cryptanalysis of the cipher to the designer. However, like any other cipher, this backdoor cipher may be vulnerable to classical cryptanalysis. If a classical attack can easily break it, the asymmetry between the designer and those who do not know the backdoor disappears; thus, the backdoor cipher is just a weak cipher.

Differential [13] and linear [74] cryptanalysis are considered as the most important attacks against block ciphers [64]. As mentioned in [41], any new cipher should at least be accompanied by a detailed analysis of its strength against these two attacks. The practical resistance of a block cipher against differential and linear cryptanalysis is assessed by the differential probability or linear potential of an optimal differential or linear trail respectively. When these values are low enough, the cipher is said to be *practically secure*. To prevent differential and linear cryptanalysis, the cipher designer chooses primitives which provides high resistance against both these attacks. Nevertheless, the mathematical structure of the backdoor strongly reduces the choice of these primitives and the usual strategies may no longer be useful.

In [76], Matsui presented an algorithm that computes an optimal trail in a Feistel cipher. In other words, the execution of this algorithm can prove the cipher practical security. The algorithm complexity remaining too high for the cipher FEAL, two

successive improvements have been proposed in [87] then [3]. Although an adaptation of Matsui’s algorithm is straightforward for SPN, the block size of modern ciphers makes it computationally infeasible. The first contribution of this thesis is an improvement of this algorithm for SPN [8] and has received a best paper award. We introduce several optimizations paying special attention to SPN whose diffusion layer is a bit permutation. Because bit permutations do not provide high diffusion, the cipher security is hard to establish without a close analysis. Such mappings are generally chosen for efficiency purposes. On the contrary, diffusion layers providing high diffusion yield sufficient bounds to prove the cipher security but can be more computationally expensive. Therefore, such an algorithm is more useful for SPN using bit permutations.

Spending months computing the practical security of a known cipher is not a problem. However, the cipher designer has to repeat this search several times in order to optimize the choice of the cipher components or the number of rounds. Our algorithm meets this need since its execution time on the full PRESENT [17] is below one second on a laptop computer.

Now we have a tool to evaluate the security of an SPN with regard to differential and linear cryptanalysis, we turn our attention to backdoor ciphers. The family of backdoor ciphers covered by this thesis is a generalization of the imprimitive ciphers introduced by Paterson in [88]. For such ciphers, the round function preserves a partition of the message space no matter the round keys used, and hence the same applies to the full cipher. This partition forms the backdoor and yields a powerful cryptanalysis with a suitably chosen key schedule. Even if the mathematical theory of the backdoor is given, no general algorithm details how to construct the primitives of the cipher. Moreover, the author wondered what are the possible partitions for this backdoor. Caranti and al. [31] answered this question by proving that only linear partitions can be considered. Along a similar line, Harpes considered in his thesis [50] backdoor ciphers mapping a partition of the plaintexts to a partition of the ciphertexts. As these partitions are not necessarily equal, this family generalizes Paterson’s one. These ciphers are called *partition-based backdoor ciphers*.

The main contribution of this thesis is an extension of Paterson and Harpes’ works for SPN. In our study, we consider an SPN mapping a partition of the plaintexts to one of the ciphertexts, no matter what the round keys are. In other words, we assume that this property holds independently of the key schedule and the cipher key. Firstly, we prove that the round function of such an SPN must at least map a linear partition to another linear one. This result generalizes [31] since we consider the full cipher and not only the round function. It should be stressed that the apparent combinatorial aspect of our assumption is reduced to an algebraic one. Since it is easy to show that any linear transformation maps every linear partition to another one, the diffusion layer can be bypassed so that the substitution layer necessarily maps a linear partition to another one.

The substitution layer consists of several S-boxes evaluated in parallel. The natural problem that arises is to determine the properties the previous result implies on each S-box. This refinement is far more complicated than the previous one because it requires a deep analysis of the structure of the linear partitions with

respect to the substitution layer. We eventually managed to prove that at least one of the S-boxes must map a linear partition to another one. To summarize, the study of the full cipher is reduced to that of the S-boxes.

In the light of this result, we are now interested in designing an S-box mapping a linear partition to another one with the best resistance against differential and linear cryptanalysis. We show how the Krasner-Kaloujnine embedding theorem yields an internal decomposition of such S-boxes. Using this decomposition, we manage to derive bounds for the differential and linear properties of backdoor S-boxes and we present an algorithm to design a-boxes which almost reach these bounds. Combining our reduction result with these bounds, we derive a criterion which can prove that an SPN does not belong to this family of backdoor ciphers. All these results were published in [9] and [12].

The last part puts into practice our theoretical treatment of partition-based backdoor ciphers. First, we present a toy backdoor SPN and break it with a key-schedule dependent attack suggested by Paterson but not detailed. Finally we present BEA-1 (standing for Backdoored Encryption Algorithm), a real-size backdoor cipher inspired by the current standard of symmetric encryption, namely the AES. Our cipher encrypts 80-bit data blocks using using 120-bit cipher key and is designed to resist linear and differential cryptanalysis. Conversely, the backdoor enables recovery of the full 120-bit cipher key in just a few seconds on a laptop computer using only 2^{16} chosen plaintext blocks. This cipher was presented in [11] as a challenge. Its cryptanalysis was then outlined in [11] and detailed in [12]. It should be mentioned that my teaching activity led me to consider Venn Diagrams. As a result, we published a new infinite family of Venn diagrams in [7].

This thesis is organized as follows. Firstly, Chapter 1 recalls the definition of substitution-permutation networks and the differential and linear cryptanalysis. An algorithm performing a security analysis with respect to these attack is then described in Chapter 2. Backdoor ciphers are then the focus of this thesis from Chapter 3 to the end. In this chapter, we investigated the structure of partition-based backdoor ciphers and prove that their study can be reduced to their S-boxes. Next, Chapter 4 is devoted to the analysis of such S-boxes and ends with a toy backdoor cipher illustrating the results of these two chapters. Finally, Chapter 5 concludes our work by introducing BEA-1, a real-size backdoor cipher, and explains how its backdoor can be exploited to break it effectively.

Contents

1	Substitution-Permutation Networks	1
1.1	Preliminaries	1
1.2	Substitution-Permutation Networks	3
1.3	Differential Cryptanalysis	9
1.3.1	General Idea of the Attack	10
1.3.2	Differential Trails	15
1.4	Linear cryptanalysis	20
1.4.1	General Idea of the Attack	20
1.4.2	Linear Approximations and Linear Trails	24
1.5	Security Evaluation of SPN and Strong Primitives	28
1.5.1	Perfect S-Boxes	30
1.5.1.a	Almost Perfect Nonlinear Functions	31
1.5.1.b	Almost Bent Functions	32
1.5.1.c	Known AB and APN Permutations	33
1.5.2	Branch Number of the Diffusion Layer	34
2	Security Evaluation of SPN	37
2.1	Search for an Optimal trail	38
2.1.1	General Principle	40
2.1.2	Proof of the Algorithm	41
2.2	A Detailed Example	43
2.2.1	Search Algorithm for the First Round	46
2.2.2	Search Algorithm for the Round Function	49
2.2.3	Search Algorithm for the Last Round	52
2.3	Optimizations	53
2.3.1	Construction of the First Output Pattern	53
2.3.2	The Round Function	55
2.3.3	Active S-Boxes in the Next Round	56
2.3.4	Test on the Bound	58
2.3.5	Automatic Management of the Estimation	60
2.4	Results	60
3	Partition-Based Backdoor Ciphers	63
3.1	Partition-Based Backdoor Ciphers	65
3.1.1	Imprimitive Group Actions	65
3.1.2	Imprimitive Backdoor ciphers	68

CONTENTS

3.1.3	Exploiting the backdoor	69
3.1.4	Generalizations	71
3.1.5	Links With Other Attacks	72
3.2	Substitution-Permutation Networks and Partitions	73
3.2.1	Linear Partitions	74
3.2.2	The Key Addition and Diffusion Layer	77
3.2.3	From the Encryption Function to the Substitution Layer	81
3.3	Structure of the Substitution Layer	84
3.3.1	Truncating the substitution layer	85
3.3.2	Structure of the Subspaces V and W	88
3.3.3	Linked and Independent S-Boxes	92
3.3.4	The Forbidden Case	95
3.3.5	Reduction to one S-Box	100
3.4	Conclusion	103
4	Analysis of a Backdoor S-Box	105
4.1	Structure of a Backdoor S-Box	105
4.1.1	Wreath Product	109
4.1.2	Krasner-Kaloujnine Embedding Theorem	113
4.1.3	Application of the Embedding Theorem	115
4.2	Differential and linear analyses	116
4.2.1	Correlation Matrices and Linear Potentials	119
4.2.2	Differential Probabilities	121
4.2.3	Designing a Backdoor S-Box	127
4.3	A Toy Partition-Based Backdoor Cipher	129
4.3.1	Specification of TBC	129
4.3.2	Differential and Linear Cryptanalysis	131
4.3.3	The Backdoor	132
4.3.3.a	Basic and Multiple Partitions Attacks	134
4.3.3.b	Key Schedule Dependent Attack	135
4.3.4	The Flaws of This Cipher	136
4.4	Preventing Partition-Based Backdoors	137
5	Backdoored Encryption Algorithm 1	141
5.1	Presentation of BEA-1	141
5.1.1	Specification of the Encryption Process	141
5.1.2	Differential and Linear Cryptanalysis	144
5.2	Design of the Backdoor	145
5.2.1	The Linear Partitions Throughout the Encryption	145
5.2.2	The Substitution Layer	148
5.2.3	The Diffusion Layer	149
5.3	Main Idea of the Cryptanalysis	152
5.3.1	A Detailed Example	152
5.3.2	Formalization of the Attack	156

5.4	Cryptanalysis of BEA-1 Using the Backdoor	157
5.4.1	Part 1: Finding the Right Output Coset	158
5.4.2	Part 2: Obtaining Candidates for the Last Round Key	160
5.4.3	Part 3: Finding the Last Round Key	161
5.4.4	Part 4: Obtaining Candidates for the Remaining Bits. . . .	164
5.4.5	Part 5: Deducing the Cipher Key	165
5.5	Conclusion	166

CONTENTS

Substitution-Permutation Networks

This chapter aims at offering an introduction of substitution-permutation networks and their cryptanalysis. We start with some notations, terminologies and basic results. Then, we present in Section 1.2 the main definitions concerning blocks ciphers and we focus particularly on Substitution-Permutation Networks (SPN). After given an example of such ciphers, we explore the two mains attacks of SPN, namely differential and linear cryptanalysis, in Sections 1.3 and 1.4 respectively. Finally, we conclude this chapter with a discussion on the security of SPN against these attacks.

1.1. Preliminaries

Let us begin with some notations and conventions. The cardinality of a finite set E is denoted by $\#E$. The *complement* of a subset F of E consists of all elements in E not in F and is denoted by F^c . Let f be a mapping from E to F and g be a mapping from F to G . The *composition* of g on f is the mapping $g \circ f$ from E to G which maps x to $g(f(x))$. A *permutation* of E is any bijective mapping from E to E . If σ and τ are permutations of E , then we often write $\sigma\tau$ instead of $\sigma \circ \tau$, namely we omit the small circle when composing permutations to fit the abstract formalism of permutation groups.

Let n and m be two positive integers. The Galois field of order two is denoted by \mathbb{F}_2 . Every the vector space considered in this thesis will be over the finite field \mathbb{F}_2 . The set of all n -bit sequences is identified with the n -dimensional vector space \mathbb{F}_2^n . In this space, the addition denoted by $+$ can be seen as a bitwise exclusive or (Xor). The zero vector $(0, \dots, 0)$ is simply denoted by 0_n . The concatenation of two binary vectors x and y is denoted $(x \parallel y)$. Similarly, if $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ and $g : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^m$ are two mappings, then $(f \parallel g)$ denote the mapping from \mathbb{F}_2^{n+m} to \mathbb{F}_2^{n+m} which maps $(x \parallel y)$ to $(f(x) \parallel g(y))$. Moreover, we should mentioned that the space \mathbb{F}_2^{nm} will be often identified with $(\mathbb{F}_2^n)^m$ by gathering the bits in m bundles of n components.

Now, let us recall basic properties on linear algebra. For a complete introduction, the reader should refer to the work of Lang [68]. The *dot product* of two vectors x

and y in \mathbb{F}_2^n , denoted by $\langle x, y \rangle$, is defined by the rule

$$\langle x, y \rangle = x \times y^\top = y \times x^\top = \sum_{i=0}^{n-1} x_i y_i,$$

where x^\top denotes the transpose of x . It is well-known that the dot product is a bilinear form, namely for all x, y, z in \mathbb{F}_2^n and all λ in \mathbb{F}_2 , it holds that

$$\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle, \quad \langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle, \quad \langle \lambda x, y \rangle = \langle x, \lambda y \rangle = \lambda \langle x, y \rangle.$$

In addition, the dot product satisfies the followings two properties:

- (symmetric) for every x and y in \mathbb{F}_2^n , $\langle x, y \rangle = \langle y, x \rangle$.
- (non-degenerate) for any x in \mathbb{F}_2^n , if $\langle x, y \rangle = 0$ for all y in \mathbb{F}_2^n , then $x = 0$.

If E is a subset of \mathbb{F}_2^n , we denote by E^\perp the set of all elements x in \mathbb{F}_2^n which are perpendicular to all elements of E with respect to the dot product, that is to say,

$$E^\perp = \{x \in \mathbb{F}_2^n \mid \forall y \in E, \langle x, y \rangle = 0\}.$$

It is easily seen from the properties of the dot product that E^\perp is a subspace of \mathbb{F}_2^n . Therefore, E^\perp is called the *orthogonal space* of E . Moreover, $E^\perp = \text{span}(E)^\perp$.

Definition 1.1 (Transpose). Let $L : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be a linear mapping. There exists a unique mapping $L^\top : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$, called the *transpose* of L , such that

$$\langle L(x), y \rangle = \langle x, L^\top(y) \rangle$$

for all x in \mathbb{F}_2^n and all y in \mathbb{F}_2^m . Furthermore, if A is the $n \times m$ matrix in \mathbb{F}_2 satisfying $L(x) = x \times A$, then L^\top is given by $L^\top(y) = y \times A^\top$.

Proof. First, observe that for all elements x and y of \mathbb{F}_2^n and \mathbb{F}_2^m , we have

$$\langle L(x), y \rangle = L(x) \times y^\top = x \times A \times y^\top = x \times (y \times A^\top)^\top = \langle x, L^\top(y) \rangle.$$

It remains to prove the uniqueness of this mapping. Suppose that L' and L'' are two mappings from \mathbb{F}_2^m to \mathbb{F}_2^n satisfying the required property. Let x and y be elements of \mathbb{F}_2^n and \mathbb{F}_2^m . By assumption, it follows that $\langle x, L'(y) \rangle = \langle x, L''(y) \rangle$, or equivalently, $\langle x, L'(y) + L''(y) \rangle = 0$ because of the bilinearity of the dot product. Therefore, $L'(y) + L''(y) = 0$ as this equality holds for all x in \mathbb{F}_2^n and as the dot product is non-degenerate. Consequently, $L'(y) = L''(y)$ as desired. ■

Proposition 1.2. Let $L : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be a linear mapping. The kernel of L^\top is the orthogonal space of the image of L , that is $\text{Ker}(L^\top) = \text{Im}(L)^\perp$.

Proof. By definition, the kernel of L^\top is the set of elements of \mathbb{F}_2^m mapped to 0_n by L^\top . If y lies in \mathbb{F}_2^m , then $L^\top(y) = 0$ if and only if $\langle x, L^\top(y) \rangle = 0$ for every x in \mathbb{F}_2^n since the dot product is non-degenerate. Next,

$$\begin{aligned} \text{Ker}(L^\top) &= \{y \in \mathbb{F}_2^m \mid \forall x \in \mathbb{F}_2^n, \langle x, L^\top(y) \rangle = 0\} = \{y \in \mathbb{F}_2^m \mid \forall x \in \mathbb{F}_2^n, \langle L(x), y \rangle = 0\} \\ &= \{y \in \mathbb{F}_2^m \mid \forall x \in \text{Im}(L), \langle x, y \rangle = 0\} = \text{Im}(L)^\perp. \end{aligned}$$

The result is proven. ■

1.2. Substitution-Permutation Networks

Cryptology is the science of secrets. It aims at enabling two people, called Alice and Bob, to communicate over an insecure channel. A channel can be any medium of communication, for instance a telephone line or a computer network. It is said insecure whenever a third party can intercept or modify the sent messages. Cryptology is divided into two complementary parts. On the one hand, cryptography gathers the methods to protect the information. Naturally, this includes *confidentiality* which ensures that an adversary intercepting the messages cannot gain information about the content of the communication. However, cryptography is equally interested in *integrity* (ensuring that the message you received was the message sent) and *authenticity* (ensuring the source of the messages). On the other hand, cryptanalysis intends to break the security provided by cryptography.

Confidentiality is provided using an encryption algorithm. In symmetric key cryptography, Alice and Bob must share a secret key before they can communicate over an insecure channel. Assume that Alice wants to communicate with Bob. The message she wants to send is called the *plaintext*. Then, using the secret key, Alice encrypts the message. The resulting data is called the *ciphertext* and should conceal any information about the plaintext. Next, Alice sends the ciphertext to Bob. The latter can decrypt this ciphertext using the same secret key and hence recover the original message.

Symmetric key encryption algorithms are divided into block ciphers and stream ciphers. In this thesis, we consider only block ciphers, for an overview of stream ciphers the reader can consult [77] for instance. A block cipher is an encryption algorithm processing fixed length blocks of data using a secret key, called the *cipher key* [39]. We introduce now its formal definition.

Definition 1.3 (Block Cipher). Let n and κ be positive integers. A *block cipher* is a mapping $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ that takes a κ -bit cipher key K and an n -bit plaintext p and returns the n -bit corresponding ciphertext $c = E(K, p)$. Furthermore, for each cipher key K in \mathbb{F}_2^κ , the mapping $E_K : p \mapsto E(K, p)$ is required to be a permutation of \mathbb{F}_2^n .

The integer n is the *block size* of the cipher and κ its *key length*. The mapping E_K is the *encryption function* associated with the *cipher key* K . Its inverse mapping is the *decryption function* and is denoted by D_K . It is worth observing that each encryption function E_K must be bijective to enable decryption. Indeed, assuming that E_K is not injective, there exist two different plaintexts p and p' such that $c = E_K(p) = E_K(p')$. Then, the receiver who wants to decrypt c cannot decide whether the corresponding plaintext is p or p' . Moreover, the mapping E_K is either bijective or not injective because \mathbb{F}_2^n is a finite set, justifying the definition.

A block cipher on its own cannot be used to encrypt any message. In fact, it processes only fixed length messages by definition. Generally the block size n ranges between 64 and 128. In order to encrypt long messages whose lengths are not

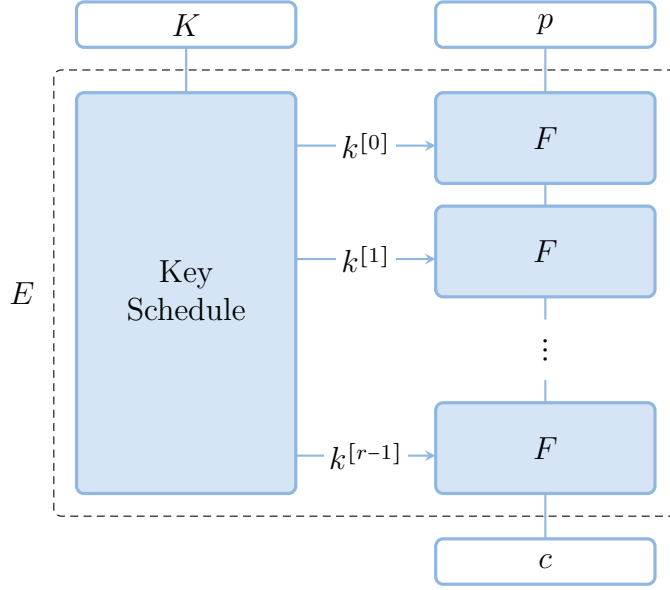


Figure 1.1: Representation of an iterative block cipher (see Definition 1.4).

necessarily a multiple of the block size, we must specify how this primitive is used. These specifications are called *modes of operation* of a block cipher. Informally, a mode of operation splits the message into several n -bit blocks. Then, these blocks are linked together and encrypted using the block cipher. The first standardization of modes of operation was in [84], originally for the Data Encryption Standard (DES [83]). In this publication, four modes were presented called ECB, CBC, CFB and OFB. After the publication of the next standard of block ciphers, namely the Advanced Encryption Standard (AES [85]), another standardization was published by the National Institute of Standards and Technology in [86].

Block ciphers come in all shapes and sizes. Let us now introduce an important class which includes almost all modern block ciphers.

Definition 1.4 (Iterative Block Cipher). A block cipher $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^n$ is said to be an r -round iterative block cipher if it can be decomposed as follows.

- An algorithm called the *key schedule* processes the cipher key K in \mathbb{F}_2^κ and produces r round keys $k^{[0]}, \dots, k^{[r-1]}$ in \mathbb{F}_2^l .
- There exists a mapping $F : \mathbb{F}_2^l \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ called the *round function*, such that any encryption function E_K can be written as

$$E_K = F_{k^{[r-1]}} \circ \dots \circ F_{k^{[0]}} .$$

Remark 1.5. The integer l is called the *round key length*. For significant proportion of iterative block ciphers, the round key length is equal to the block size. The mapping F_k from \mathbb{F}_2^n to \mathbb{F}_2^n which maps a block x to $F(k, x)$ is called the *round function associated with the round key k* . Naturally, each mapping F_k must be a permutation of \mathbb{F}_2^n . A diagrammatic representation of an iterative block cipher is given in Figure 1.1.

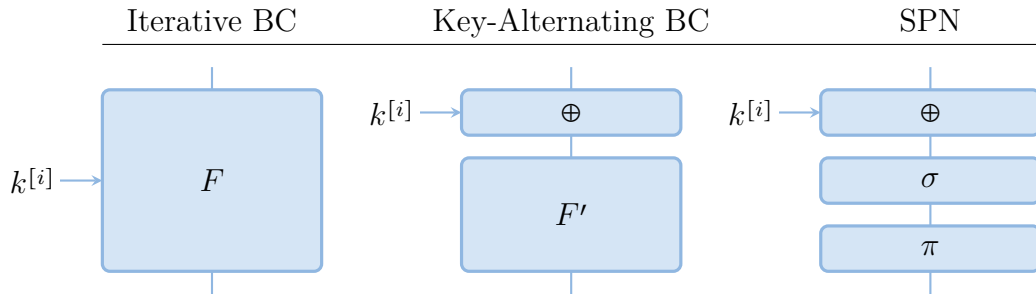


Figure 1.2: The round functions of iterative block ciphers, key-alternating block ciphers and SPNs (Definitions 1.4, 1.6 and 1.10).

Definition 1.6 (Key-Alternating Cipher). A key-alternating block cipher is an iterative block cipher such that

- the round key length is equal to the block size n ,
- there exists an unkeyed permutation F' of \mathbb{F}_2^n such that each round function F_k maps a block x to $F'(x + k)$.

The round function of a key-alternating cipher is illustrated in Figure 1.2. As said in [40], a key-alternating cipher consists of an alternating sequence of unkeyed rounds and simple bitwise key additions. Indeed, when considering two elements of \mathbb{F}_2^n , their addition corresponds with the addition in the vector space \mathbb{F}_2^n , namely a bitwise addition in \mathbb{F}_2 . Moreover, the addition in the finite field \mathbb{F}_2 is just an exclusive or (XOR). Consequently, this operation is often denoted by the symbol \oplus in cryptography. However, since we will consider direct sums of vector spaces, we denote the sum in \mathbb{F}_2^n by the symbol $+$ to avoid confusion.

Remark 1.7. It should be mentioned that in the encryption process of a key-alternating block cipher, a round key is added after the last round. According to Kerckhoffs' Principle [59], the attacker knows every detail of the encryption, except of course the secret key. Therefore, if the encryption ends with the unkeyed permutation F' , the attacker can undo this step. To save useless processing, the last step must be unknown to the cryptanalyst. Summarizing, the key schedule of an r -round key-alternating cipher derives $r + 1$ round keys $k^{[0]}, \dots, k^{[r]}$ in \mathbb{F}_2^n from a cipher key K in \mathbb{F}_2^κ and the last key $k^{[r]}$ is added at the end of the encryption.

Denote by α_k the permutation of \mathbb{F}_2^n which maps x to $x + k$, that is to say α_k represents the addition of the round key k . Assume that $k^{[0]}, \dots, k^{[r]}$ are the round keys derived from a cipher key K . Then, the encryption function is given by

$$E_k = \alpha_{k^{[r]}} \circ \underbrace{F' \circ \alpha_{k^{[r-1]}} \circ \dots \circ F' \circ \alpha_{k^{[0]}}}_{F_{k^{[0]}}} .$$

Closely related to key-alternating block ciphers is the concept of long-key cipher introduced by Daemen and Rijmen in [40].

Definition 1.8 (Long-Key Cipher). A long-key cipher is a key-alternating cipher with a trivial key schedule. The cipher key consists of the concatenation of the $r + 1$ round keys, and hence is $n(r + 1)$ -bit long.

Certainly, long-key ciphers are not used in practice as they require too long cipher keys. They are used only to study key-alternating ciphers. By the long-key cipher associated with a key-alternating cipher, we mean the cipher obtained when we ignore its key schedule and consider independent round keys.

In his seminal paper in 1949 [91], Shannon introduced the main design principles used nowadays in block ciphers, namely *confusion* and *diffusion*. To quote Shannon, “The method of confusion is to make the relation between the simple statistics of the ciphertext and the simple description of the key a very complex and involved one” and “in the method of diffusion the statistical structure of the plaintext which leads to its redundancy is dissipated into long range statistics in the ciphertext”. These concepts can be interpreted in several ways, a nice adaptation is due to Massey [73]:

Confusion: The ciphertext statistics should depend on the plaintext statistics in a manner too complicated to be exploited by the cryptanalyst.

Diffusion: Each digit of the plaintext and each digit of the secret key should influence many digits of the ciphertext.

A class of key-alternating block ciphers directly inspired by Shannon’s work is the *Substitution Permutation Networks* (shorten as SPN). The round function of an SPN consists of three distinct stages: a *key addition*, a *substitution layer* and a *permutation* or *diffusion layer*. The key addition includes unknown material to the cryptanalyst, the substitution and diffusion layers provide respectively confusion and diffusion. One of the primitives of any SPN is called a *Substitution-box* or simply an *S-box*.

Definition 1.9 (S-Box). An n -bit S-box is a mapping from \mathbb{F}_2^n to \mathbb{F}_2^n . In this thesis, we require an S-box to be a permutation of \mathbb{F}_2^n .

In the substitution layer, the nm -bit data block is seen as m bundles of n bits. Then, m S-boxes are evaluated in parallel on each bundle of the block. For this reason, the substitution layer is said to be a bricklayer function [39]. On the other hand, the diffusion layer consists of the evaluation of some linear mappings (generally one) but processes the data block as a whole since it is intended to provide diffusion.

Definition 1.10 (SPN). Let m and n be positive integers and let S_0, \dots, S_{m-1} be n -bit S-boxes.

- The *substitution layer* is denoted by σ and maps $(x_i)_{0 \leq i < m}$ to $(S_i(x_i))_{0 \leq i < m}$.
- The *diffusion layer* is a linear permutation denoted by $\pi : \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$.

A *Substitution-Permutation Network* is a key-alternating block cipher such that the unkeyed round function F' is equal to $\pi \circ \sigma$.

Denoting by $k^{[0]}, \dots, k^{[r]}$ the round keys derived from a cipher key K . The

encryption function of an SPN is hence defined to be

$$E_K = \alpha_{k[r]} \circ \underbrace{\pi \circ \sigma \circ \alpha_{k[r-1]} \circ \cdots \circ \pi \circ \sigma \circ \alpha_{k[0]}}_{F_{k[r-1]}} \circ \underbrace{\pi \circ \sigma \circ \alpha_{k[0]}}_{F_{k[0]}} .$$

A comparison between the round function of an iterative cipher, a key-alternating cipher and an SPN is illustrated in Figure 1.2. It is worthwhile to note that the substitution layer is the only step which is not linear or affine. In order to make the cipher secure, the S-boxes must be highly nonlinear. The exact meaning of this statement will be detailed in Section 1.5.1.

Remark 1.11. We should mention that the last round of an SPN may not include the diffusion layer, see for instance the AES. Indeed, for all k and x in \mathbb{F}_2^{nm} we have

$$\alpha_k \circ \pi(x) = k + \pi(x) = \pi(\pi^{-1}(k)) + \pi(x) = \pi(\pi^{-1}(k) + x) = \pi \circ \alpha_{\pi^{-1}(k)}(x) .$$

Thus, using an equivalent last round key, the encryption process ends with a permutation known by the cryptanalyst. As a consequence, the diffusion layer in the last round does not offer any additional security and is often removed for efficiency purposes.

Before concluding this section with an example of a SPN, we introduce a class of diffusion layers particularly used in lightweight block ciphers.

Definition 1.12 (bit permutation). A linear mapping $\pi : \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$ is said to be a *bit permutation* if there exists a permutation ϕ of $\llbracket 0, nm \rrbracket$ such that

$$\pi(x_0, \dots, x_{nm-1}) = (x_{\phi^{-1}(0)}, \dots, x_{\phi^{-1}(nm-1)}) .$$

Remark 1.13. Despite appearances, using ϕ^{-1} on the indices is natural. The bit with index i is mapped to the index $\phi(i)$. Equivalently, the bit mapped to the index i was originally at the index $\phi^{-1}(i)$.

Example 1.14. Let us introduce a 5-round SPN, called TOYCIPHER, which encrypts a 16-bit block using a 16-bit cipher key. Thus, for this cipher $r = 5$ and $nm = \kappa = 16$. The substitution layer σ evaluates in parallel one 4-bit S-box denoted by S . This S-box is given in hexadecimal notation at the bottom of Figure 1.3. For instance, S maps 2 to D or equivalently $S(0010) = 1101$. Similarly,

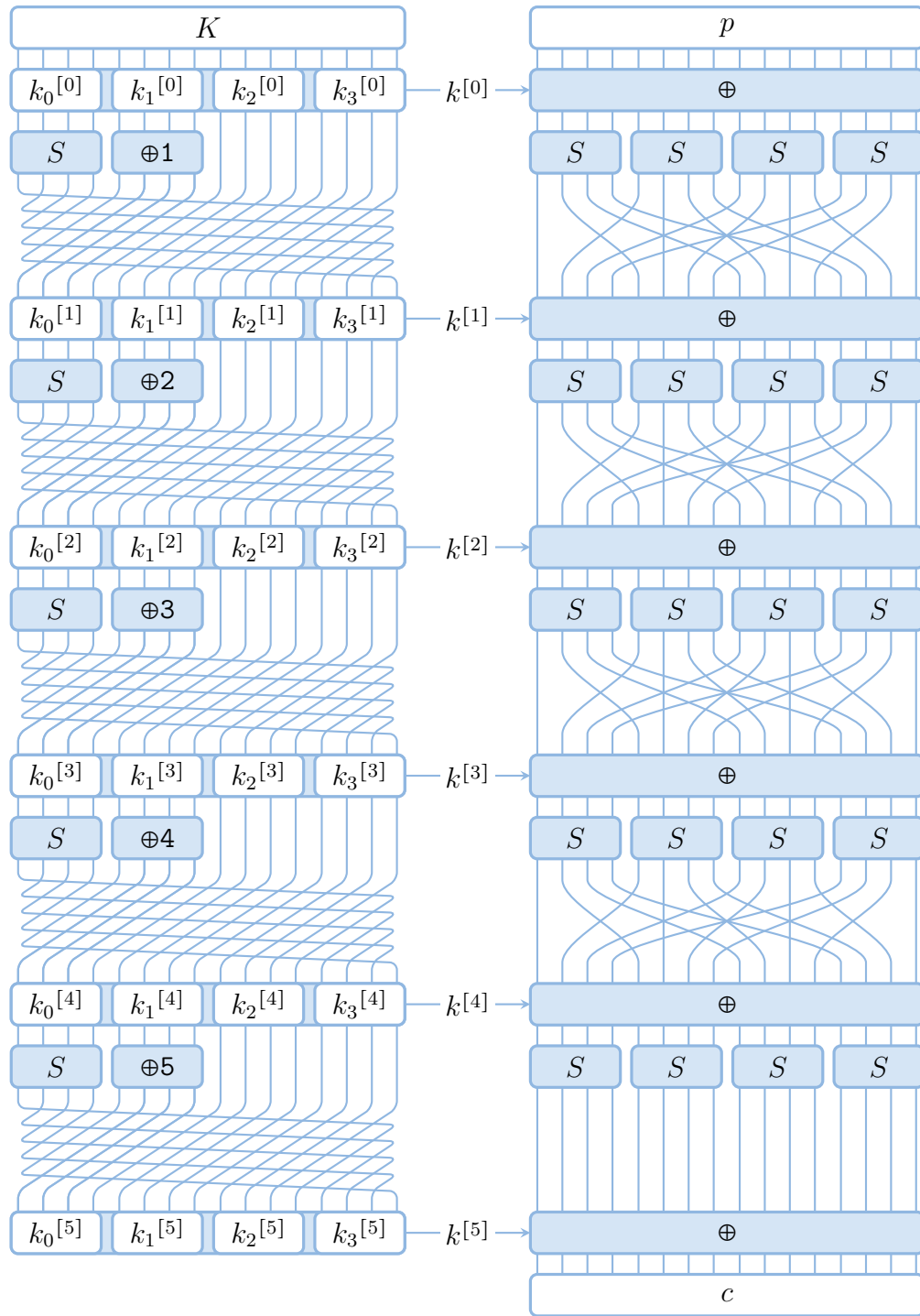
$$\sigma(2, F, F, 1) = (S(2), S(F), S(F), S(1)) = (D, 6, 6, 3) .$$

The diffusion layer π is the bit permutation associated with the permutation ϕ of $\llbracket 0, 16 \rrbracket$ defined by the formula

$$\phi(i) = 4(i \bmod 4) + \left\lfloor \frac{i}{4} \right\rfloor .$$

This bit permutation is drawn from the block cipher PRESENT [17] and its small scale variants SMALL-PRESENT [69]. For instance $\phi(0) = 0$, $\phi(1) = 4$, $\phi(2) = 8$, $\phi(3) = 12$, $\phi(4) = 1$ and

$$\begin{aligned} \pi(\text{D663}) &= \pi \left(\begin{array}{cccc|cccc|cccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right) \\ &= \left(\begin{array}{cccc|cccc|cccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 4 & 8 & 12 & 1 & 5 & 9 & 13 & 2 & 6 & 10 & 14 & 3 & 7 & 11 & 15 \end{array} \right) = 8\text{E79} . \end{aligned}$$



x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	7	3	D	9	C	2	4	8	A	B	1	0	E	F	5	6

Figure 1.3: The encryption algorithm TOYCIIPHER.

Key Schedule		Message encryption	
		0000 0000 0000 0000 = 0000	Plaintext
$k^{[0]}$	0000 0000 0000 0000 = 0000	→ 0000 0000 0000 0000 = 0000	After $\oplus k^{[0]}$
$S/\oplus 1$	0111 0001 0000 0000 = 7100	0111 0111 0111 0111 = 7777	After σ
		0000 1111 1111 1111 = 0FFF	After π
$k^{[1]}$	0010 0000 0000 1110 = 200E	→ 0010 1111 1111 0001 = 2FF1	After $\oplus k^{[1]}$
$S/\oplus 2$	1101 0010 0000 1110 = D20E	1101 0110 0110 0011 = D663	After σ
		1000 1110 0111 1001 = 8E79	After π
$k^{[2]}$	0100 0001 1101 1010 = 41DA	→ 1100 1111 1010 0011 = CFA3	After $\oplus k^{[2]}$
$S/\oplus 3$	1100 0010 1101 1010 = C2DA	1110 0110 0001 1001 = E619	After σ
		1001 1100 1100 0011 = 9CC3	After π
$k^{[3]}$	0101 1011 0101 1000 = 5B58	→ 1100 0111 1001 1011 = C79B	After $\oplus k^{[3]}$
$S/\oplus 4$	0010 1111 0101 1000 = 2F58	1110 1000 1011 0000 = E8B0	After σ
		1110 1000 1010 0010 = E8A2	After π
$k^{[4]}$	1110 1011 0000 0101 = EB05	→ 0000 0011 1010 0111 = 03A7	After $\oplus k^{[4]}$
$S/\oplus 5$	0101 1110 0000 0101 = 5E05	0111 1001 0001 1000 = 7918	After σ
$k^{[5]}$	1100 0000 1010 1011 = C0AB	→ 1011 1001 1011 0011 = B9B3	Ciphertext

Figure 1.4: Encryption of 0000 with TOYCIIPHER using the cipher key $K = 0000$.

The round function F_k consists of an addition with k , a substitution layer then a diffusion layer. With $k = 200E$ and $x = 0FFF$, we have

$$F_k(x) = \pi \circ \sigma(0FFF + 200E) = \pi \circ \sigma(2FF1) = \pi(D663) = 8E79.$$

As explained in Remark 1.11, the last round does not have a diffusion layer. An illustration of the whole encryption of TOYCIIPHER function is given in Figure 1.3.

The key schedule derives 6 round keys $k^{[0]}, \dots, k^{[5]}$ from the cipher key K . The first round key $k^{[0]}$ is equal to the cipher key K . To compute the round key $k^{[i+1]}$ from $k^{[i]}$, apply the S-box S to the first bundle and add a round constant r_i to the next bundle. The constant r_i is just the binary decomposition of the integer $i + 1$. Then rotate by 5 bit positions to the left all the bits to obtain $k^{[i+1]}$. In Figure 1.4, we describe step by step the whole encryption process of the plaintext block 0000 using the cipher key 0000. ▴

1.3. Differential Cryptanalysis

One of the most important and powerful [39, 64] attack against block ciphers is differential cryptanalysis, proposed by Biham and Shamir in [13, 14]. A formalization of this attack was then proposed by Lai, Massey and Murphy in [67]. Differential cryptanalysis is a chosen plaintext attack which requires the encryption of pairs of plaintexts that have a fixed difference. Then, the attack exploits a non-uniform distribution of the differences between pairs of outputs to recover partial information on the last round key of the cipher. This section is organized as follows. First we

give the main idea of the attack and illustrate it with an example. Then, we explore the theoretical framework of this attack in Section 1.3.2

1.3.1. General Idea of the Attack

The difference between two elements x and x^* of \mathbb{F}_2^n is defined to be $x - x^*$ but since each element is its own additive inverse in \mathbb{F}_2^n , it is simply equal to $x + x^*$. The main property used in differential cryptanalysis is that for every round key k , we have

$$(x + k) + (x^* + k) = x + x^*. \quad (1.1)$$

In other words, the difference between x and x^* is invariant under the round key addition.

A successful differential cryptanalysis relies on the existence of a differential holding with high probability, which we define now. Let f be a mapping from \mathbb{F}_2^n to \mathbb{F}_2^n . A *differential* over f is a pair (a, b) of elements of \mathbb{F}_2^n . Given a differential (a, b) , the elements a and b are called the *input* and *output difference patterns* respectively. Then, a differential (a, b) predicts that when two inputs x and x^* have difference a , then their images $f(x)$ and $f(x^*)$ have difference b with a certain probability. It is easily seen that x and x^* have difference a if and only if $x^* = x + a$. Equivalently, a differential (a, b) predicts that when x is uniformly distributed over \mathbb{F}_2^n , the value $f(x) + f(x + a)$ is equal to b with a certain probability. This probability is naturally defined as follows.

Definition 1.15 (Differential Probability). Let $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be a mapping. The *differential probability* of (a, b) over f is denoted by $\text{DP}_f(a, b)$ and defined to be

$$\text{DP}_f(a, b) = \frac{\#\{x \in \mathbb{F}_2^n \mid f(x) + f(x + a) = b\}}{2^n}.$$

Consider an r -round substitution permutation network $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$ and assume that the last round does not have a diffusion layer. Given a cipher key K , we denote by $E_K^{(r-1)}$ the $r-1$ first rounds of the encryption function E_K . Therefore

$$E_K^{(r-1)} = F_{k[r-2]} \circ \cdots \circ F_{k[0]} \quad \text{and hence} \quad E_K = (\alpha_{k[r]} \circ \sigma \circ \alpha_{k[r-1]}) \circ E_K^{(r-1)}.$$

In a classical differential cryptanalysis of E , we do not use a differential over the r -round encryption function E_K , but on the $(r-1)$ -round encryption $E_K^{(r-1)}$. Then, a differential (a, b) over $r-1$ rounds can be exploited in a differential cryptanalysis when its average probability over all the cipher keys is large enough.

The main idea of the attack is the following. Assume that (a, b) is an $(r-1)$ -round differential which holds with probability q for a significant proportion of the cipher keys. Let K be the unknown cipher key. First, generate pairs $(p, p + a)$ of plaintexts and require their encryption under the cipher key K . The pairs obtained are denoted by (c, c^*) . In order to have some pairs (c, c^*) satisfying $c + c^* = b$, we may use $C \times q^{-1}$ plaintext pairs with $C \geq 5$. Assume that k is a candidate for the

last round key $k^{[r]}$. Then we decrypt the last round of each pair (c, c^*) using the candidate k and we denote

$$y = \sigma^{-1}(c + k) \quad \text{and} \quad y^* = \sigma^{-1}(c^* + k).$$

If the candidate key k is the right key, then the equation $y + y^* = b$ should hold with probability q since (a, b) is an $(r-1)$ -round differential. Otherwise, when k is a wrong candidate, we hope that the equation $y + y^* = b$ holds with probability significantly less than q . This assumption is known as the *hypothesis of wrong-key randomization* [51]. Indeed, it seems natural to think that the $(r+1)$ -round differential (a, b) holds with probability less than the $(r-1)$ -round differential (a, b) , and when k is a wrong key, the pair (y, y^*) is equivalent to an $(r+1)$ -round encryption of $(p, p+a)$.

To recover information on the last round key, we may proceed as follows. For each candidate k for the last round key, decrypt the last round for each pair (c, c^*) and save the number N_k of pairs (y, y^*) such that $y + y^* = b$. Then the key k maximizing the value N_k should be equal to the last round key $k^{[r]}$. As will be seen in the next example, we only decrypt partially the last round in an effective cryptanalysis, and thus we recover a few bits of the last round key.

Example 1.16. Let us now present a differential cryptanalysis of our SPN TOY-CIPHER introduced in Example 1.14. Since this cipher consists of five rounds, we must first find a 4-round differential holding with high probability. Finding such a differential is generally not easy because the vast majority of the differentials are useless in a differential cryptanalysis. For instance, the 4-round differential (a, b) with

$$a = (0, 8, 0, 0) \quad \text{and} \quad b = (0, 0, 7, 0)$$

has an average probability over the cipher keys equal to 1.22×2^{-16} . This value was computed via an exhaustive search, which is possible thanks to the small block size and key length of this cipher. A differential cryptanalysis based on this differential would require the encryption of more than 2^{16} different pairs of plaintexts $(x, x+a)$. However, there are only 2^{16} such pairs since there are 2^{16} different blocks.

We will explain in the next section and in Chapter 2 how to find high probability differentials, in this example we explain how such a differential can be used to recover key information. Consider the 4-round differential (a, b) where

$$a = b = (0, 4, 0, 0).$$

The average probability of this differential over all the cipher keys is

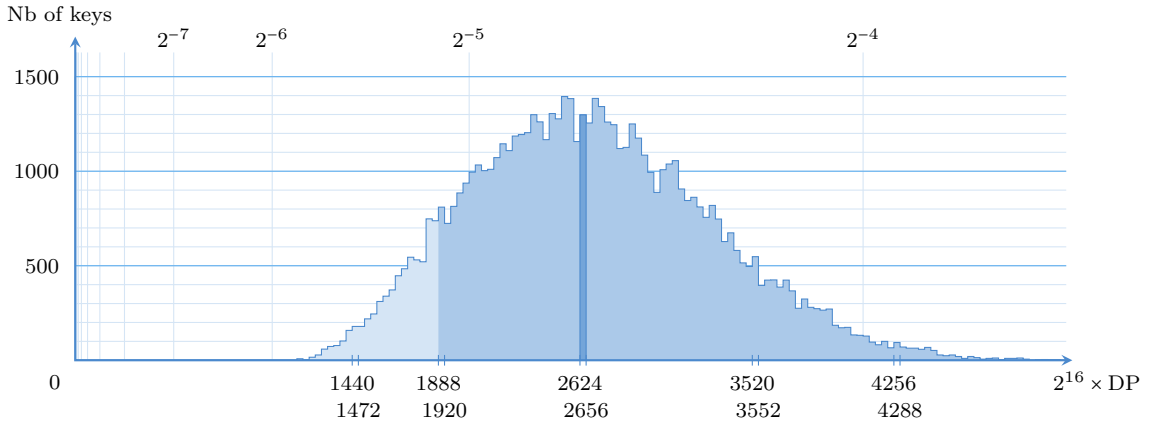
$$\frac{1}{2^{16}} \sum_{K \in \mathbb{F}_2^{16}} \text{DP}_{E_K^{(4)}}(0400, 0400) \approx \frac{2688.2}{2^{16}} \approx 1.31 \times 2^{-5}.$$

The repartition of all the differential probabilities $\text{DP}_{E_K^{(4)}}(0400, 0400)$ is illustrated at the top of Figure 1.5. For instance, there are 1298 cipher keys K such that

$$\frac{2624}{2^{16}} < \text{DP}_{E_K^{(4)}}(0400, 0400) \leq \frac{2656}{2^{16}}.$$

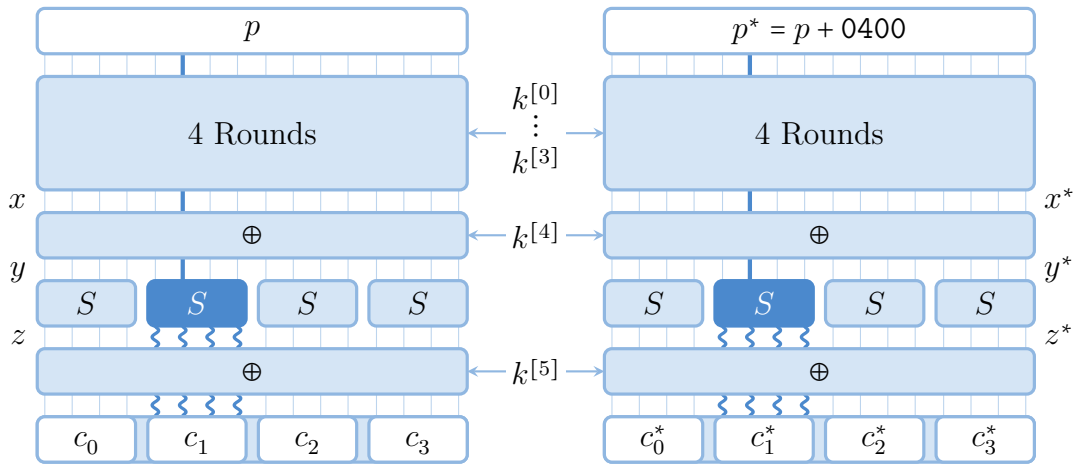
Initialisation: Find a High Probability 4-Round Differential

More than 90% of the cipher keys K satisfy $\text{DP}_{E_K^{(4)}}(0400, 0400) > 1888 / 2^{16} \approx 1.8 \times 2^{-6}$.
 More than 50% of the cipher keys K satisfy $\text{DP}_{E_K^{(4)}}(0400, 0400) > 2624 / 2^{16} \approx 1.3 \times 2^{-5}$.



Part 1: Get Plaintext/Ciphertext Pairs

Choose N plaintext pairs (p, p^*) such that $p + p^* = 0400$ and request the corresponding ciphertext pairs (c, c^*) encrypted under the unknown cipher key K .



Part 2: Recover Some Bits of the Last Round Key

For each candidate $\tilde{k}_1^{[5]}$, decrypt partially the last round of the pairs (c, c^*) such that $c_i = c_i^*$ for all i in $\{0, 2, 3\}$ and $c_1 + c_1^*$ lies in $\{1, 4, 6, 9, B\}$. The key $\tilde{k}_1^{[5]}$ maximizing the number of pairs (y_1, y_1^*) satisfying $y_1 + y_1^* = 4$ should be equal to $k_1^{[5]}$.

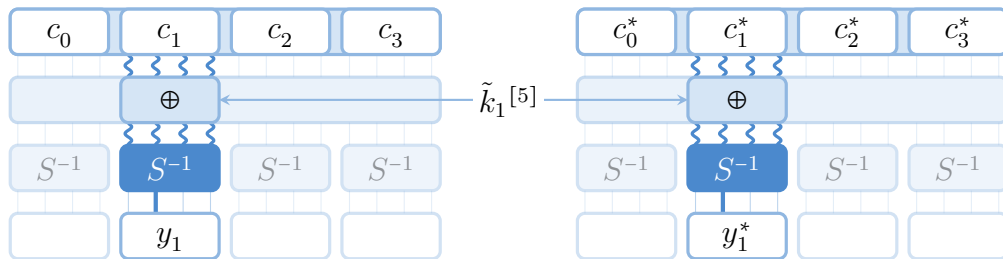


Figure 1.5: A differential cryptanalysis of TOYCIPHER.

Moreover, this differential holds with probability greater than 2^{-5} for 85% of the cipher keys. In view of these results, the differential (a, b) holds with high probability (compared to 2^{-16}) for a significant proportion of the cipher keys and can hence be used in a differential cryptanalysis.

As explained before, a differential cryptanalysis requires the encryption of $C \times q^{-1}$ pairs of plaintexts where q is the average probability of the 4-round differential. With $C = 5$, this amounts to

$$N = 5 \times \frac{2^{16}}{2688.2} \approx 122 \approx 2^7$$

pairs of chosen plaintexts. To generate these pairs, choose N random plaintexts p and form the pairs $(p, p + 0400)$. Thus, all the bits of p and $p^* = p + 0400$ are equal except the bit with index 5, starting from 0. Let K be the unknown cipher key and request for the encryption of all these pairs. Then, we obtain N pairs (c, c^*) such that

$$c = E_K(p) \quad \text{and} \quad c^* = E_K(p + 0400).$$

Let (p, p^*) be one of these pairs and denote by (x, x^*) its 4-round encryption, that is $x = E_{K(4)}(p)$ and $x^* = E_{K(4)}(p^*)$. The following reasoning is illustrated in Part 1 of Figure 1.5. Assume that $x + x^* = 0400$. Such a pair is called a *right pair*, or equivalently we say that (p, p^*) *follows* the differential (a, b) . By extension, its corresponding ciphertext pair (c, c^*) is also said to be a right pair. Adding the next round key, the pair (x, x^*) becomes

$$(y, y^*) = (x + k^{[4]}, x^* + k^{[4]}).$$

By assumption and according to Equation (1.1), the difference between y and y^* remains unchanged and is equal to 0400. We must now understand how this difference propagates through the substitution layer. First, note that

$$y_0 = y_0^*, \quad y_1 = y_1^* + 4, \quad y_2 = y_2^*, \quad y_3 = y_3^*.$$

Denote by z and z^* the images of y and y^* under σ . It goes without saying that for each i in $\{0, 2, 3\}$, we have $z_i = z_i^*$ and hence $z_i + z_i^* = 0$. It remains to explain what are the possible values for the difference $z_1 + z_1^*$. This is done by computing the values $z_1 + z_1^*$ for all possibles y_1 and $y_1^* = y_1 + 4$ in \mathbb{F}_2^4 :

$$\begin{aligned} S(0) + S(4) &= S(4) + S(0) = \mathbf{C} + 7 = \mathbf{B}, & S(8) + S(\mathbf{C}) &= S(\mathbf{C}) + S(8) = \mathbf{E} + \mathbf{A} = \mathbf{4}, \\ S(1) + S(5) &= S(5) + S(1) = 2 + 3 = \mathbf{1}, & S(9) + S(\mathbf{D}) &= S(\mathbf{D}) + S(9) = \mathbf{F} + \mathbf{B} = \mathbf{4}, \\ S(2) + S(6) &= S(6) + S(2) = 4 + \mathbf{D} = \mathbf{9}, & S(\mathbf{A}) + S(\mathbf{E}) &= S(\mathbf{E}) + S(\mathbf{A}) = 5 + 1 = \mathbf{4}, \\ S(3) + S(7) &= S(7) + S(3) = 8 + 9 = \mathbf{1}, & S(\mathbf{B}) + S(\mathbf{F}) &= S(\mathbf{F}) + S(\mathbf{B}) = 6 + 0 = \mathbf{6}. \end{aligned}$$

Therefore, the difference $z_1 + z_1^*$ lies in the set $\{1, 4, 6, 9, \mathbf{B}\}$. Finally, c and c^* are obtained by adding the last round key $k^{[5]}$ to z and z^* respectively, so $c + c^* = z + z^*$. To summarize, we have proven that if (p, p^*) is a right pair, then

$$c_0 + c_0^* = c_2 + c_2^* = c_3 + c_3^* = 0, \quad \text{and} \quad c_1 + c_1^* \in \{1, 4, 6, 9, \mathbf{B}\}. \quad (1.2)$$

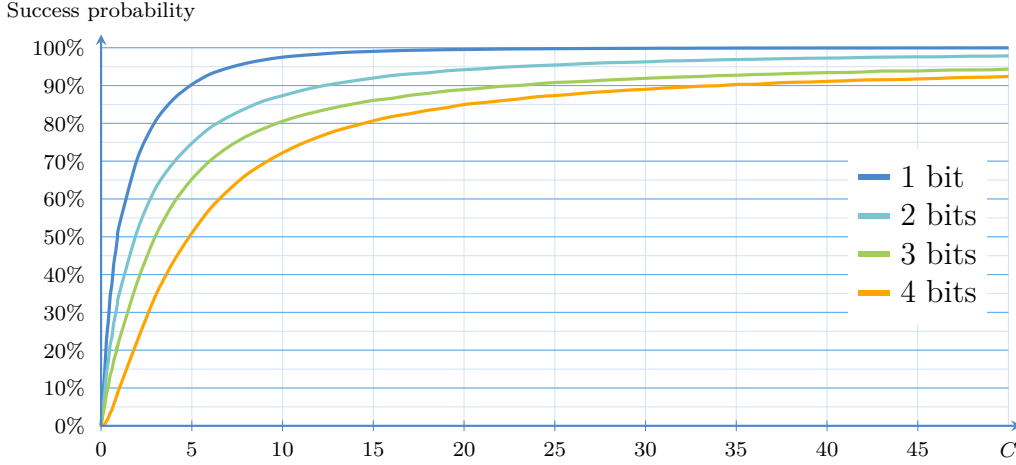


Figure 1.6: A differential cryptanalysis of TOYCIIPHER.

To recover information on the last round key, we must count for each candidate key k the number of pairs (c, c^*) satisfying the equation

$$\sigma^{-1}(c + k) + \sigma^{-1}(c^* + k) = 0400.$$

According to the preceding discussion, if a pair (c, c^*) does not satisfy (1.2), then this pair is necessarily a *wrong pair*. To avoid useless computing, we apply a filtering process which discards the wrong pairs and we denote by \mathcal{F} the set of the filtered pairs. Let k be a candidate key and let i be a bundle index in $\{0, 2, 3\}$. For each filtered pair (c, c^*) , we have

$$c_i = c_i^* \quad \text{and hence} \quad S^{-1}(c_i + k_i) + S^{-1}(c_i^* + k_i) = 0.$$

Therefore, this differential gives no information on $k_i^{[5]}$ for each i in $\{0, 2, 3\}$, so we can only recover information on $k_1^{[5]}$. This is good news because we would otherwise have to decrypt the last round for all of the 2^{16} possible round keys, yielding a complexity greater than the brute force. Finally, the cryptanalysis ends as follows. For each k_1 in \mathbb{F}_2^4 , compute its score

$$N_{k_1} = \#\{(c, c^*) \in \mathcal{F} \mid S^{-1}(c_1 + k_1) + S^{-1}(c_1^* + k_1) = 4\}.$$

Then, the higher the score N_{k_1} is, the more likely k_1 is equal to $k_1^{[5]}$. This step is illustrated in Part 2 of Figure 1.5. The experimental success probabilities of this differential cryptanalysis with respect to the constant C are given in Figure 1.6. Define the rank of the key $k_1^{[5]}$ to be

$$\text{Rk} = \#\{k_1 \in \mathbb{F}_2^4 \mid N_{k_1} \geq N_{k_1^{[5]}}\}.$$

If the rank is equal to 1, then the cryptanalysis recovers the right bundle within the set \mathbb{F}_2^4 , giving four bits of information. When the rank is less than 2, two choices remain for $k_1^{[5]}$ instead of 16. Thus, we have at least three bits of information. Similarly, we have (at least) two bits of information when $Nk \geq 4$ and one bit if $Nk \geq 8$. As can be seen in Figure 1.6, when $C = 5$ this attack recovers one bit of information with probability 90.2%, two and three bits with probability 74.7% and 65.2%, and recovers the exact bundle with probability 50.9%. ▀

1.3.2. Differential Trails

Having explained how a high probability differential can be exploited in a cryptanalysis, now is the time to present the theory of differentials. This presentation is inspired by the works of Lai, Massey [67] and Daemen, Rijmen [39, 40]. In this section we consider a generic r -round SPN $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$ such that for each cipher key K ,

$$E_K = F_{k[r-1]} \circ \cdots \circ F_{k[0]} \quad \text{with} \quad F_{k[i]} = \pi \circ \sigma \circ \alpha_{k[i]}.$$

It is worth observing that the last round includes and ends with a diffusion layer. This definition makes sense here because the differentials used in an attack have less rounds than the whole cipher. Thus, the SPN considered here should be thought as a reduced-round version of the cipher attacked.

The standard method used to find a high probability differential relies on the study of a difference propagation through the components of the SPN. We have already seen that a difference remains unchanged by a key addition. The next proposition describes how a difference propagates through the substitution and diffusion layers.

Proposition 1.17. Let a and b be two difference patterns in \mathbb{F}_2^{nm} . Then

$$\text{DP}_\sigma(a, b) = \prod_{i=0}^{m-1} \text{DP}_{S_i}(a_i, b_i) \quad \text{and} \quad \text{DP}_\pi(a, b) = \begin{cases} 1 & \text{if } \pi(a) = b, \\ 0 & \text{otherwise.} \end{cases}$$

In other words, given an input difference pattern a , each S-box S_i transforms independently a_i to b_i with a certain probability and the diffusion layer always maps a to $\pi(a) = b$. Following the idea of propagation of a difference through the encryption process, we introduce the next definition.

Definition 1.18 (Differential Trail). An r -round *differential trail* is a family $\mathcal{T} = (a^{[0]}, \dots, a^{[r]})$ of $r + 1$ difference patterns in \mathbb{F}_2^{nm} . Let K be a cipher key. The fixed-key differential probability of \mathcal{T} is defined to be

$$\text{DP}_{E_K}(\mathcal{T}) = \frac{\#\{x \in \mathbb{F}_2^{nm} \mid \forall 1 \leq i \leq r, E_K^{(i)}(x) + E_K^{(i)}(x + a^{[0]}) = a^{[i]}\}}{2^{nm}}.$$

Let (x, x^*) be a pair of plaintexts. We should say that (x, x^*) *follows* the differential trail $\mathcal{T} = (a^{[i]})_{i \leq r}$ if

- the difference between x and x^* is equal to $a^{[0]}$, and
- for each $1 \leq i \leq r$, the difference between the i -round encryptions of x and x^* is equal to $a^{[i]}$.

Using this vocabulary, the fixed-key differential probability of \mathcal{T} can equivalently be defined as the probability that a pair chosen uniformly at random follows the trail \mathcal{T} given that its difference is equal to the input pattern $a^{[0]}$. Therefore, the trail \mathcal{T} predicts the evolution of an input difference after each round of the encryption

process whereas a differential only predicts its output difference. These two concepts are related via the following proposition.

Proposition 1.19. Let (a, b) be an r -round differential and let K be a cipher key. Denote by $E_{a,b}$ the set of all trails $(a^{[i]})_{i \leq r}$ such that $a^{[0]} = a$ and $a^{[r]} = b$. Then

$$\text{DP}_{E_K}(a, b) = \sum_{\mathcal{T} \in E_{a,b}} \text{DP}_{E_K}(\mathcal{T}).$$

This result is in fact quite intuitive since each pair following the differential naturally follows one and only one trail, namely the trail consisting of its intermediate differences. Conversely, a pair following a trail $(a^{[i]})_{i \leq r}$ such that $a^{[0]} = a$ and $a^{[r]} = b$ obviously follows the r -round differential (a, b) . So far we have only considered fixed-key probabilities. However, these results do not describe what can be expected when attacking an unknown cipher key. For this purpose we introduce the next definition.

Definition 1.20 (DP(\mathcal{T})). The *(expected) differential probability* of a trail \mathcal{T} , denoted by $\text{DP}(\mathcal{T})$, is the average fixed-key differential probability of \mathcal{T} over the associated long-key cipher.

Recall that the long-key cipher associated with our SPN is the cipher obtained by disregarding the key schedule and considering independent round keys. Explicitly, the differential probability of \mathcal{T} is given by

$$\text{DP}(\mathcal{T}) = \frac{1}{(2^{nm})^r} \times \sum_{K \in (\mathbb{F}_2^{nm})^r} \text{DP}_{E_K}(\mathcal{T}).$$

Remark 1.21. By virtue of Proposition 1.17, a difference can have a probabilistic transition only during the substitution layer. Given a trail $\mathcal{T} = (a^{[i]})_{i \leq r}$, we denote by $b^{[i]}$ the element $\pi^{-1}(a^{[i+1]})$ for each $i < r$. Thus, an r -round trail can alternatively be seen as the sequence $((a^{[i]}, b^{[i]}))_{i < r}$.

Theorem 1.22. Let $\mathcal{T} = (a^{[i]})_{i \leq r}$ be a differential trail. The differential probability of \mathcal{T} is given by

$$\text{DP}(\mathcal{T}) = \prod_{i=0}^{r-1} \text{DP}_{\sigma}(a^{[i]}, b^{[i]}) = \prod_{i=0}^{r-1} \prod_{j=0}^{m-1} \text{DP}_{S_j}(a_j^{[i]}, b_j^{[i]}).$$

This theorem has a significant practical impact since the differential probability of a trail can be computed by multiplying a few differential probabilities over the S-boxes. Computing the full differential probability matrix DP_{S_j} of the n -bit S-box S_j has complexity $O(2^{2n})$. Since substitution-permutation networks generally have n -bit S-boxes with n less than or equal to 8, all these matrices are easily computed. Moreover, the formula of the previous theorem can be simplified further by introducing the notion of active S-boxes.

Definition 1.23 (Active S-Box). Let $\mathcal{T} = (a^{[i]})_{i \leq r}$ be a differential trail with a nonzero differential probability. The S-box S_j is said to be *active* at round i if the pattern $a_j^{[i]}$ is nonzero, otherwise S_j is *inactive*.

Proposition 1.24. Let $\mathcal{T} = (a^{[i]})_{i \leq r}$ be a nonzero probability differential trail. Let $i < r$ and $j \leq m$ be nonnegative integers. Then $a_j^{[i]} = 0_n$ if and only if $b_j^{[i]} = 0_n$. Consequently, Theorem 1.22 can be restated as

$$\text{DP}(\mathcal{T}) = \prod_{i,j \mid a_j^{[i]} \neq 0} \text{DP}_{S_j}(a_j^{[i]}, b_j^{[i]}).$$

Proof. To simplify the notations, denote by a' and b' the patterns $a_j^{[i]}$ and $b_j^{[i]}$. According to Theorem 1.22, $\text{DP}_{S_j}(a', b')$ is nonzero as $\text{DP}(\mathcal{T})$ is nonzero by hypothesis. Because S_j is one-to-one, $\text{DP}_{S_j}(a', 0_n)$ is nonzero if and only if $a' = 0_n$. Further, $\text{DP}_{S_j}(0_n, b')$ is nonzero only when $b' = 0_n$. Finally, observe that $\text{DP}_{S_j}(0_n, 0_n) = 1$. The result follows. ■

Definition 1.25 (EDP). The *expected differential probability* of an r -round differential (a, b) , denoted by $\text{EDP}(a, b)$, is the average fixed-key differential probability of (a, b) over the associated long-key cipher.

Theorem 1.26. Let (a, b) be an r -round differential. Denote by $E_{a,b}$ the set of all trails $(a^{[i]})_{i \leq r}$ such that $a^{[0]} = a$ and $a^{[r]} = b$. The expected differential probability of (a, b) is given by

$$\text{EDP}(a, b) = \sum_{\mathcal{T} \in E_{a,b}} \text{DP}(\mathcal{T}).$$

The expected differential probability of a differential is the theoretical value reflecting its usefulness. However, this notion has two downsides. First, the set $E_{a,b}$ generally grows exponentially with the number of rounds and it should be very difficult to enumerate all its trails. Consequently, in real size substitution permutation networks, it is almost impossible to compute an expected differential probability. However, this value can be approximated using several high probability differential trials.

Secondly, the expected differential probability does not take into account the effect of the key schedule and provides only an average value. Thus, in a cryptanalysis, we tacitly assume that the fixed-key differential probability over the cipher key being attacked is approximately equal to its expected differential probability. This assumption is known as the *hypothesis of stochastic equivalence* [67].

Example 1.27. In Example 1.16, we have considered the 4-round differential (a, b) with $a = b = 0400$. Denote by (x, x^*) the pair $(4100, 4500)$ of plaintexts and consider the cipher key $K = 0000$. Since $x + x^*$ is equal to 0400 , this input pair has the required input difference. Throughout the 4-round encryption process, this pair

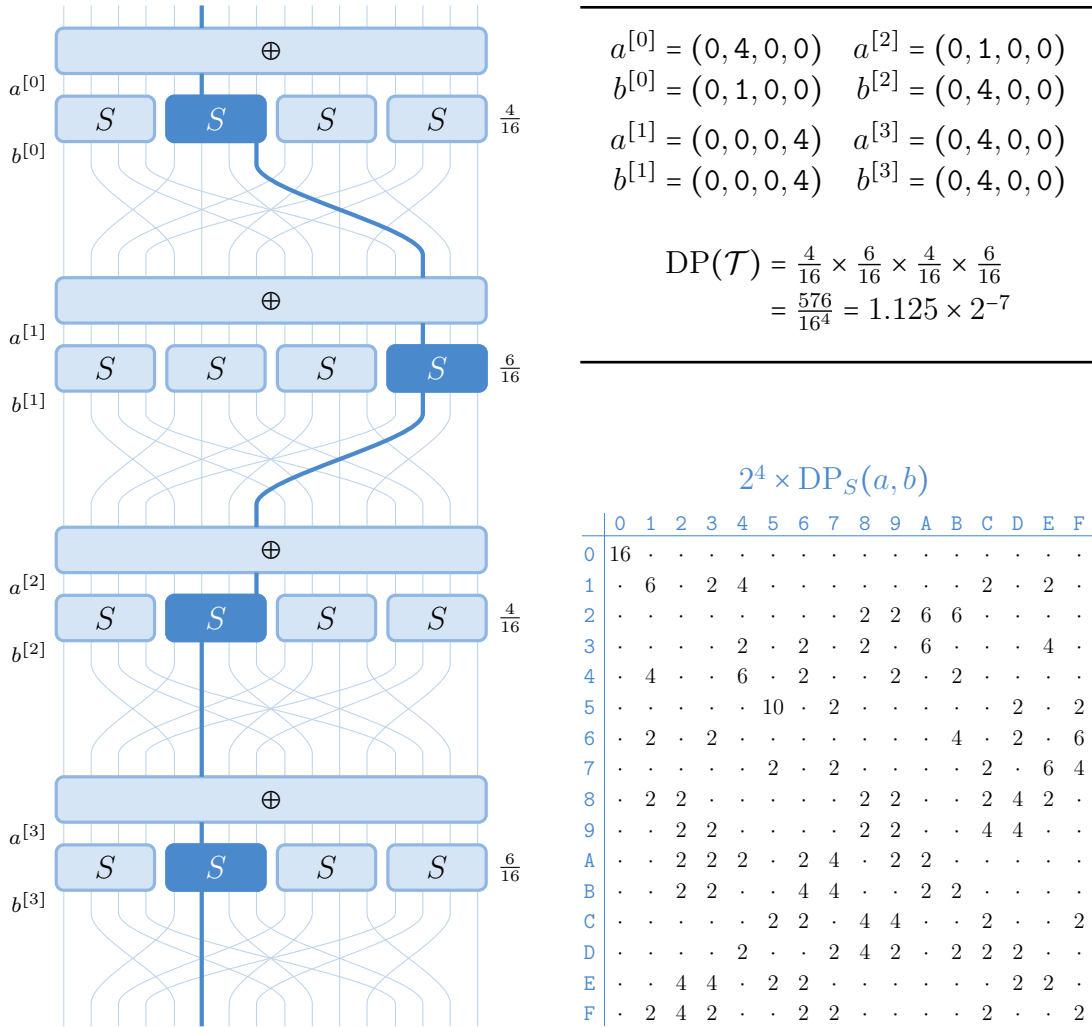


Figure 1.7: A differential trail included in the 4-round differential (0400, 0400).

is transformed as follows:

	x	x^*		Difference
	4100	4500	→	0400
$E_K^{(1)} :$	8B77	8B73	→	0001
$E_K^{(2)} :$	3019	3119	→	0100
$E_K^{(3)} :$	B265	B665	→	0400
$E_K^{(4)} :$	795F	7D5F	→	0400

Since the output difference is equal to the output pattern b , this pair follows the differential (a, b) . More precisely, this pair follows the differential trail $\mathcal{T} = (a^{[i]})_{i \leq 4}$ where

$$a^{[0]} = 0400, \quad a^{[1]} = 0001, \quad a^{[2]} = 0100, \quad a^{[3]} = 0400, \quad a^{[4]} = 0400.$$

This trail is illustrated in Figure 1.7 where $b^{[i]} = \pi^{-1}(a^{[i]})$, as suggested by Remark 1.21. The active S-boxes are emphasized. Thus there are only four active S-boxes in this trail, which is the minimum. The differential probability matrix of S is also given in the same figure. As seen in Example 1.16,

$$S(y_1) + S(y_1 + 4) = \begin{cases} 1 & \text{if } y_1 \in \{1, 3, 5, 7\}, \\ 4 & \text{if } y_1 \in \{8, 9, A, C, D, E\}, \\ 6 & \text{if } y_1 \in \{B, F\}, \end{cases} \quad \begin{cases} 9 & \text{if } y_1 \in \{2, 6\}, \\ B & \text{if } y_1 \in \{0, 4\}. \end{cases}$$

This relation explains the row indexed by 4 of the matrix DP_S . By virtue of Corollary 1.24, it suffices to multiply the probabilities of the active S-boxes to find that the (expected) probability of the trail \mathcal{T} is equal to 1.125×2^{-7} , as shown in Figure 1.7.


It turns out that there are six trails associated with the differential (a, b) . These trails are given in Figure 1.8, sorted by differential probability. According to Theorem 1.26, the expected differential probability of (a, b) is the sum of the differential probabilities of these trails, that is

$$\text{EDP}(a, b) = \sum_{i=1}^6 \text{DP}(\mathcal{T}_i) = \frac{346133}{8388608} \approx 1.32 \times 2^{-5}.$$

Recall that in the preceding example, we have found that the average fixed-key of this differential including the key schedule is approximately equal to 1.31×2^{-5} . Thus, the theoretical expected differential probability is very close to the real value in this example.

To conclude, it should be mentioned that the trail \mathcal{T}_1 is not the 4-round trail which has the highest probability. Using the algorithm given in the next chapter, it can be proven that the optimal 4-round trail is the trail $\mathcal{T}_o^{(=)}(a^{[i]})_{i \leq 4}$ such that $a^{[i]} = 0505$ for every $i \leq 4$. This trail has a differential probability equal to $(\frac{10}{16})^8 \approx 1.49 \times 2^{-6}$ which is greater than $\text{DP}(\mathcal{T}_1) \approx 1.2 \times 2^{-6}$. However, we have computed that

$$\frac{1}{2^{16}} \sum_{K \in \mathbb{F}_2^{16}} \text{DP}_{E_K^{(4)}}(0505, 0505) \approx \frac{1714.3}{2^{16}} \approx 1.67 \times 2^{-6}.$$

Therefore, the differential associated with an optimal trail is not necessarily an optimal differential. 

	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6
$a^{[0]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)
$b^{[0]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)	(0, B, 0, 0)	(0, B, 0, 0)
$a^{[1]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 0, 0, 4)	(0, 0, 0, 4)	(4, 0, 4, 4)	(4, 0, 4, 4)
$b^{[1]}$	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, 0, 0, 4)	(0, 0, 0, 1)	(4, 0, 4, 9)	(9, 0, 9, 4)
$a^{[2]}$	(0, 4, 0, 0)	(0, 0, 0, 4)	(0, 1, 0, 0)	(0, 0, 0, 1)	(1, A, 0, 1)	(A, 1, 0, A)
$b^{[2]}$	(0, 4, 0, 0)	(0, 0, 0, 4)	(0, 4, 0, 0)	(0, 0, 0, 4)	(4, 4, 0, 4)	(4, 4, 0, 4)
$a^{[3]}$	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, D, 0, 0)	(0, D, 0, 0)
$b^{[3]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)
DP	$6^4/16^4$	$4^2 \times 6^2/16^4$	$4^2 \times 6^2/16^4$	$4^4/16^4$	$2^4 \times 4^2 \times 6^2/16^8$	$2^6 \times 4 \times 6/16^8$
\approx	1.2×2^{-6}	1.1×2^{-7}	1.1×2^{-7}	1.0×2^{-8}	1.1×2^{-19}	1.5×2^{-22}

Figure 1.8: The trails composing the 4-round differential (0400, 0400).

1.4. Linear cryptanalysis

After differential cryptanalysis, linear cryptanalysis is the main attack against block ciphers. This cryptanalysis was introduced by Matsui in [74, 75] for the DES and was the first attack which recovered experimentally a DES key. However, it should be mentioned that the idea of linear cryptanalysis was proposed earlier by Tardy-Corffdir and Gilbert [96] in an attack against the cipher FEAL-4.

Linear cryptanalysis is a known plaintext attack, which is an advantage over differential cryptanalysis. The main idea of this attack is to use a linear approximation of a reduced-round version of the cipher to recover information on some round keys. As was the case for the presentation of differential cryptanalysis, we first formalize the idea of linear cryptanalysis and give an example. Then we describe the theory of this attack in Section 1.4.2.

1.4.1. General Idea of the Attack

For a linear cryptanalysis to be successful, one must find a linear approximation of the cipher with high linear potential. First, let us define the concepts of linear approximation and linear potential. Let f be a mapping from \mathbb{F}_2^n to \mathbb{F}_2^n . Intuitively, we want to approximate a linear combination of the output bits of f by a linear combination of its input bits. In other words, we want a relation of the form

$$\langle a, x \rangle = \langle b, f(x) \rangle, \quad (1.3)$$

where the n -bit vectors a and b are called the *input* and *output selection patterns* of the approximation. Thus, a *linear approximation* over f is simply defined to be a pair (a, b) of elements of \mathbb{F}_2^n . Of course, such an approximation holds with a certain

probability. But it is worthwhile to note that, if Equation (1.3) quite never holds then the relation

$$\langle a, x \rangle = \langle b, f(x) \rangle + 1 \quad (1.4)$$

holds with high probability. From a cryptanalytic point of view, using Equation (1.3) or (1.4) does not matter as they yield the same amount of information. The worst case is when (1.3) holds for exactly the half of the inputs x . In this case, the left side gives no information on the right side, and hence on f . The usefulness of an approximation is characterized by its correlation or linear potential.

Definition 1.28 (Correlation and Linear Potential). Let $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be a mapping and (a, b) be an approximation over f . The *correlation* of the approximation (a, b) is defined to be

$$C_f(a, b) = 2 \cdot \frac{\#\{x \in \mathbb{F}_2^n \mid \langle a, x \rangle = \langle b, f(x) \rangle\}}{2^n} - 1.$$

The *linear potential* LP of (a, b) is the square of its correlation, namely

$$LP_f(a, b) = C_f(a, b)^2.$$

Remark 1.29. The definition of the correlation can be equivalently restated as

$$C_f(a, b) = 2 \cdot \mathbb{P}_x(\langle a, x \rangle = \langle b, f(x) \rangle) - 1.$$

Thus, the correlation of any approximation ranges from -1 to 1 included. Then, the linear potential of an approximation ranges from 0 to 1 . A correlation or linear potential equal to zero gives no information. The closer the absolute correlation or linear potential is to one, the more information it yields on f . Finally, it should be noted that several authors speak about *linear probability* rather than *potential*. We strongly encourage the term *potential* as this quantity is not a probability.

As for differential cryptanalysis, consider an r -round substitution permutation network $E : \mathbb{F}_2^k \times \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$ and assume that the last round does not have a diffusion layer, thus

$$E_K = (\alpha_{k[r]} \circ \sigma \circ \alpha_{k[r-1]}) \circ E_K^{(r-1)}.$$

A classical linear cryptanalysis of E is based on an approximation (a, b) over the $(r-1)$ -round encryption $E_K^{(r-1)}$ which has high linear potential for virtually all cipher keys K . Let K be a cipher key. Then note that

$$\begin{aligned} \langle a, x \rangle &= \langle b, E_K^{(r-1)}(x) + k^{[r-1]} \rangle \\ \iff \langle a, x \rangle &= \langle b, E_K^{(r-1)}(x) \rangle + \langle b, k^{[r-1]} \rangle. \end{aligned}$$

Since $\langle b, k^{[r-1]} \rangle$ does not depend on x , the correlation of the approximation (a, b) over $\alpha_{k[r-1]} \circ E_K^{(r-1)}$ is equal to the correlation of (a, b) over $E_K^{(r-1)}$ up to the sign. Therefore,

$$\begin{aligned} C_{R'}(a, b) &= \pm C_R(a, b) \\ LP_{R'}(a, b) &= LP_R(a, b) \end{aligned} \quad \text{where} \quad \begin{cases} R = E_K^{(r-1)}, \\ R' = \alpha_{k[r-1]} \circ E_K^{(r-1)}. \end{cases} \quad (1.5)$$

Assume that (a, b) is an $(r - 1)$ -round approximation with linear potential q for a significant fraction of the cipher keys and let K denote the unknown cipher key. For the cryptanalysis to be successful, it turns out that we need $N = C \times q^{-1}$ known plaintext/ciphertext pairs (p, c) . To recover information on the last round key, proceed as follows. For each candidate key k for $k^{[r]}$, compute the value

$$P_k = \left(2 \times \#\{(p, c) \mid \langle a, p \rangle = \langle b, \sigma^{-1}(c + k) \rangle\} - N \right)^2.$$

Then the key k maximizing the value P_k should be equal to the last round key $k^{[r]}$. Again, the assumption that a wrong key k should have a value P_k less than $P_{k^{[r]}}$ is called the *hypothesis of wrong-key randomization* [51].

Example 1.30. In this example we describe a linear cryptanalysis of our TOYCI-PHER introduced in Example 1.14. Again we will not focus on how to find a suitable linear approximation of the reduced-round cipher but rather explain how to use one. Consider the 4-round linear approximation (a, b) where

$$a = b = (0, 0, 2, 0).$$

The average linear potential of this approximation over every cipher key is

$$\frac{1}{2^{16}} \sum_{K \in \mathbb{F}_2^{16}} \text{LP}_{E_K^{(4)}}(0020, 0020) \approx \frac{6914.6}{2^{16}} \approx 1.69 \times 2^{-4}.$$

This value was computed via an exhaustive search. The distribution of all linear potentials $\text{LP}_{E_K^{(4)}}(0020, 0020)$ is given at the top of Figure 1.9. For instance, the inequalities

$$\frac{4416}{2^{16}} < \text{LP}_{E_K^{(4)}}(0020, 0020) \leq \frac{4480}{2^{16}}$$

hold for 2400 cipher keys. Compared with Figure 1.5, the repartition of these potentials is more complicated and key-dependent than the repartition of the differential probabilities. We will explain this weird behavior in Section 1.4.2.

Let K be an unknown cipher key. This linear cryptanalysis is known to be successful only when $C \times q^{-1}$ plaintext/ciphertext pairs are available. With $C = 5$, this attack requires

$$N = 5 \times \frac{2^{16}}{6914.6} \approx 47 \approx 2^6$$

known plaintexts. Assume we are given N pairs (p, c) such that $c = E_K(c)$. Let k be a candidate of $k^{[5]}$. Following the principle of linear cryptanalysis, we must compute the value

$$P_k = \left(2 \times \#\{(p, c) \mid \langle a, p \rangle = \langle b, \sigma^{-1}(c + k) \rangle\} - N \right)^2. \quad (1.6)$$

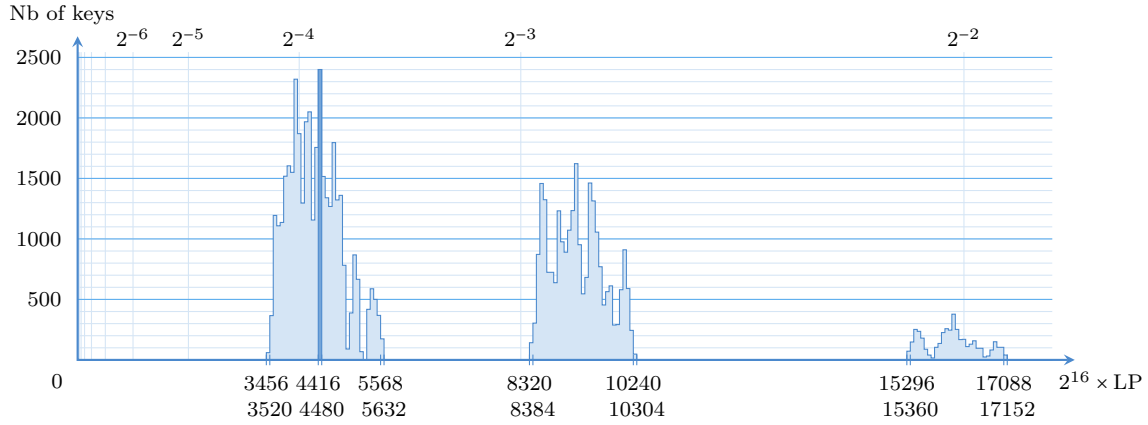
Observe that for each pair (p, c) , it holds that

$$\begin{aligned} \langle a, p \rangle = \langle b, \sigma^{-1}(c + k) \rangle &\iff \langle 0020, p \rangle = \langle 0020, \sigma^{-1}(c + k) \rangle \\ &\iff \langle 2, p_2 \rangle = \langle 2, S^{-1}(c_2 + k_2) \rangle. \end{aligned}$$

 Initialisation: Find a High Potential 4-Round approximation

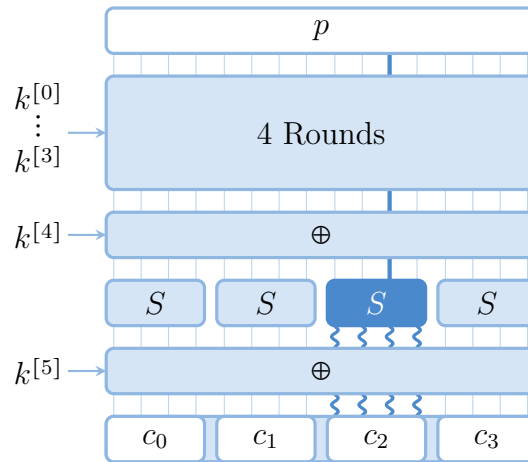
More than 90% of the cipher keys K satisfy $\text{LP}_{E_K^{(4)}}(0020, 0020) \geq 3840 / 2^{16} \approx 1.8 \times 2^{-6}$.

More than 43% of the cipher keys K satisfy $\text{LP}_{E_K^{(4)}}(0020, 0020) \geq 8384 / 2^{16} \approx 1.2 \times 2^{-5}$.



 Part 1: Get Plaintext/Ciphertext Pairs

Choose N plaintext pairs (p, p^*) such that $p + p^* = 0400$ and request the corresponding ciphertext pairs (c, c^*) encrypted under the unknown cipher key K .



 Part 2: Recover Some Bits of the Last Round Key

For each candidate $\tilde{k}_2^{[5]}$, decrypt partially the last round for every ciphertext.

The key $\tilde{k}_2^{[5]}$ maximizing $(2 \times \#\{(p_2, y_2) \mid \langle p_2, 2 \rangle = \langle y_2, 2 \rangle\} - N)^2$ should be equal to $k_1^{[5]}$.

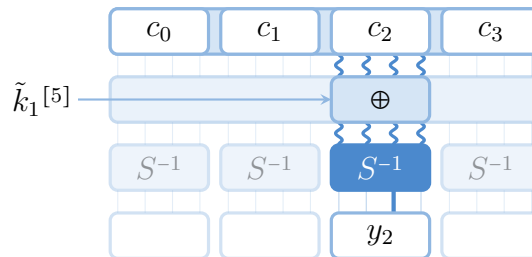


Figure 1.9: A linear cryptanalysis of TOYCIPHER.

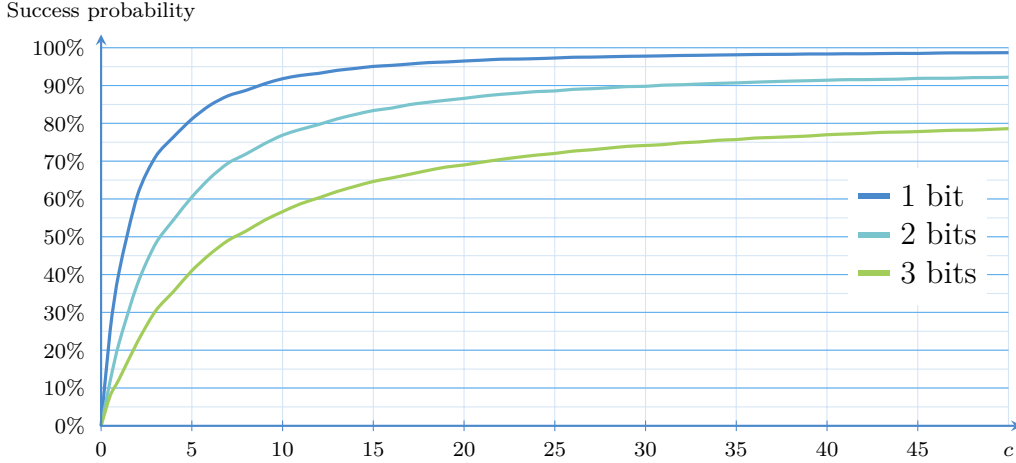


Figure 1.10: A differential cryptanalysis of TOYCIPIPER.

Replacing in (1.6) yields

$$P_k = \left(2 \times \#\{(p, c) \mid \langle 2, p_2 \rangle = \langle 2, S^{-1}(c_2 + k_2) \rangle\} - N \right)^2. \quad (1.7)$$

Since the value P_k does not depend on k_0 , k_1 and k_3 , the cryptanalysis cannot give any information on the corresponding bundles of the last round key. To recover information on $k_2^{[5]}$, proceed as follows. For each candidate k_2 for $k_2^{[5]}$, compute the value P_{k_2} given in (1.7). Then the higher P_{k_2} is, the more likely k_2 is equal to $k_2^{[5]}$. This cryptanalysis is illustrated in Parts 1 and 2 of Figure 1.9 and its success probability is given in Figure 1.10. For instance, if $C = 5$, the attack recovers one, two and three bits of information with probability 81.1%, 60.5% and 41.0% respectively. Unlike the differential cryptanalysis in Example 1.16, this attack never obtains four bits of information. This does not mean that the correct bundle $k_2^{[5]}$ is never recovered but that there is at least another key with a score greater than or equal to its score. ▴

1.4.2. Linear Approximations and Linear Trails

In this section, we explore the theory of linear approximations. All the results and definition are drawn from the works of Daemen and Rijmen [39, 40]. Let f be a mapping from \mathbb{F}_2^n to \mathbb{F}_2^n . The matrix C_f formed by the correlations between all the selection patterns is called the *correlation matrix* of f . The next lemma explains how the correlation matrix of a composition can be derived from the correlation matrices of its components.

Lemma 1.31. Let f and g be two mappings from \mathbb{F}_2^n to \mathbb{F}_2^n . The correlation matrix of the composite $g \circ f$ is equal to $C_f \times C_g$. Thus, for all a, b in \mathbb{F}_2^n , we have

$$C_{g \circ f}(a, b) = \sum_{i \in \mathbb{F}_2^n} C_f(a, i) \times C_g(i, b).$$

Consider a generic r -round substitution-permutation network $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$

where the encryption function can be expressed for each cipher key K as

$$E_K = F_{k[r-1]} \circ \cdots \circ F_{k[0]} \quad \text{with} \quad F_{k[i]} = \pi \circ \sigma \circ \alpha_{k[i]}.$$

Again, the last round includes and ends with a diffusion layer. First, we look at the correlation of an approximation over the basic steps of the round function, namely the key addition, the substitution layer and the diffusion layer.

Proposition 1.32. Let a, b be two selection patterns in \mathbb{F}_2^{nm} and let k be a round key. The correlations of the approximation (a, b) over each step of the round function are given by

$$C_{\alpha_k}(a, b) = \delta_{a,b}(-1)^{\langle a, k \rangle}, \quad C_\sigma(a, b) = \prod_{i=0}^{m-1} C_{S_i}(a_i, b_i), \quad C_\pi(a, b) = \delta_{a, \pi^\top(b)}.$$

By analogy with differential trails, let us introduce the concept of linear trails. Even if the applications of linear trails are similar to the ones of differential trails, these two concepts are by nature very different.

Definition 1.33 (Linear Trail). An r -round *linear trail* is a family $\mathcal{T} = (a^{[i]})_{i \leq r}$ of $r + 1$ selection patterns. The *correlation contribution* of \mathcal{T} is defined to be

$$C(\mathcal{T}) = \prod_{i=0}^{r-1} C_{\pi\sigma}(a^{[i]}, a^{[i+1]}).$$

When considering the fixed-key correlation of an r -round approximation or the average of these correlations, the correlation contribution of a linear trail is just an intermediate variable. Unlike differential trails, a linear trail does not have a concrete meaning. Indeed, a pair can follow a differential trail but it is meaningless to say that the messages (or worse one message) follow a linear trail. An approximation does not consider the messages individually but the whole encryption function.

Definition 1.34 (Active S-Box). Let $\mathcal{T} = (a^{[i]})_{i \leq r}$ be a linear trail with a nonzero correlation contribution. The S-box S_j is said to be *active* at round i if the pattern $a_j^{[i]}$ is nonzero, otherwise S_j is *inactive*.

Remark 1.35. Generally, we say that S_j is active when b_j is nonzero. However, these two definitions are equivalent when considering bijective S-boxes, as ensured by the following proposition.

Proposition 1.36. Let $\mathcal{T} = (a^{[i]})_{i \leq r}$ be a nonzero correlation linear trail. Let $i < r$ and $j \leq m$ be nonnegative integers. Denote by $b^{[i]}$ the element $\pi^\top(a^{[i+1]})$. Then $a_j^{[i]} = 0_n$ if and only if $b_j^{[i]} = 0_n$. Consequently, Definition 1.33 can be restated as

$$C(\mathcal{T}) = \prod_{i,j \mid a_j^{[i]} \neq 0} C_{S_j}(a_j^{[i]}, b_j^{[i]}) \quad \text{and} \quad LP(\mathcal{T}) = \prod_{i,j \mid a_j^{[i]} \neq 0} LP_{S_j}(a_j^{[i]}, b_j^{[i]}).$$

Proof. To simplify the notations, denote by a' and b' the patterns $a_j^{[i]}$ and $b_j^{[i]}$ and by S the S-box S_j . Since $C(\mathcal{T})$ is assumed to be nonzero, it must be the case that

$C_S(a', b')$ is nonzero. We contend that $C_S(a', 0_n) = \delta_{a', 0_n}$, where δ is the Kronecker delta. To prove this, first observe that $C_S(0_n, 0_n) = 1$. Now, assume that a' is nonzero. By definition, $\langle 0_n, S(x) \rangle = 0$ for any x in \mathbb{F}_2^n . Next, the cardinality of $\{x \in \mathbb{F}_2^n \mid \langle a, x \rangle = 0\}$ is equal to 2^{n-1} since any linear Boolean function is balanced. This proves that $C_S(a', 0_n) = 0$ whenever a' is nonzero. The same argument proves that $C_{S^{-1}}(b', 0_n) = \delta_{b', 0_n}$. Then, it is well-known that $C_{S^{-1}}(b', 0_n) = C_S(0_n, b')$ (see [39, Equation 7.30]) and thus $C_S(0_n, b') = \delta_{0_n, b'}$. It follows that $a' = 0_n$ if and only if $b' = 0_n$. Finally, Lemma 1.31 and Proposition 1.32 imply that

$$\begin{aligned} C_{\pi\sigma}(a^{[i]}, a^{[i+1]}) &= \sum_{c \in \mathbb{F}_2^{nm}} C_\sigma(a^{[i]}, c) \times C_\pi(c, a^{[i+1]}) = \sum_{c \in \mathbb{F}_2^{nm}} C_\sigma(a^{[i]}, c) \times \delta(c, b^{[i]}) \\ &= C_\sigma(a^{[i]}, b^{[i]}) = \prod_{j=0}^{m-1} C_{S_j}(a_j^{[i]}, b_j^{[i]}). \end{aligned}$$

The result follows. ■

Let us now present the result relating linear trails and r -round linear approximations. The following proposition should be compared to Proposition 1.19, which is its counterpart about differential cryptanalysis.

Proposition 1.37. Let (a, b) be an r -round approximation and let K be a cipher key. Denote by $E_{a,b}$ the set of all trails $(a^{[i]})_{i \leq r}$ such that $a^{[0]} = a$, $a^{[r]} = b$. Given a trail \mathcal{T} in $E_{a,b}$, denote by $\langle \mathcal{T}, K \rangle$ the element $\sum_{i=0}^r \langle a^{[i]}, k^{[i]} \rangle$ of \mathbb{F}_2 . Then the fixed-key correlation of (a, b) is given by

$$C_{E_K}(a, b) = \sum_{\mathcal{T} \in E_{a,b}} (-1)^{\langle \mathcal{T}, K \rangle} C(\mathcal{T}).$$

In contrast with differential, the correlation of an r -round approximation is a *signed* sum of the correlation contributions of its associated linear trails. When the high absolute correlation trails are added with the same sign, the amplitude of the whole correlation will be higher. In this case, we speak of *constructive interference*. Otherwise, when these trails have different signs, the whole correlation can be close or even equal to zero and we speak of *destructive interference*. This result explains the strange distribution of the correlations in Figure 1.9 of Example 1.30.

Definition 1.38 (ELP). The *expected linear potential* of an r -round approximation (a, b) , denoted by $\text{ELP}(a, b)$, is the average fixed-key correlation of (a, b) over the associated long-key cipher.

Theorem 1.39. Let (a, b) be an r -round approximation and denote by $E_{a,b}$ the set of all trails $(a^{[i]})_{i \leq r}$ such that $a^{[0]} = a$, $a^{[r]} = b$. The expected linear potential of (a, b) is given by

$$\text{ELP}(a, b) = \sum_{\mathcal{T} \in E_{a,b}} \text{LP}(\mathcal{T}).$$

This time, the sum consists of nonnegative terms and thus there is no destructive interference. The expected linear potential is a powerful indicator of the cipher's

LP	#Trails	LP	#Trails	LP	#Trails
1.60×2^{-4}	1	1.80×2^{-21}	1	1.00×2^{-28}	2
1.27×2^{-10}	2	1.13×2^{-21}	2	1.13×2^{-29}	15
1.42×2^{-11}	4	1.27×2^{-22}	2	1.27×2^{-30}	21
1.00×2^{-16}	1	1.42×2^{-23}	1	1.00×2^{-32}	15
1.27×2^{-18}	1	1.00×2^{-24}	5	1.12×2^{-33}	44
1.42×2^{-19}	1	1.13×2^{-25}	3	1.00×2^{-36}	55
1.60×2^{-20}	2	1.27×2^{-26}	15		
1.00×2^{-20}	4	1.42×2^{-27}	7		

Figure 1.11: All linear potentials of the linear trails associated with the 4-round approximation (0020, 0020).

security against linear cryptanalysis. Nonetheless, it must be kept in mind that the actual correlation is highly key-dependent as established by Proposition 1.37. Finally, the expected linear potential has the same downsides as the expected differential probability. This value is generally impossible to compute precisely and one should make the *hypothesis of stochastic equivalence* [51] to relate this notion with the cipher's security.

Example 1.40. Using the preceding theory of linear approximations, we now study the 4-round linear approximation (a, b) with $a = b = 0020$ introduced in Example 1.30. With an exhaustive search, we found that there are exactly 204 linear trails associated with the linear approximation (a, b) . In Figure 1.11, we gather the trails according to their linear potentials. The trail which has the best linear potential is simply the trail $\mathcal{T} = (a^{[i]})_{i \leq 4}$ where $a^{[i]} = 0020$. Since the diffusion layer is a bit permutation, its transpose is equal to its inverse, that is $\pi^\top = \pi^{-1}$. It is then easily seen that $b^{[i]} = 0020$ for each $i < r$. By virtue of Proposition 1.36, the correlation of \mathcal{T} can be computed using the correlation matrix of S , given in Figure 1.12. Thus,

$$C(\mathcal{T}) = \prod_{i,j \mid a_j^{[i]} \neq 0} C_S(a_j^{[i]}, b_j^{[i]}) = C_S(2, 2)^4 = \left(-\frac{12}{16}\right)^4 = \left(\frac{9}{16}\right)^2.$$

The linear potential of \mathcal{T} is then $(\frac{9}{16})^4 \approx 1.60 \times 2^{-4}$. According to Theorem 1.39, the expected linear potential of the approximation (a, b) can be computed as follows:

$$\text{ELP}(a, b) = \sum_{\mathcal{T} \in E_{a,b}} \text{LP}(\mathcal{T}) = 1.63 \times 2^{-4}.$$

As we can see, this value is dominated by the linear potential of the best trail. In Example 1.30 we have found that the average fixed-key linear potential including the key schedule is equal to 1.69×2^{-4} , so is very close to the expected linear potential. This shows that the linear potential of one high potential linear trail can well approximate the average potential of the associated approximation.

We will not explain here the complex and surprising distribution of the fixed-key correlations (a, b) illustrated in Figure 1.9. Indeed, these correlations depend on

$2^4 \times C_S(a, b)$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16
1	.	.	4	-4	-8	.	-4	-4	.	-8	4	4	.	.	4	-4
2	.	.	-12	4	.	.	-4	-4	-4	-4	.	.	4	4	.	.
3	.	8	.	.	.	8	.	-4	-4	-4	4	-4	4	4	4	4
4	.	-8	.	.	8	8	.	.	8	.	.
5	.	.	-4	-4	.	8	4	-4	.	.	-4	-4	.	-8	4	-4
6	.	.	-4	4	.	.	-4	4	4	-4	.	.	-12	-4	.	.
7	8	.	-8	4	4	4	4	-4	4	-4	4
8	.	.	4	4	.	.	-4	-4	.	.	-12	4	.	.	-4	-4
9	.	.	.	8	-8	8	8	.
A	12	-4	-4	-4	4	4	4	4	4
B	.	-8	4	4	.	.	4	-4	-4	-4	.	-8	-4	4	.	.
C	.	-8	-4	-4	-8	.	4	4	.	.	-4	4	.	.	-4	4
D	8	.	8	8	.	-8
E	.	.	.	8	.	.	8	.	4	-4	4	4	4	-4	-4	-4
F	.	.	-4	-4	.	-8	4	-4	4	4	.	.	-4	4	.	-8

$2^4 \times LP_S(a, b)$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16
1	.	.	1	1	4	.	1	1	.	4	1	1	.	.	1	1
2	.	.	9	1	.	.	1	1	1	1	.	.	1	1	.	.
3	.	4	4	.	1	1	1	1	1	1	1	1
4	.	4	.	.	4	4	.	.	4	.
5	.	.	1	1	.	4	1	1	.	.	1	1	.	4	1	1
6	.	.	1	1	.	.	1	1	1	1	.	.	9	1	.	.
7	4	.	4	1	1	1	1	1	1	1	1
8	.	.	1	1	.	.	1	1	.	.	9	1	.	.	1	1
9	.	.	.	4	4	4	4	.
A	9	1	1	1	1	1	1	1	1
B	.	4	1	1	.	.	1	1	1	1	.	4	1	1	.	.
C	.	4	1	1	4	.	1	1	.	.	1	1	.	.	1	1
D	4	.	4	4	.	4
E	.	.	.	4	.	.	4	.	1	1	1	1	1	1	1	1
F	.	.	1	1	.	4	1	1	1	1	1	.	.	1	1	4

 Figure 1.12: The correlation and linear potential matrices of S .

the interaction between the expanded key and the 204 linear trails associated with (a, b) . However, we study another approximation whose linear potential distribution is even more surprising but simpler to explain.

In the remainder of this example, consider the 4-round linear approximation (a, b) where $a = b = 0400$. It can be proven via an exhaustive search that the four linear trails $\mathcal{T}_1, \dots, \mathcal{T}_4$ presented in Figure 1.13 are the only trails associated with (a, b) . At the left of Figure 1.14 is illustrated the distribution of the fixed-key correlations of (a, b) , including the key schedule. Therefore, the correlation $C_{E_K^{(4)}}(0400, 0400)$ respectively is equal to 0, 2^{-3} and 2^{-2} for a proportion of $\frac{3}{8}$, $\frac{1}{2}$ and $\frac{1}{8}$ of the cipher keys. According to Proposition 1.37, the fixed-key correlation of (a, b) is a signed combination of the correlations of the \mathcal{T}_i , that is

$$C_{E_K}(0400, 0400) = \sum_{i=1}^4 (-1)^{\langle \mathcal{T}_i, K \rangle} C(\mathcal{T}_i).$$

Seeing the trails \mathcal{T}_i as elements of $(\mathbb{F}_2^{16})^5$, we have

$$\begin{aligned} \mathcal{T}_1 &= (0400, 0400, 0400, 0400, 0400), & \mathcal{T}_3 &= (0400, 0004, 0100, 0400, 0400), \\ \mathcal{T}_2 &= (0400, 0400, 0004, 0100, 0400), & \mathcal{T}_4 &= (0400, 0004, 0001, 0100, 0400). \end{aligned}$$

Clearly, these four trails are linearly independent and thus, all the possible signed sums are equally likely when considering the long-key cipher. These sums are given at the right of Figure 1.14. This explains the distribution of the correlations of (a, b) . ▀

1.5. Security Evaluation of SPN and Strong Primitives

As explained in Sections 1.3.2 and 1.4.2, the effectiveness of a differential is assessed by its expected differential probability and the effectiveness of a linear approxima-

	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4
$a^{[0]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)
$b^{[0]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, 1, 0, 0)
$a^{[1]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 0, 0, 4)	(0, 0, 0, 4)
$b^{[1]}$	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, 0, 0, 4)	(0, 0, 0, 1)
$a^{[2]}$	(0, 4, 0, 0)	(0, 0, 0, 4)	(0, 1, 0, 0)	(0, 0, 0, 1)
$b^{[2]}$	(0, 4, 0, 0)	(0, 0, 0, 4)	(0, 4, 0, 0)	(0, 0, 0, 4)
$a^{[3]}$	(0, 4, 0, 0)	(0, 1, 0, 0)	(0, 4, 0, 0)	(0, 1, 0, 0)
$b^{[3]}$	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)	(0, 4, 0, 0)
C	$8^4/16^4$	$8^4/16^4$	$8^4/16^4$	$8^4/16^4$
=	2^{-4}	2^{-4}	2^{-4}	2^{-4}

Figure 1.13: The trails associated with the 4-round approximation (0400, 0400).

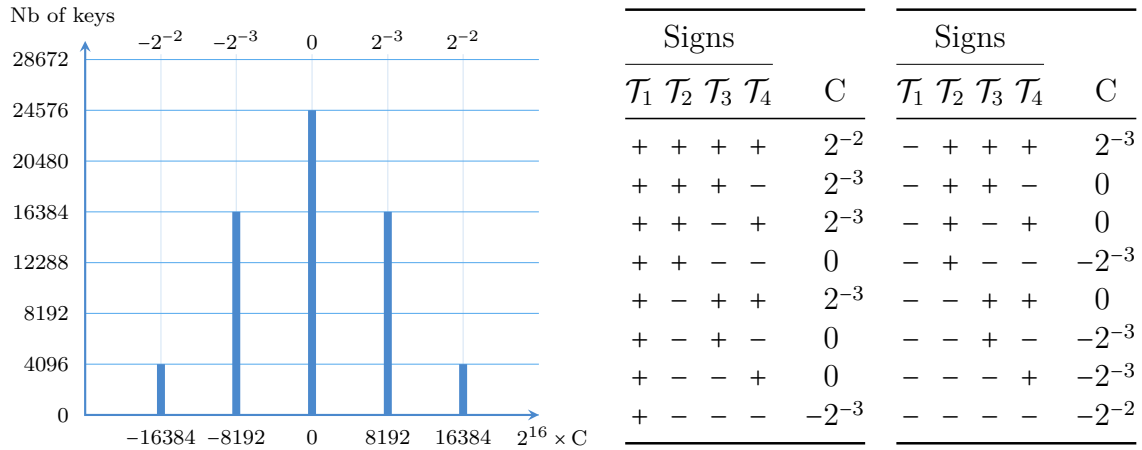


Figure 1.14: Correlations of the 4-round approximation (0400, 0400).

tion by its expected linear potential. Naturally, a block cipher is resistant against differential and linear cryptanalysis when there exists no effective differential and approximation over the $(r - 1)$ -round encryption function. Equivalently, the cipher is secure when the maximum expected differential probability (MEDP) or linear potential (MELP) is low enough, namely if the corresponding attacks require more plaintext/ciphertext pairs than the block size allows.

According to Theorem 1.26, the expected probability of a given differential is hard to compute and Theorem 1.39 establishes the same result for linear cryptanalysis. Therefore, computing the MEDP and MELP are even harder. Nonetheless, we have seen that the expected probability of a differential can be approximated by the probability of its best trail and the same holds for a linear approximation. Relying on these facts, Kanda et al. [57] introduced four measures of security which can be divided into two categories according to the security they imply.

- The *provable security* of a cipher is evaluated by two measures called *precise* and *theoretical*. The precise measure gives the MEDP (resp. MELP) whereas the theoretical measure only gives an upper-bound of this value.
- The *practical security* of a cipher is assessed by two measures called *heuristic* and *practical*. The heuristic measure gives the maximum differential probability (resp. linear potential) of all trails while the practical measure upper-bounds this value.

Because computing even the heuristic measure can be a challenging problem, most of ciphers' security is assessed by the practical measure.

The standard strategy to design a secure cipher is to ensure that each differential or linear trail activates many S-boxes and that all the S-boxes of the cipher have good resistances against linear and differential cryptanalysis.

Definition 1.41. Let S be an n -bit S-box. The maximum differential probability, correlation and linear potential of S , denoted respectively by DP_S^{\max} , C_S^{\max} and LP_S^{\max} , are defined to be

$$\begin{aligned} DP_S^{\max} &= \max\{DP(a, b) \mid a \in (\mathbb{F}_2^n)^*, b \in \mathbb{F}_2^n\}, \\ C_S^{\max} &= \max\{C(a, b) \mid a \in \mathbb{F}_2^n, b \in (\mathbb{F}_2^n)^*\}, \\ LP_S^{\max} &= \max\{LP(a, b) \mid a \in \mathbb{F}_2^n, b \in (\mathbb{F}_2^n)^*\} = (C_S^{\max})^2. \end{aligned}$$

Remark 1.42. According to Propositions 1.24 and 1.36, these maximums can be searched only for a and b both nonzero.

1.5.1. Perfect S-Boxes

This section deals with the resistance of S-boxes with respect to differential and linear cryptanalysis. The theory covering this topic considers a larger class function. Following [33], an (n, m) -function is defined to be a mapping from \mathbb{F}_2^n to \mathbb{F}_2^m . There are also known as vectorial Boolean function. Thus, according to our conventions, an n -bit S-box is a bijective (n, n) -function.

1.5.1.a. Almost Perfect Nonlinear Functions

Let F be an (n, m) -function. For each a in \mathbb{F}_2^n and b in \mathbb{F}_2^m , denote by $\delta_F(a, b)$ the number of solutions to the equation $F(x) + F(x + a) = b$. These values are clearly related to the differential probabilities of F by the formula

$$2^n \times \text{DP}_F(a, b) = \delta_F(a, b). \quad (1.8)$$

Remark 1.43. In Section 1.3.2, we have only defined the differential probabilities for n -bit S-boxes. This notion can naturally be extended to (n, m) -functions, precisely using Equation (1.8).

Definition 1.44 (δ -uniform function [80]). Let δ be an integer. An (n, m) -function F is said to be δ -uniform if for all nonzero a in \mathbb{F}_2^n and all b in \mathbb{F}_2^m , it holds that $\delta_F(a, b) \leq \delta$. Equivalently, F is δ -uniform if

$$2^n \times \text{DP}_F^{\max} \leq \delta.$$

Let a be any nonzero element of \mathbb{F}_2^n . Obviously, for each x in \mathbb{F}_2^n , there exists a unique b in \mathbb{F}_2^m such that $F(x) + F(x + a) = b$. Therefore

$$\sum_{b \in \mathbb{F}_2^m} \delta_F(a, b) = 2^n.$$

In order to minimize the maximum of the $\delta_F(a, b)$ with b in \mathbb{F}_2^m , their sum must be uniformly distributed over the all these values, proving the bound

$$\max_{a \in (\mathbb{F}_2^n)^*, b \in \mathbb{F}_2^m} \delta_F(a, b) \geq 2^{n-m}. \quad (1.9)$$

Consequently, any (n, m) -function is at least 2^{n-m} -uniform. An (n, m) -function which meets this bound with equality is called *perfect nonlinear* [79]. Referring to Equation (1.8), we see that F is perfect nonlinear if and only if DP_F^{\max} is minimal. Thus, perfect nonlinear functions provide optimal resistance against differential cryptanalysis.

It worthwhile to note that if x is solution to the equation $F(x) + F(x + a) = b$, then so is $x + a$. Thus, when a is nonzero, $\delta_F(a, b)$ is even (this result remains true when $a = 0$, but requires another argument). It follows that

$$\max_{a \in (\mathbb{F}_2^n)^*, b \in \mathbb{F}_2^m} \delta_F(a, b) \geq 2. \quad (1.10)$$

Assume that F is an (n, n) -function. According to Equation (1.9), F must be 1-uniform to be perfect nonlinear. However, Equation (1.10) ensures that any function is at least 2-uniform. This proves that there does not exist perfect nonlinear (n, n) -functions. Because (n, n) -functions are widely used in cryptography, particularly n -bit S-boxes, Nyberg introduced the following definition in [82].

Definition 1.45 (APN function). Any 2-uniform (n, n) -function is said to be *almost perfect nonlinear* (APN).

Remark 1.46. Using the notation introduced at the beginning of Section 1.5.1, an (n, n) -function F is almost perfect nonlinear if and only if $\text{DP}_F^{\max} = 2^{-(n-1)}$. Additionally, we may stress that the term *almost* is misleading because APN functions are optimal, as noted in [33].

1.5.1.b. Almost Bent Functions

Let F be an (n, m) -function. For each a in \mathbb{F}_2^n and b in \mathbb{F}_2^m , denote by $\lambda_F(a, b)$ the integer

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

The family Λ_F consisting of all the values $\lambda_F(a, b)$ is called the *Walsh spectrum* of F . Up to scaling, the Walsh spectrum of F is equivalent to its correlation matrix. More precisely, for all selection patterns a and b in \mathbb{F}_2^n and \mathbb{F}_2^m respectively, we have

$$\lambda_F(a, b) = 2^n \times C_F(a, b).$$

Indeed, denoting by E_i the set of all elements x in \mathbb{F}_2^n such that the sum $\langle a, x \rangle + \langle b, F(x) \rangle$ is equal to i in \mathbb{F}_2 , we have

$$\lambda_F(a, b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle a, x \rangle + \langle b, F(x) \rangle} = \sum_{x \in E_0} (-1)^0 + \sum_{x \in E_1} (-1)^1 = \#E_0 - \#E_1. \quad (1.11)$$

Clearly, the set E_1 is the relative complement of E_0 in \mathbb{F}_2^n and thus $\#E_1 = 2^n - \#E_0$. Replacing in (1.11), we obtain

$$\lambda_F(a, b) = 2\#E_0 - 2^n. \quad (1.12)$$

Next, observe that the set E_0 is equal to $\{x \in \mathbb{F}_2^n \mid \langle a, x \rangle = \langle b, F(x) \rangle\}$. The result then follows from the definition of $C_F(a, b)$.

The nonlinearity of an (n, m) -function F was introduced by Nyberg in [81] and is defined to be

$$\mathcal{NL}(F) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n, b \in (\mathbb{F}_2^m)^*} |\lambda_F(a, b)| = 2^{n-1} (1 - C_F^{\max}).$$

Referring to (1.12), it is easily seen that each value $\lambda_F(a, b)$ in the Walsh spectrum of F is even. Therefore, $\mathcal{NL}(F)$ is an integer. In addition, it can be proven that the nonlinearity of F is upper-bounded as follows:

$$\mathcal{NL}(F) \leq 2^{n-1} - 2^{\frac{n}{2}-1}.$$

This inequality is known as the *covering radius bound*. An (n, m) -function meeting this bound with equality is said to be *bent*. Since the right side of this inequality is an integer if and only if n is even, bent functions cannot exist when n is odd. Using the maximum absolute correlation of F , the covering radius bound can be equivalently rewritten as

$$2^n \times C_F^{\max} \geq 2^{\frac{n}{2}}.$$

Thus, the function F is bent if and only if C_F^{\max} is minimal. This restatement stresses that bent functions are exactly the (n, m) -functions which are the most resistant to linear cryptanalysis. However, Nyberg proved in [79] that such functions exist only if $n \geq 2m$. Therefore, n -bit S-boxes cannot be bent.

Remark 1.47. It turns out that these bent functions are exactly the perfect non-linear functions introduced in the previous section, as shown in [79]. Thus, bent functions are optimal with respect to differential and linear cryptanalysis.

Since the covering radius bound is not tight for every (n, m) -function, Chabaud and Vaudenay improved this inequality in [35, Theorem 4]. Their bound is now called the *Sidelnikov-Chabaud-Vaudenay bound* (shorten as *SCV bound*) because Sidelnikov had published earlier an equivalent result in [92]. For the particular case of (n, n) -functions, this bound gives

$$\mathcal{NL}(F) \leq 2^{n-1} - 2^{\frac{n-1}{2}}. \quad (1.13)$$

Definition 1.48 (AB function [35]). An (n, n) -function is said to be *Almost Bent* (AB) if its nonlinearity meets the bound (1.13) with equality.

Remark 1.49. Using the notations introduced at the beginning of Section 1.5.1, the SCV bound (1.13) can be restated in the following equivalent ways

$$2^n \times C_F^{\max} \geq 2^{\frac{n+1}{2}}, \quad C_F^{\max} \geq 2^{\frac{n-1}{2}}, \quad \text{LP}_F^{\max} \geq 2^{-(n-1)}. \quad (1.14)$$

Again, the term *almost* in the previous definition is misleading because AB functions are optimal.

When n is even, the right side of the SCV bound (1.13) is not an integer. Therefore, almost bent functions exist only if n is odd. To conclude, we should recall the result of Chabaud and Vaudenay [35] linking AB and APN functions.

Theorem 1.50. Any almost bent function is almost perfect nonlinear.

1.5.1.c. Known AB and APN Permutations

By a *power function*, we mean an (n, n) -function F which has the form $F(x) = x^d$ when we identify the space \mathbb{F}_2^n with the finite field \mathbb{F}_{2^n} . Several almost perfect nonlinear power functions are known when n is even, but they cannot be used as S-boxes in a substitution permutation network because they are not bijective. Indeed, it is proven in [33, Section 2.1.3] that an APN power function is a permutation of \mathbb{F}_2^n if and only if n is odd.

In Figure 1.15, we enumerate the known almost bent power functions of \mathbb{F}_{2^n} with n odd. According to Theorem 1.50, these functions are also almost perfect nonlinear, which proves in particular that they are permutations of \mathbb{F}_{2^n} . In other words, these power permutations are S-boxes with optimal resistance against differential and linear cryptanalysis.

Function	Exponent d	Conditions	Proven in
Gold functions	$2^i + 1$	$\gcd(i, n) = 1$	[45, 80]
Kasami functions	$2^{2^i} - 2^i + 1$	$\gcd(i, n) = 1$	[58]
Welch function	$2^t + 3$	$n = 2t + 1$	[28, 29, 44]
Niho function	$2^t + 2^{\frac{t}{2}} - 1, \quad t \text{ even}$ $2^t + 2^{\frac{3t+1}{2}} - 1, \quad t \text{ odd}$	$n = 2t + 1$	[43, 53]

Figure 1.15: Known AB (and APN) power permutations $x \mapsto x^d$ over \mathbb{F}_{2^n} with n odd.

Function	Exponent d	Conditions	Proven in
Gold functions	$2^i + 1$	$\gcd(i, n) = 1, n \equiv 2 \pmod{4}$	[45, 80]
Kasami functions	$2^{2^i} - 2^i + 1$	$\gcd(i, n) = 1, n \equiv 2 \pmod{4}$	[58]
Inverse function	$2^n - 2$	–	[66]

Figure 1.16: Known power permutations $F : x \mapsto x^d$ over \mathbb{F}_{2^n} with n even such that F is 4-uniform and $2^n \times C_F^{\max} = 2^{\frac{n+2}{2}}$.

When n is even, we already know that AB functions, and hence AB permutations do not exist. The existence of APN permutations with n even has been a long-standing open question. However, in 2009 Dillon et al. [18] exhibited an APN permutation on 6 bits. So far, it is the only APN permutation of \mathbb{F}_2^n known when n is even, up to equivalence.

Therefore, when n is even we generally use n -bit S-boxes F with parameters close to AB and APN functions, namely 4-uniform permutations such that $2^n \times C_F^{\max} = 2^{\frac{n+2}{2}}$. All the 4-bit S-boxes have been classified in [72] and we now know that these values are optimal for $n = 4$. Figure 1.16 gives the known power permutations with these parameters. Observe that if n is a multiple of 4, the only known power permutation reaching these values is the inversion. This explains why the S-box of the AES is affine-equivalent to the inversion in \mathbb{F}_{2^8} .

1.5.2. Branch Number of the Diffusion Layer

In the previous section, we have describe how to design a substitution layer resistant to differential and linear cryptanalysis. Now we focus on the diffusion layer of an SPN. By virtue of Propositions 1.17 and 1.32, the diffusion layer alone cannot provide any resistance to these attacks since it is linear. However, the diffusion layer can enhance the resistance provided by the substitution layer. The wide trail strategy is a design principle of block ciphers introduced by Daemen and Rijmen in [39]. Following this strategy, the diffusion layer should ensure that any linear of differential trail activate a large number of S-boxes.

The *Hamming weight* of an element x of \mathbb{F}_2^n , denoted by $w(x)$, is the number of nonzero components of x , that is $w(x) = \#\{0 \leq i < n \mid x_i \neq 0\}$. By analogy, we define the *bundle weight* of an element x in $(\mathbb{F}_2^n)^m$ to be the number $w_n(x)$ of nonzero

bundles of x , so $w_n(x) = \#\{0 \leq i < m \mid x_i \neq 0_n\}$. The following definition characterizes the efficiency of the diffusion provided by the diffusion layer with respect to differential and linear cryptanalysis.

Definition 1.51 (Branch Number). Let $\lambda : (\mathbb{F}_2^n)^m \rightarrow (\mathbb{F}_2^n)^m$ be a \mathbb{F}_2 -linear mapping. The *differential branch number* \mathcal{B}_D and the *linear branch number* \mathcal{B}_L of λ are defined by

$$\begin{aligned}\mathcal{B}_D(\lambda) &= \min\{w_n(x) + w_n(\lambda(x)) \mid x \in (\mathbb{F}_2^n)^m, x \neq 0\}, \\ \mathcal{B}_L(\lambda) &= \min\{w_n(x) + w_n(\lambda^\top(x)) \mid x \in (\mathbb{F}_2^n)^m, x \neq 0\}.\end{aligned}$$

Clearly, the differential and linear branch numbers of λ are upper-bounded by $m + 1$. The linear mapping λ whose branch numbers meet this bound with equality is said to be *MDS* or a *perfect diffusion layer*. In fact, perfect diffusion layers can be constructed from MDS codes, the reader can refer to [39, Sections 2.2 and 9.6].

Let us consider a generic SPN. The branch numbers of the diffusion layer can be used to derive important bounds on the maximal differential probability or linear potential of a trail. Therefore, the cipher's security can easily be assessed using the practical measure. The following theorem comes from [39, Theorem 9.3.1].

Theorem 1.52. Consider a generic SPN and denote by S_0, \dots, S_{m-1} its n -bit S-boxes and by $\pi : \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$ its diffusion layer. The maximum differential probability and linear potential of any 2-round trail are respectively upper-bounded by

$$\left(\max_{i < m} \text{DP}_{S_i}^{\max}\right)^{\mathcal{B}_D(\pi)} \quad \text{and} \quad \left(\max_{i < m} \text{LP}_{S_i}^{\max}\right)^{\mathcal{B}_L(\pi)}.$$

Security Evaluation of SPN

Differential [13] and linear [74] cryptanalysis are considered as the most important attacks against block ciphers [64]. As mentioned in [41], any new cipher should at least be accompanied by a detailed analysis of its strength against these two attacks. We have seen in Chapter 1 that security of a cipher is assessed by the maximum expected differential probability (MEDP) or linear potential (MELP). When these values are low enough, the cipher is *provably secure* [57]. Nevertheless, computing the MEDP and MELP or even finding a useful upper bound remains a challenging open problem and the common proofs of security focus only on differential and linear trails. A cipher is then said to be *practically secure* when the maximum differential probability or linear potential of all trails gives rise to an ineffective cryptanalysis. Finally, it should be stressed that all these security measures tacitly assume that the round keys are independent. The cryptographer then assumes that these theoretical measures reflect the actual security when the round keys are fixed and derived from a key schedule. This hypothesis, called *stochastic equivalence* [67], seems to hold for almost all secure ciphers.

To prevent differential and linear cryptanalysis, the SPN designer must first choose S-boxes providing high resistance against both these attacks. These choices define the substitution layer of the cipher. Concerning the diffusion layer, two main families stand out. On one hand, the diffusion of the cipher can be done using a bit permutation. Even if bit permutations do not provide the best security, they are generally chosen for efficiency purposes. Indeed, in the last few years, many lightweight block ciphers using bit permutations have been suggested [17, 36, 95]. A recent survey of lightweight block ciphers can be found in [15]. On the other hand, the diffusion layer can involve a more complicated linear mapping defined for example as a matrix product over finite fields. Such mappings are generally more computationally expensive but they also provide high diffusion, ensuring that every trail activates a minimum number of S-boxes. Relying on this property, the designer can derive bounds on the maximum differential probability and linear potential of any trail and simply prove the practical security of its cipher.

However, the bounds obtained for an SPN which uses bit permutations may not suffice to prove its security. In fact, bit permutations have the smallest branch number possible among all linear permutations and the cipher security is hard to establish without a close analysis. The same observation may apply for backdoor ciphers since the mathematical structure of the backdoor strongly reduces the choice

of the cipher's primitives. Thus, the usual strategies to thwart differential and linear cryptanalysis may no longer be useful. This motivates alternative methods to prove the security with respect to these attacks.

In this chapter, we describe a fully automatic algorithm finding an optimal differential or linear trail in an SPN. Our contribution was presented in [8]. The first algorithm finding optimal trails was introduced by Matsui in [76] for Feistel ciphers. Running his algorithm several times on the DES, Matsui found a permutation of the S-boxes making the DES stronger against differential and linear cryptanalysis. The algorithm complexity remaining too high for the cipher FEAL, two successive improvements have been proposed in [87] then [3]. Although an adaptation of Matsui's algorithm is straightforward for SPN, the block size (from 64 to 128 bits) of modern ciphers makes it computationally infeasible. This fact was also highlighted by Collard et al. [37] who then proposed a few improvements to use this algorithm on the cipher SERPENT. In addition, it should be mentioned that another variation was exposed by Ali and Heys in [1]. They gave up finding an optimal trail to reduce the complexity. On the other side, their algorithm cannot prove the cipher practical security, but may still help the cryptanalyst to perform a differential or linear cryptanalysis. Our algorithm is an adaptation of [3, 76, 87] for SPN. We introduce several optimizations paying special attention to ciphers which have a bit permutation as diffusion layer.

After a brief summary of differential and linear cryptanalysis, the next section exposes a straightforward adaptation of Matsui's algorithm to compute optimal trails in substitution-permutation networks. An example of execution is then given in Section 2.2. All our optimizations are explained intuitively in this example and then formalized in Section 2.3. Finally, we present our results and close this chapter in Section 2.4.

2.1. Search for an Optimal trail

Throughout this section, we consider a generic r -round substitution-permutation network $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$ such that for each cipher key K ,

$$E_K = F_{k[r-1]} \circ \cdots \circ F_{k[0]} \quad \text{with} \quad F_{k[i]} = \pi \circ \sigma \circ \alpha_{k[i]} .$$

As explained in Section 1.3.2 and 1.4.2, the last round includes and ends with a diffusion layer because the linear approximations or differentials used in an attack have less rounds than the whole cipher. We denote by S_0, \dots, S_{m-1} the n -bit S-boxes of the substitution layer. Recall that the differential probability and linear potential matrices of an S-box S are defined for all a, b in \mathbb{F}_2^n by the formulae

$$\begin{aligned} \text{DP}_S(a, b) &= 2^{-n} \times \#\{x \in \mathbb{F}_2^n \mid S(x) + S(x + a) = b\} , \\ \text{LP}_S(a, b) &= \left(2^{-(n-1)} \times \#\{x \in \mathbb{F}_2^n \mid \langle a, x \rangle = \langle b, S(x) \rangle\} - 1 \right)^2 . \end{aligned}$$

The maximum differential probability and linear potential of S are then defined to be

$$\text{DP}_S^{\max} = \max\{\text{DP}(a, b) \mid a, b \in (\mathbb{F}_2^n)^*\}, \quad \text{LP}_S^{\max} = \max\{\text{LP}(a, b) \mid a, b \in (\mathbb{F}_2^n)^*\}.$$

According to Definitions 1.18 and 1.33, an r -round differential or linear trail is a family $(a^{[i]})_{i \leq r}$ of $r + 1$ patterns in \mathbb{F}_2^{nm} . In this chapter, it is however more convenient to specify for each round the input and output patterns of the substitution layer. Therefore, we equivalently define a differential or linear trail \mathcal{T} to be a family $((a^{[i]}, b^{[i]}))_{i < r}$ of r pairs of input/output patterns such that for each $i < r - 1$,

$$a^{[i+1]} = \begin{cases} \pi(b^{[i]}) & \text{for differential trails,} \\ (\pi^\top)^{-1}(b^{[i]}) & \text{for linear trails.} \end{cases}$$

The equivalence between these two definitions follows from Propositions 1.17 and 1.31. Additionally, if the diffusion layer π is a bit permutation, it can be proven that $(\pi^\top)^{-1} = \pi$. In this case, the same structure can be seen as a differential or linear trail. Next, the differential probability and the linear potential of the trail \mathcal{T} are given by

$$\begin{aligned} \text{DP}(\mathcal{T}) &= \prod_{i=0}^{r-1} \text{DP}_\sigma(a^{[i]}, b^{[i]}) = \prod_{i,j \mid a_j^{[i]} \neq 0} \text{DP}_{S_j}(a_j^{[i]}, b_j^{[i]}), \\ \text{LP}(\mathcal{T}) &= \prod_{i=0}^{r-1} \text{LP}_\sigma(a^{[i]}, b^{[i]}) = \prod_{i,j \mid a_j^{[i]} \neq 0} \text{LP}_{S_j}(a_j^{[i]}, b_j^{[i]}), \end{aligned}$$

as established by Propositions 1.24 and 1.36. In other words, the differential probability of a trail is obtained by multiplying the differential probabilities of its active S-boxes.

Definition 2.1 (Optimal Trail). An r -round differential trail which has maximum probability among all r -round trails is said to be *optimal*. Naturally, we define an optimal linear trail to be a trail which has maximal linear potential. In this case, its probability (or potential) is denoted by $p_o^{(r)}$.

It is worth noting that there may exist more than one optimal trail. In the context of our search algorithm, a *candidate* for an input pattern a in \mathbb{F}_2^{nm} is an output pattern b such that $\text{DP}_\sigma(a, b)$ is nonzero. Of course, if we search an optimal linear trail, this condition becomes $\text{LP}_\sigma(a, b) \neq 0$. If $\mathcal{T} = ((a^{[i]}, b^{[i]}))_{i < r}$ is an r -round trail, we denote by $\mathcal{T}^{[i,j]}$ the sub-trail $((a^{[k]}, b^{[k]}))_{i \leq k \leq j}$. Finally, we will need the following definition.

Definition 2.2 (Trail Extension). Let r_1 and r_2 be integers such that $0 \leq r_1 \leq r_2$. Let \mathcal{T}_1 and \mathcal{T}_2 be r_1 and r_2 -round trails respectively. The trail \mathcal{T}_2 *extends* \mathcal{T}_1 if $\mathcal{T}_2^{[0, r_1-1]} = \mathcal{T}_1$. In this case, $\mathcal{T}_2 = \mathcal{T}_1 \parallel \mathcal{T}_2^{[r_1, r_2-1]}$.

2.1.1. General Principle

Let us now present a straightforward adaptation of Matsui's search algorithm for substitution-permutation networks. First, we explain how this algorithm computes an optimal differential trail. Then we will detail the changes that need to be made to compute an optimal linear trail.

Let us denote R the actual number of rounds of the SPN. The algorithm presented in this chapter computes an optimal R -round trail without requiring any *a priori* knowledge. This algorithm is based on another search algorithm called **OptTrailEst** which takes as arguments:

- an integer $r \geq 2$ representing the current number of rounds,
- the probabilities $(p_o^{(i)})_{1 \leq i < r}$ of optimal i -round trails,
- an estimation $p_e^{(r)}$ of the probability $p_o^{(r)}$ of the optimal trail searched,

and returns an optimal r -round trail denoted by $\mathcal{T}_o^{(r)}$. The knowledge of $(p_o^{(i)})_{1 \leq i < r}$ and $p_e^{(r)}$ speeds up the search. Next, an automatic management of the estimation $p_e^{(r)}$ will be proposed in Section 2.3.5 yielding the algorithm **OptTrail**. To summarize, the search algorithm **OptTrail** takes only r and $(p_o^{(i)})_{1 \leq i < r}$ as inputs and still outputs an optimal r -round trail.

Let us now explain how the algorithm **OptTrail** can be used to compute an optimal R -round trail from scratch. First, observe that $p_o^{(1)}$ can be easily computed (cf Remark 2.8). Then, compute

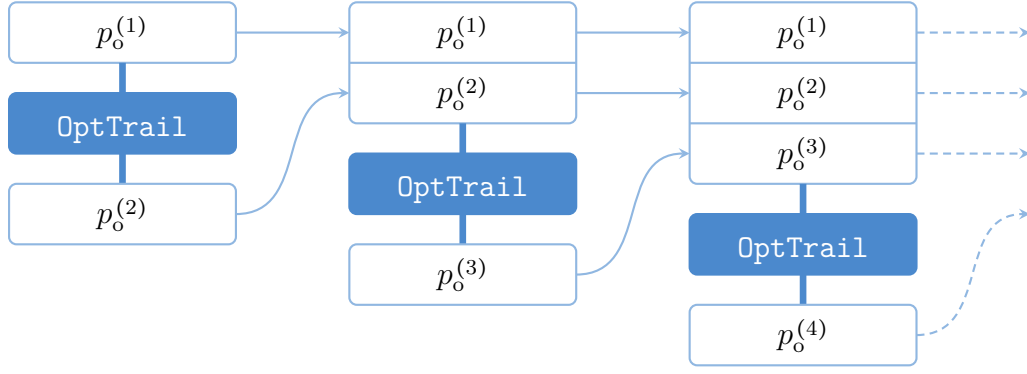
$$\mathcal{T}_o^{(r)} = \text{OptTrail}(r, (p_o^{(i)})_{1 \leq i < r}) \quad \text{and} \quad p_o^{(r)} = \text{DP}(\mathcal{T}_o^{(r)})$$

for r from 2 to R . The latter computation gives the desired result, as illustrated in Figure 2.1.

The rest of is dedicated to the algorithm **OptTrailEst** given in Figure 2.2. Let us explain how this algorithm works. First, suppose that the conditions on lines 8 and 16 are always true and that $p_e^{(r)}$ is equal to 0. Under this assumption, the algorithm runs implicitly through the tree of all r -round trails and saves one which has a maximum probability in the variable $\mathcal{T}_o^{(r)}$. Observe that the first and last rounds have a special treatment that speeds up the search. When the program reaches the function **Round**($s, \mathcal{T}^{(s-1)}, p^{(s-1)}$), the current trail is

$$\begin{aligned} \mathcal{T}^{(s-1)} &= ((a^{[0]}, b^{[0]}), \dots, (a^{[s-2]}, b^{[s-2]})) , \\ \text{DP}(\mathcal{T}^{(s-1)}) &= \prod_{i=0}^{s-2} \text{DP}_\sigma(a^{[i]}, b^{[i]}) = p^{(s-1)} . \end{aligned}$$

The input pattern $a^{[s-1]}$ for this round equals $\pi(b^{[s-2]})$. Then, for each candidates $b^{[s-1]}$ for $a^{[s-1]}$, the current trail $\mathcal{T}^{(s-1)}$ is extended by $(a^{[s-1]}, b^{[s-1]})$ and the search for the next round is called. Therefore, the program performs a depth-first search. When the algorithm reaches the function **LastRound**(), it is not hard to compute the output pattern $b^{[r-1]}$ maximizing the probability of the last round. The trail is then saved only if its probability is greater than the probability $p_e^{(r)}$ of the best trail $\mathcal{T}_o^{(r)}$ found up to this point. It remains to explain the conditions on lines 8 and 16.



Input. The number R of round of the cipher.

Output. An optimal R -round trail $\mathcal{T}_o^{(r)}$.

```

1  $p_o^{(1)} \leftarrow \max\{\text{DP}_\sigma(a, b) \mid a, b \in (\mathbb{F}_2^n)^m\}$ 
2 For  $r$  from 2 to  $R$  do
3    $\mathcal{T}_o^{(r)} \leftarrow \text{OptTrail}(r, (p_o^{(i)})_{1 \leq i < r})$ 
4    $p_o^{(r)} \leftarrow \text{DP}(\mathcal{T}_o^{(r)})$ 
5 Return  $\mathcal{T}_o^{(R)}$ 
    
```

Figure 2.1: Use of `OptTrail`.

Definition 2.3 (rank- s bound). Let \mathcal{T} be an s -round trail with $1 \leq s < r$. Its probability is said to be *less than the rank- s bound* if

$$\text{DP}(\mathcal{T}) < \frac{p_e^{(r)}}{p_o^{(r-s)}}.$$

This condition on the probability of the current trail allows to prune the search tree without missing an optimal trail. It can be rewritten as

$$\text{DP}(\mathcal{T}) \times p_o^{(r-s)} < p_e^{(r)}$$

and means that even if the trail is extended by an optimal $(r - s)$ -round trail, the probability of the whole trail would be less than $p_e^{(r)}$.

The significance of $p_e^{(r)}$ is now clear. If $p_e^{(r)} > p_o^{(r)}$, a trail expandable into an optimal r -round trail can be cut. Furthermore, no trail will be saved because of the condition on line 25. On the other hand, the closer $p_e^{(r)}$ is from $p_o^{(r)}$, the stronger is the pruning condition and the lower is the complexity of `OptTrailEst`.

Theorem 2.4. According to the results recalled in introduction of this section, the algorithm `OptTrailEst` can compute an optimal *linear* trail simply by replacing every `DP` by `LP` and every $\pi(\dots)$ by $(\pi^\top)^{-1}(\dots)$.

2.1.2. Proof of the Algorithm

Having explained the general principle of the algorithm, it remains now to prove the optimality of the trail returned.

Algorithm 1 – OptTrailEst($r, (p_o^{(i)})_{1 \leq i < r}, p_e^{(r)}$)

Input. The current number r of rounds ($r \geq 2$), the probabilities $(p_o^{(i)})_{1 \leq i < r}$ and an estimation $p_e^{(r)}$ of $p_o^{(r)}$

Output. Depending on the estimation $p_e^{(r)}$, this algorithm returns :

- an optimal r -round trail $\mathcal{T}_o^{(r)}$ if $p_e^{(r)} \leq p_o^{(r)}$;
- the empty trail if $p_e^{(r)} > p_o^{(r)}$.

```

1   $\mathcal{T}_o^{(r)} \leftarrow ()$ 
2  For each non-zero output pattern  $b^{[0]}$  do
3    Call FirstRound( $b^{[0]}$ )
4  Return  $\mathcal{T}_o^{(r)}$ 

Function FirstRound( $b^{[0]}$ )
5     $a^{[0]} \leftarrow \arg \max \{ \text{DP}_\sigma(a, b^{[0]}) \mid a \in \mathbb{F}_2^{nm} \}$ 
6     $\mathcal{T}^{(1)} \leftarrow ((a^{[0]}, b^{[0]}))$ 
7     $p^{(1)} \leftarrow \text{DP}_\sigma(a^{[0]}, b^{[0]})$ 
8    If  $p^{(1)}$  is not less than the rank-one bound then
9      If  $r > 2$  then
10       | Call Round(2,  $\mathcal{T}^{(1)}, p^{(1)}$ )
11     Else
12       | Call LastRound( $\mathcal{T}^{(1)}, p^{(1)}$ )

Function Round( $s, \mathcal{T}^{(s-1)}, p^{(s-1)}$ )
13    $a^{[s-1]} \leftarrow \pi(b^{[s-2]})$ 
14   For each candidate  $b^{[s-1]}$  for  $a^{[s-1]}$  do
15      $p^{(s)} \leftarrow p^{(s-1)} \times \text{DP}_\sigma(a^{[s-1]}, b^{[s-1]})$ 
16     If  $p^{(s)}$  is not less than the rank- $s$  bound then
17        $\mathcal{T}^{(s)} \leftarrow \mathcal{T}^{(s-1)} \parallel (a^{[s-1]}, b^{[s-1]})$ 
18       If  $s + 1 < r$  then
19         | Call Round( $s + 1, \mathcal{T}^{(s)}, p^{(s)}$ )
20       Else
21         | Call LastRound( $\mathcal{T}^{(s)}, p^{(s)}$ )

Function LastRound( $\mathcal{T}^{(r-1)}, p^{(r-1)}$ )
22    $a^{[r-1]} \leftarrow \pi(b^{[r-2]})$ 
23    $b^{[r-1]} \leftarrow \arg \max \{ \text{DP}_\sigma(a^{[r-1]}, b) \mid b \in \mathbb{F}_2^{nm} \}$ 
24    $p^{(r)} \leftarrow p^{(r-1)} \times \text{DP}_\sigma(a^{[r-1]}, b^{[r-1]})$ 
25   If  $p^{(r)} \geq p_e^{(r)}$  then
26      $\mathcal{T}^{(r)} \leftarrow \mathcal{T}^{(r-1)} \parallel (a^{[r-1]}, b^{[r-1]})$ 
27      $\mathcal{T}_o^{(r)} \leftarrow \mathcal{T}^{(r)}$ 
28    $p_e^{(r)} \leftarrow p^{(r)}$ 

```

The current trail is saved

Figure 2.2: The search algorithm OptTrailEst for an optimal trail.

Lemma 2.5. Let s be an integer such that $1 \leq s < r$. Let \mathcal{T} be an s -round trail whose probability is less than the rank- s bound. Then, there does not exist any r -round trail extending \mathcal{T} with probability greater than or equal to $p_e^{(r)}$.

Proof. By contradiction, assume that \mathcal{T}_r is an r -round trail extending \mathcal{T} such that $\text{DP}(\mathcal{T}_r) \geq p_e^{(r)}$. Then the probability of the $(r-s)$ -round trail $\mathcal{T}_r^{[s,r-1]}$ is

$$\text{DP}(\mathcal{T}_r^{[s,r-1]}) = \frac{\text{DP}(\mathcal{T}) \times \text{DP}(\mathcal{T}_r^{[s,r-1]})}{\text{DP}(\mathcal{T})} = \frac{\text{DP}(\mathcal{T} \parallel \mathcal{T}_r^{[s,r-1]})}{\text{DP}(\mathcal{T})} = \frac{\text{DP}(\mathcal{T}_r)}{\text{DP}(\mathcal{T})}.$$

By assumption, $\text{DP}(\mathcal{T})$ is strictly less than $p_e^{(r)} / p_o^{(r-s)}$. Note that this strict inequality implies that $p_e^{(r)}$ is nonzero. It follows that

$$\text{DP}(\mathcal{T}_r^{[s,r-1]}) = \frac{\text{DP}(\mathcal{T}_r)}{\text{DP}(\mathcal{T})} \geq \frac{p_e^{(r)}}{\text{DP}(\mathcal{T})} > \frac{p_e^{(r)}}{p_e^{(r)} / p_o^{(r-s)}} = p_o^{(r-s)}.$$

By definition of $p_o^{(r-s)}$, this leads to a contradiction which proves the result. ■

Theorem 2.6 (validity of the algorithm). Depending on the estimation $p_e^{(r)}$, the algorithm **OptTrailEst** returns

- an optimal r -round trail $\mathcal{T}_o^{(r)}$ if $p_e^{(r)} \leq p_o^{(r)}$;
- the empty trail if $p_e^{(r)} > p_o^{(r)}$.

Proof. Suppose the condition on the bound to be removed. If $p_e^{(r)}$ is less than $p_o^{(r)}$, an optimal trail is saved in $\mathcal{T}_o^{(r)}$, otherwise $\mathcal{T}_o^{(r)}$ remains empty. Then, Lemma 2.5 ensures that the pruning condition avoids only the trails which have probabilities strictly less than $p_e^{(r)}$. The result still holds. ■

2.2. A Detailed Example

The algorithm **OptTrailEst** is a depth first search within the tree of all r -round differential trails together with a pruning mechanism which cuts only non-optimal trails. Any enhancement of this pruning condition directly impacts the algorithm complexity. Thus, a good algorithmic optimization of **OptTrailEst** rests on the right balance between cost and efficiency of such an enhancement.

Before addressing the formal treatment of our optimizations in Section 2.3, we introduce an example explaining each of them intuitively. Let us consider a 16-bit substitution permutation network quite similar to the **TOY CIPHER** presented in Example 1.14. First, it is worth recalling that the differential probability of a trail is computed over the associated long-key cipher. The key-schedule is thus disregarded throughout this chapter. To make this example more interesting, the substitution layer involves now four different 4-bit S-boxes denoted by S_0 , S_1 , S_2 and S_3 . The definition of these S-boxes and their respective differential probability matrices are given in Figure 2.3. As can be seen, the S-box S_0 is optimal with

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x)$	E	A	1	2	7	F	D	6	C	3	0	9	8	4	5	B
$S_1(x)$	C	A	E	0	D	3	1	8	B	2	9	4	5	6	7	F
$S_2(x)$	5	9	F	8	6	0	A	3	7	C	4	1	E	2	D	B
$S_3(x)$	9	5	7	D	A	C	2	4	E	F	6	B	0	1	8	3

$2^4 \times \text{DP}_{S_0}(a, b)$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16
1	.	.	.	2	2	.	.	.	2	2	.	2	2	.	2	2
2	2	2	4	.	2	2	.	4
3	.	4	2	4	.	2	2	2	.	.	.
4	.	.	2	.	4	4	.	2	.	2	.	.	2	.	.	.
5	.	2	2	2	.	.	4	2	2	.	2
6	.	.	.	2	.	.	2	.	4	2	.	.	2	4	.	.
7	.	2	.	.	2	2	2	4	2	2	.
8	.	2	2	2	2	.	4	.	2	.	2
9	.	.	2	4	.	.	4	2	2	2	.	.
A	.	2	4	2	2	2	2	2	.
B	.	.	2	2	.	2	4	.	2	.	4	.
C	2	.	2	.	.	2	.	2	2	2	2	2
D	.	.	2	2	4	.	2	2	.	.	4
E	.	4	.	.	.	2	4	2	.	2	.	2
F	4	4	.	.	.	4	4

$2^4 \times \text{DP}_{S_1}(a, b)$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16
1	.	.	.	2	.	.	2	.	2	4	.	.	.	2	4	.
2	.	.	6	.	.	.	2	.	.	2	2	2	2	.	.	.
3	.	2	2	.	2	2	2	2	2	.	.	2
4	.	2	.	.	2	.	.	.	2	2	.	2	.	.	4	2
5	.	2	.	2	.	.	4	4	2	.	2
6	.	.	4	4	4	4	.	.
7	.	2	.	.	4	2	2	.	4	.	2
8	2	2	2	6	4
9	.	2	2	.	.	2	2	2	.	.	4	2
A	.	.	2	.	2	4	2	.	2	.	4	.
B	.	.	2	2	2	.	.	2	2	.	.	2	2	2	.	.
C	.	2	2	.	2	4	.	.	4	.	.	2
D	.	4	.	.	.	2	.	2	2	.	2	4
E	2	2	2	2	.	.	4	4
F	.	.	.	6	.	2	.	.	2	2	2	.	.	2	.	.

$2^4 \times \text{DP}_{S_2}(a, b)$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16
1	2	4	2	.	2	.	2	4	.	.	.
2	.	2	.	6	2	2	.	2	2	.	.
3	4	4	.	2	.	2	.	.	2	.	2
4	.	.	.	2	.	2	.	.	.	6	2	2	.	.	2	.
5	.	.	4	.	.	4	4	.	.	4	.
6	.	.	.	2	.	.	.	2	2	2	6	.	.	.	2	.
7	.	2	.	2	.	.	4	2	.	2	4
8	.	.	4	.	.	2	.	2	4	2	.	2
9	.	2	.	.	2	2	.	.	2	.	8	.
A	.	4	.	.	4	.	.	.	4	.	.	4
B	.	.	.	2	2	.	.	.	2	8	.	2
C	.	2	4	2	4	2	.	2	.	.
D	2	2	.	8	.	.	2	2
E	.	4	4	2	.	2	.	.	2	.	2
F	6	.	4	2	2	2	.

$2^4 \times \text{DP}_{S_3}(a, b)$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16
1	.	4	4	.	.	.	2	2	2	2	.	.
2	.	.	2	.	2	.	.	.	10	2	.
3	.	.	2	2	2	2	.	.	.	4	4	.
4	.	.	.	2	.	2	.	.	2	4	6	.
5	.	.	.	4	.	4	8
6	.	4	4	.	.	.	2	2	2	2	.	.
7	8	.	.	.	4	.	4	.	.
8	.	2	2	4	.	.	6	.	.	2	.	.
9	.	2	2	6	6	.	.	.
A	.	.	6	.	.	2	.	.	.	2	2	4
B	.	.	2	6	4	.	.	.	2	2
C	.	.	.	2	6	2	2	4
D	.	.	4	.	2	6	.	.	2	2
E	.	2	2	4	.	.	.	2	6	.	.	.
F	.	2	2	.	.	.	6	.	.	6	.	.

Figure 2.3: The S-boxes and their difference probability matrices used in Section 2.2.

respect to differential cryptanalysis [72] whereas the others are not. More precisely, their maximal differential probabilities are given by

$$\text{DP}_{S_0}^{\max} = \frac{4}{16}, \quad \text{DP}_{S_1}^{\max} = \frac{6}{16}, \quad \text{DP}_{S_2}^{\max} = \frac{8}{16}, \quad \text{DP}_{S_3}^{\max} = \frac{10}{16}.$$

The substitution layer of this cipher is the permutation σ of $(\mathbb{F}_2^4)^4$ defined by the formula

$$\sigma(x_0, x_1, x_2, x_3) = (S_0(x_0), S_1(x_1), S_2(x_2), S_3(x_3)).$$

For instance, σ maps 0000 to EC59. Next, the diffusion layer is the bit permutation associated with the permutation ϕ of $\llbracket 0, 16 \rrbracket$ defined by the rule

$$\phi(i) = 4(i \bmod 4) + \left\lfloor \frac{i}{4} \right\rfloor.$$

Thus, this diffusion layer is exactly the same as the one of TOY CIPHER. We now know the full specification of the round function.

Clearly, an optimal 1-round differential trail activates only the S-box which has the highest differential probability, namely S_3 for this cipher. As can be seen in Figure 2.3, $\text{DP}_{S_3}(2, 8) = \frac{10}{16}$, so the pair $(2, 8)$ of input/output difference patterns has the maximal probability over S_3 . Consequently, the differential trail

$$\mathcal{T}_o^{(1)} = ((a^{[0]}, b^{[0]})) = (((0, 0, 0, 2), (0, 0, 0, 8)))$$

is optimal and holds with probability $p_o^{(1)} = \frac{10}{16}$. In this example, the trail $\mathcal{T}_o^{(1)}$ happens to be the only 1-round optimal trail. However, if all the S-boxes are equal to S_0 , it goes without saying that there are many optimal 1-round differential trails. For each $1 \leq w \leq 4$, we denote by $\text{DP}_{(w)}^{\max}$ the maximal probability of a 1-round differential trail activating w S-boxes. The previous discussion ensures that

$$\text{DP}_{(1)}^{\max} = \text{DP}(\mathcal{T}_o^{(1)}) = \frac{10}{16}.$$

Naturally, a 1-round trail activating two S-boxes have maximum probability if and only if it activates S_2 and S_3 with their maximum differential probabilities. Therefore,

$$\text{DP}_{(2)}^{\max} = \text{DP}_{S_2}^{\max} \times \text{DP}_{S_3}^{\max} = \frac{8}{16} \times \frac{10}{16} = \frac{80}{16^2}.$$

Similarly, we can compute that

$$\text{DP}_{(3)}^{\max} = \prod_{i=1}^3 \text{DP}_{S_i}^{\max} = \frac{480}{16^3} \quad \text{and} \quad \text{DP}_{(4)}^{\max} = \prod_{i=0}^3 \text{DP}_{S_i}^{\max} = \frac{1920}{16^4}.$$

Assume that the SPN described above consists of seven rounds. To compute an optimal 7-round differential trail, the algorithm **OptTrailEst** requires the probabilities $p_o^{(i)}$ of optimal i -round trails for each $1 \leq i < 7$ and an estimation $p_e^{(7)}$ of the probability of the 7-round optimal trail searched. As explained in Section 2.1.1, the probabilities $(p_o^{(i)})_{i < 7}$ are obtained with five previous iterations of **OptTrail**.

	$\mathcal{T}_o^{(1)}$	$\mathcal{T}_o^{(2)}$	$\mathcal{T}_o^{(3)}$	$\mathcal{T}_o^{(4)}$	$\mathcal{T}_o^{(5)}$	$\mathcal{T}_o^{(6)}$
$a^{[0]}$	(0, 0, 0, 2)	(0, 0, A, 0)	(0, 0, F, 0)	(0, 0, F, 0)	(0, 0, 0, 2)	(0, 0, A, 0)
$b^{[0]}$	(0, 0, 0, 8)	(0, 0, 1, 0)	(0, 0, 4, 0)	(0, 0, 4, 0)	(0, 0, 0, 8)	(0, 0, 1, 0)
$a^{[1]}$		(0, 0, 0, 2)	(0, 2, 0, 0)	(0, 2, 0, 0)	(1, 0, 0, 0)	(0, 0, 0, 2)
$b^{[1]}$		(0, 0, 0, 8)	(0, 2, 0, 0)	(0, 2, 0, 0)	(4, 0, 0, 0)	(0, 0, 0, 8)
$a^{[2]}$			(0, 0, 4, 0)	(0, 0, 4, 0)	(0, 8, 0, 0)	(1, 0, 0, 0)
$b^{[2]}$			(0, 0, 9, 0)	(0, 0, 9, 0)	(0, 8, 0, 0)	(4, 0, 0, 0)
$a^{[3]}$				(2, 0, 0, 2)	(4, 0, 0, 0)	(0, 8, 0, 0)
$b^{[3]}$				(A, 0, 0, 8)	(4, 0, 0, 0)	(0, 8, 0, 0)
$a^{[4]}$					(0, 8, 0, 0)	(4, 0, 0, 0)
$b^{[4]}$					(0, 7, 0, 0)	(4, 0, 0, 0)
$a^{[5]}$						(0, 8, 0, 0)
$b^{[5]}$						(0, 7, 0, 0)
DP	$10/16$	$40/16^2$	$6^3/16^3$	$40 \times 6^3/16^5$	$120 \times 4^2/16^5$	$120 \times 4^3/16^6$

Figure 2.4: The optimal trails of the example

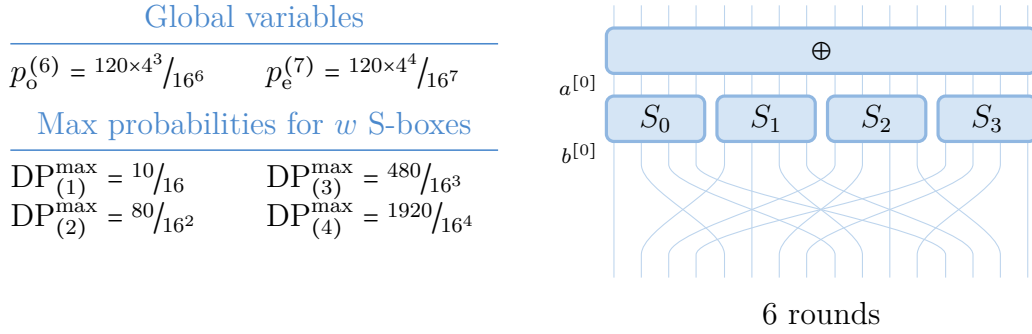


Figure 2.5: Example of Run for OptTrailEst

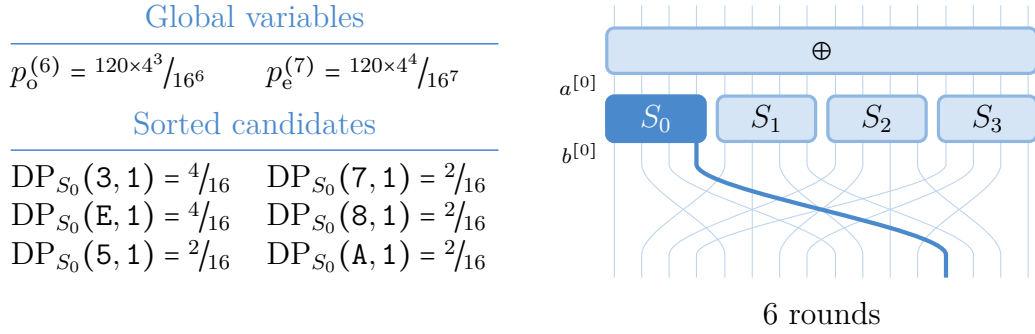
These optimal trails are given in Figure 2.4. Recall that their probabilities are computed by simply multiplying the differential probabilities of their active S-boxes. For instance,

$$\begin{aligned}
 DP(\mathcal{T}_o^{(3)}) &= DP_{S_3}(F, 4) \times DP_{S_2}(2, 2) \times DP_{S_3}(4, 9) \times DP_{S_0}(2, A) \times DP_{S_4}(2, 8) \\
 &= \left(\frac{6}{16}\right)^3 \times \frac{4}{16} \times \frac{10}{16} = \frac{40 \times 6^3}{16^5}.
 \end{aligned}$$

Finally, we choose $p_e^{(7)} = (120 \times 4^4) / 16^7$ as estimation of $p_o^{(7)}$. Recall that an automatic management of this estimation will given later in Section 2.3.5. But for now, let us detail some carefully chosen steps of the algorithm `OptTrailEst`. The parameters of this execution are summarized in Figure 2.5.

2.2.1. Search Algorithm for the First Round

Following the algorithm `OptTrailEst`, we must try to extend all the output patterns $b^{[0]}$ of the first round. Intuitively, a trail activating $(w + 1)$ S-boxes in the first

Figure 2.6: Example of Run for `FirstRound()` – Part 1

round is less likely to be optimal than a trail activating w S-boxes in this same round. Therefore, the patterns $b^{[0]}$ should not be tested in the natural order but according to the number of S-boxes they activate, or equivalently to their bundle weights (see Section 1.5.2). Indeed, once a 7-round trail with probability higher than the estimation $p_e^{(7)}$ is found, this estimation is updated and the pruning condition is enhanced for the rest of the execution. Consequently, the earlier high probability trails are found, the lower is the complexity of this algorithm.

Thus, the first pattern we may try to extend is $b^{[0]} = (1, 0, 0, 0)$. This step is illustrated in Figure 2.6. Since a first output pattern $b^{[0]}$ have just we been chosen, we are now executing the function `FirstRound`. First, an input pattern $a^{[0]}$ maximizing the probability $DP_\sigma(a^{[0]}, b^{[0]})$ must be picked out. Because S_0 is the only S-box activated by $b^{[0]}$, this amounts to find an element a_0 maximizing $DP_{S_0}(a_0, 1)$ and then define $a^{[0]} = (a_0, 0, 0, 0)$. All the nonzero differential probabilities of the form $DP_{S_0}(a_0, 1)$ are enumerated in Figure 2.6. These values are directly drawn from the difference probability matrix of S_0 previously given in Figure 2.3. Thus, a_0 can be equal to 3 or E, it does not matter. Our current trail $\mathcal{T}^{(1)}$ is then equal to

$$\mathcal{T}^{(1)} = ((a^{[0]}, b^{[0]})) = (((3, 0, 0, 0), (1, 0, 0, 0)))$$

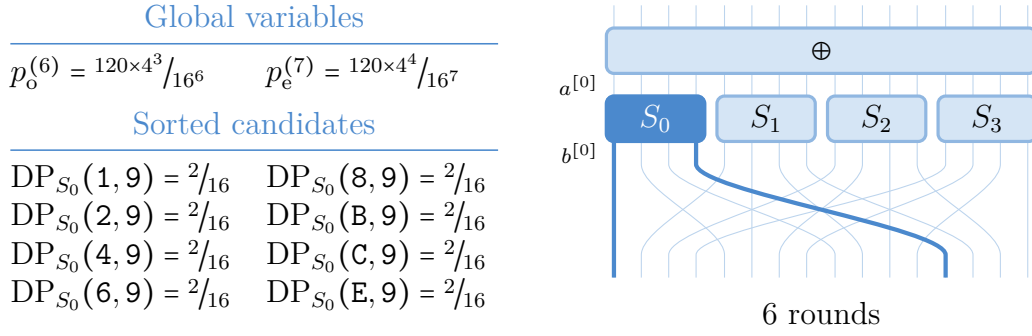
and $DP(\mathcal{T}^{(1)}) = \frac{4}{16}$. To obtain a 7-round trail, the current trail $\mathcal{T}^{(1)}$ must be extended by a 6-round trail. In the best-case scenario, its 6-round extension is optimal and the resulting 7-round trail has probability

$$DP(\mathcal{T}^{(1)}) \times p_o^{(6)} = \frac{4}{16} \times \frac{120 \times 4^3}{16^6} = \frac{120 \times 4^4}{16^7}$$

which is (greater than or) equal to the estimation $p_e^{(7)}$. Consequently, its probability is consistent with our estimation and it is worth trying to extend the trail $\mathcal{T}^{(1)}$. Assuming that our estimation is less than $p_o^{(7)}$, this means that the current trail can potentially be extended into an optimal 7-round trail. This trail is then handled by the function `Round(2)` and the input pattern for this second round is

$$a^{[1]} = \pi(b^{[0]}) = (0, 0, 0, 8).$$

This function, helped by all its recursive calls, tries all possible extensions of $\mathcal{T}^{(1)}$. Unfortunately any of them yields an optimal 7-round trail.


 Figure 2.7: Example of Run for **FirstRound()** – Part 2

The next step of the algorithm **OptTrailEst** is simply to try to extend another difference pattern $b^{[0]}$. Certainly, we try $b^{[0]} = (2, 0, 0, 0)$, then $b^{[0]} = (3, 0, 0, 0)$ and so on. Let us skip these steps until we reach the pattern $b^{[0]} = (9, 0, 0, 0)$, represented in Figure 2.7. Again, the function **FirstRound** requires to select an input pattern $a^{[0]}$ that maximizes the probability of the current trail. Since this step is repeated many times during the execution of **OptTrailEst**, all the input patterns

$$\arg \max \{ DP_{S_i}(a_i, b_i) \mid a_i \in \mathbb{F}_2^n \}$$

for every $i < n$ and every b_i in \mathbb{F}_2^n should be computed and stored before running the algorithm. This time, all the candidates a_0 have the same differential probability, so we choose $a_0 = 1$ and the current trail becomes

$$\mathcal{T}^{(1)} = ((a^{[0]}, b^{[0]})) = (((1, 0, 0, 0), (9, 0, 0, 0))) \quad \text{and} \quad DP(\mathcal{T}^{(1)}) = \frac{2}{16}.$$

If $\mathcal{T}^{(1)}$ is extended by an optimal 6-round trail, the resulting trail has probability

$$DP(\mathcal{T}^{(1)}) \times p_o^{(6)} = \frac{2}{16} \times \frac{120 \times 4^3}{16^6} = \frac{120 \times 2 \times 4^3}{16^7}$$

which is less than our estimation. Using the vocabulary introduced in Section 2.1.1, the probability of the current trail is less than the rank-1 bound and this trail (and thus all its extensions) can be discarded without missing an optimal 7-round trail.

Once all the patterns $b^{[0]}$ activating one S-box in the first round are handled, we consider the patterns activating two S-boxes. Before trying to extend all these patterns, we should test if this effort is worthwhile. Since $DP_{(2)}^{\max}$ is equal to $\frac{80}{16^2}$, the best 1-round trail that can be obtained in the function **FirstRound** has probability $\frac{80}{16^2}$. By computing

$$DP_{(2)}^{\max} \times p_o^{(6)} = \frac{80}{16^2} \times \frac{120 \times 4^3}{16^6} = \frac{20}{16} \times p_e^{(7)}$$

we see that this probability is greater than $p_e^{(7)}$ so these patterns must be considered. Nevertheless, applying the same test to the patterns which activate three S-boxes in the first round yields

$$DP_{(3)}^{\max} \times p_o^{(6)} = \frac{480}{16^3} \times \frac{120 \times 4^3}{16^6} = \frac{120}{16^2} \times p_e^{(7)} < p_e^{(7)}.$$

Global variables	
$p_o^{(3)} = 6^3/16^3$	$p_e^{(7)} = 120 \times 4^4/16^7$
$p_o^{(4)} = 40 \times 6^3/16^5$	
Current trail	
$a^{[0]} = (0, 2, 0, 0)$	$a^{[1]} = (0, 0, 4, 0)$
$b^{[0]} = (0, 2, 0, 0)$	$b^{[1]} = (0, 0, 9, 0)$
$DP(\mathcal{T}^{(2)}) = 6^2/16^2$	
Max probabilities for w S-boxes	
$DP_{(1)}^{\max} = 10/16$	$DP_{(3)}^{\max} = 480/16^3$
$DP_{(2)}^{\max} = 80/16^2$	$DP_{(4)}^{\max} = 1920/16^4$
Sorted Candidates	
$DP_{S_0}(2, A) = 4/16$	$DP_{S_3}(2, 8) = 10/16$
$DP_{S_0}(2, F) = 4/16$	$DP_{S_3}(2, 2) = 2/16$
$DP_{S_0}(2, 8) = 2/16$	$DP_{S_3}(2, 4) = 2/16$
$DP_{S_0}(2, 9) = 2/16$	$DP_{S_3}(2, E) = 2/16$
$DP_{S_0}(2, C) = 2/16$	
$DP_{S_0}(2, D) = 2/16$	

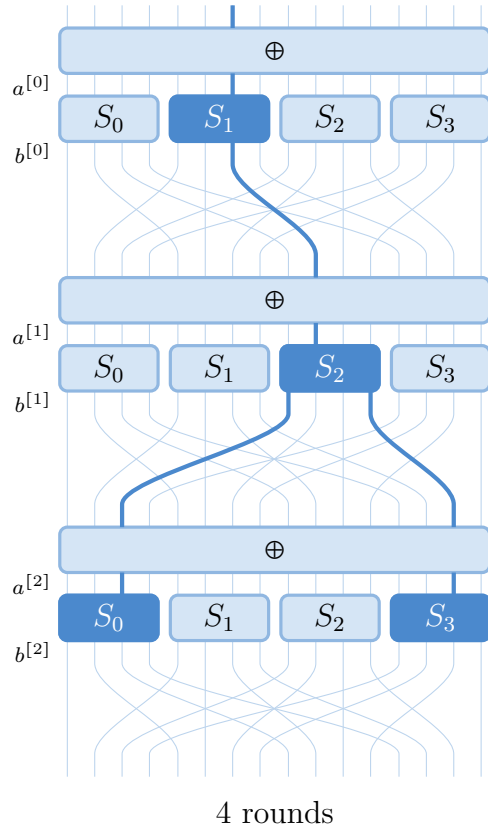


Figure 2.8: Example of Run for Round(3) – Part 1

Consequently, it is useless to consider the patterns $b^{[0]}$ activating three or four S-boxes in the first round. To conclude, using this additional costless test, we have considered

$$\binom{4}{1} \times 16^1 + \binom{4}{2} \times 16^2 = 1\,600$$

patterns instead of $2^{16} = 65\,536$.

2.2.2. Search Algorithm for the Round Function

To explain our optimizations of the function **Round**, assume that we have already handle the first two rounds and that the current trail $\mathcal{T}^{(2)}$ is as illustrated in Figure 2.8. This trail has a differential probability equal to $\frac{6^2}{16^2}$ and the input pattern of the third round is

$$a^{[2]} = \pi(b^{[1]}) = (2, 0, 0, 2).$$

According to the function **Round(3)**, every candidate $b^{[2]}$ for $a^{[2]}$ must be considered and then tested by the pruning mechanism. However, we will create these candidates recursively bundle by bundle. For this purpose, the output candidate patterns of the two active S-boxes are sorted according to their probabilities. First, we choose the best output pattern for S_0 , namely $b_0^{[2]} = A$, represented in Figure 2.9. Before selecting the output pattern for the other active S-box, this first choice should be tested as follows. The current probability for this round is $DP_{S_0}(2, A)$ and it remains

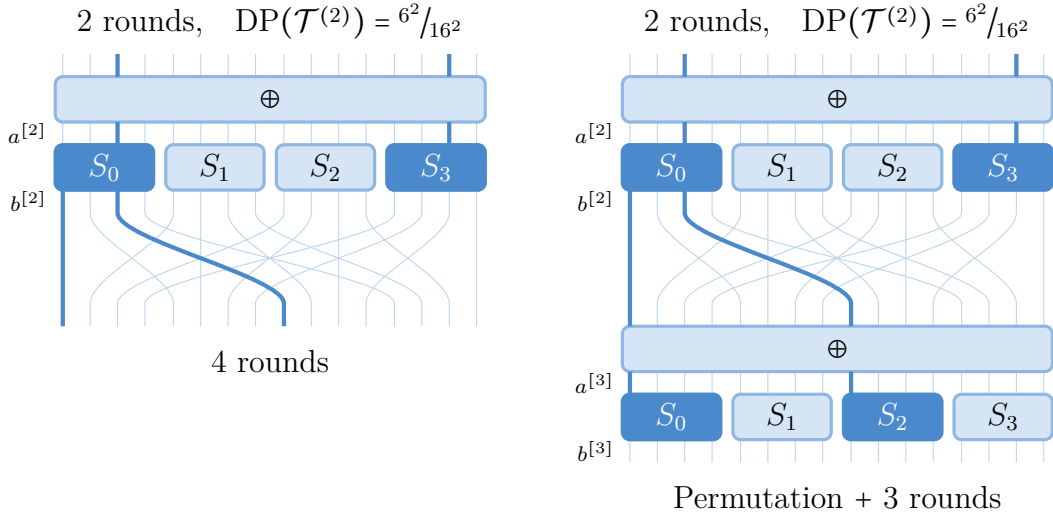


Figure 2.9: Example of Run for Round(3) – Part 2

one active S-box. The differential probability of this other S-box is clearly upper-bounded by $DP_{(1)}^{\max}$ and thus the probability of the round is upper-bounded by the product $DP_{S_0}(2, A) \times DP_{(1)}^{\max}$. Since it remains four rounds before reaching a 7-round trail, the probability of any extension can be upper-bounded by

$$DP(\mathcal{T}^{(2)}) \times (DP_{S_0}(2, A) \times DP_{(1)}^{\max}) \times p_o^{(4)} = \frac{6^2}{16^2} \times \frac{4}{16} \times \frac{10}{16} \times \frac{40 \times 6^3}{16^5}. \quad (2.1)$$

This value is greater than the estimation $p_e^{(7)}$, and hence our first choice $b_0^{[2]} = A$ seems to be a good one.

We now introduce a second pruning condition for this same candidate. This new condition applies whenever the diffusion layer of the SPN is a bit permutation. Observe that the candidate $b_0^{[0]} = A$ activates two S-boxes in the next round. And no matter what the choice of second candidate is, the input pattern of the next round will activate at least two S-boxes. We have already upper-bounded the probability of this round by $DP_{S_0}(2, A) \times DP_{(1)}^{\max}$. The probability of the next round is at most equal to $DP_{(2)}^{\max}$ and it remains three rounds to reach the seven rounds. Therefore, the probability of any extension is upper-bounded by

$$DP(\mathcal{T}^{(2)}) \times (DP_{S_0}(2, A) \times DP_{(1)}^{\max}) \times DP_{(2)}^{\max} \times p_o^{(3)} = \frac{6^2}{16^2} \times \frac{4}{16} \times \frac{10}{16} \times \frac{10}{16} \times \frac{6^3}{16^3}. \quad (2.2)$$

This probability is greater than the estimation and our first choice is now completely confirmed.

Next, we focus on the second active S-box, namely S_3 . Referring to Figure 2.8, the first candidate that must be chosen is $b_3^{[2]} = 8$. With this choice, the output pattern $b^{[2]}$ of this round is complete and equal to $(A, 0, 0, 8)$. However, this pattern must pass the two pruning tests before the current trail can be extended. It is easily checked from Figure 2.10 that these tests involves the same computation as in (2.1) and (2.2), so they accept the output pattern $b^{[2]}$. Finally, the pattern $a^{[3]} = \pi(b^{[2]}) = (9, 0, 8, 0)$ is handled by the function **Round(4)**.

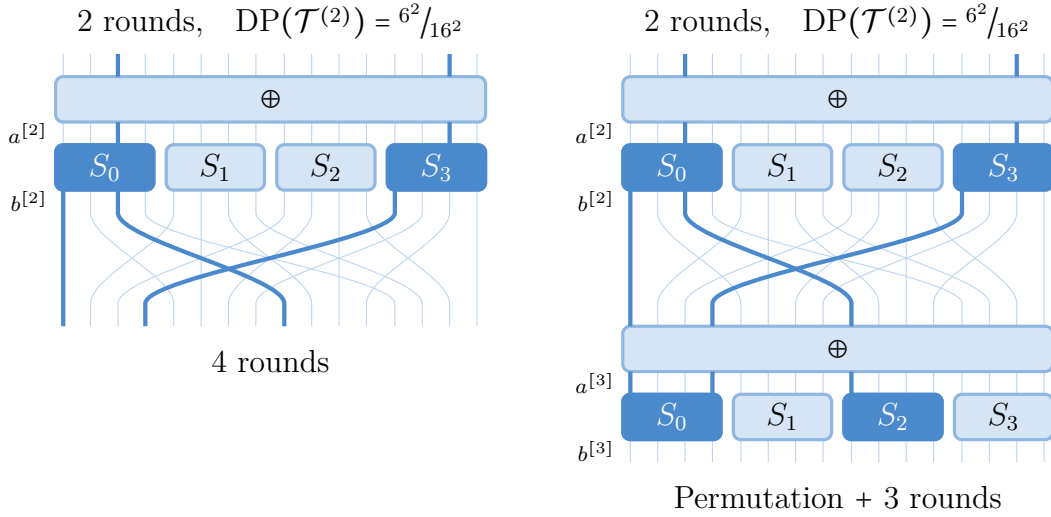


Figure 2.10: Example of Run for Round(3) – Part 3

Once all the extensions of $a^{[3]}$ are explored, the other candidates for $a^{[2]}$ need to be considered. According to Figure 2.8, the second best candidate for $a_3^{[2]}$ is $b_3^{[2]} = 2$. Again, the probability of any 4-round extension of the current trail is upper-bounded by

$$\text{DP}(\mathcal{T}^{(2)}) \times (\text{DP}_{S_0}(2, \mathbf{A}) \times \text{DP}_{S_3}(2, 2)) \times p_o^{(4)} = \frac{6^2}{16^2} \times \frac{4}{16} \times \frac{2}{16} \times \frac{40 \times 6^3}{16^5}.$$

This time, this upper bound is less than the estimation $p_e^{(7)}$ and this candidate is discarded. In other words, the probability of the current trail is less than the rank-3 bound. Recall that the candidates are sorted according to their probabilities. Hence the remaining two candidates 4 and E for $a_3^{[2]}$ can also be discarded without any additional computing. Since every candidate of the second active S-box has been considered, this recursive call ends and we go back to the first active S-box.

Referring to Figure 2.8, the next candidate for $a_0^{[2]}$ is $b_0^{[2]} = \mathbf{F}$. Since this new candidate has the same differential probability as its predecessor, the first pruning mechanism computes the same upper bound as in (2.1) and validates this choice. However, Figure 2.11 illustrates that this candidate activates every S-box in the next round. The second upper bound is hence

$$\text{DP}(\mathcal{T}^{(2)}) \times (\text{DP}_{S_0}(2, \mathbf{F}) \times \text{DP}_{(1)}^{\max}) \times \text{DP}_{(4)}^{\max} \times p_o^{(3)} = \frac{6^2}{16^2} \times \frac{4}{16} \times \frac{10}{16} \times \frac{1920}{16^4} \times \frac{6^3}{16^3},$$

which is less than the estimation, discarding this candidate. The next candidate is $b_0^{[2]} = 8$. The probability of any complete extension of this trail is upper-bounded by

$$\text{DP}(\mathcal{T}^{(2)}) \times (\text{DP}_{S_0}(2, 8) \times \text{DP}_{(1)}^{\max}) \times p_o^{(4)} = \frac{6^2}{16^2} \times \frac{2}{16} \times \frac{10}{16} \times \frac{40 \times 6^3}{16^5}.$$

This bound is less than the estimation. As a consequence this and the three remaining candidates for $a_0^{[2]}$ are all rejected, which completes the search for all extensions of the trail $\mathcal{T}^{(2)}$ and this recursive call to Round(3).

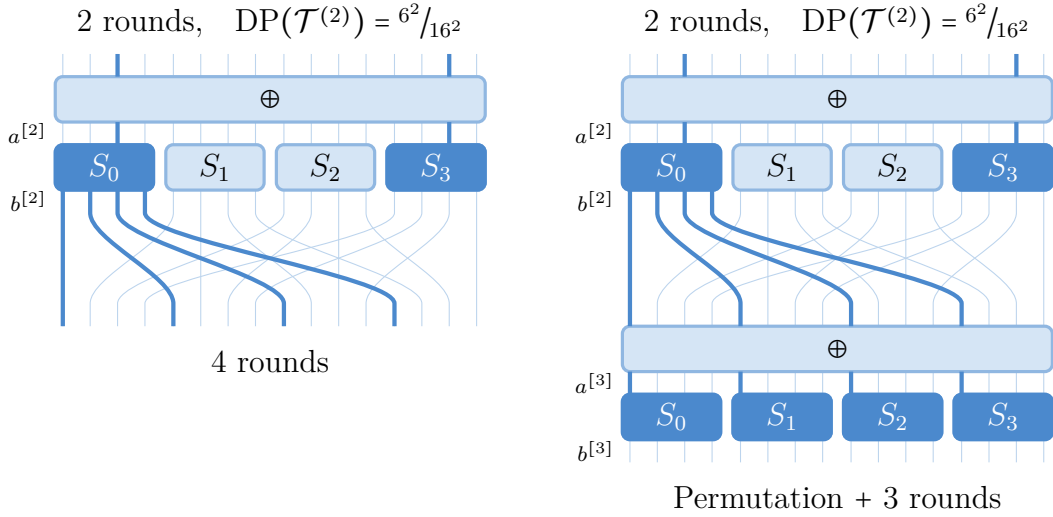


Figure 2.11: Example of Run for Round(3) – Part 4

To conclude, let us compare the number of patterns considered by the optimized and non-optimized versions of **Round**. Following the algorithm **OptTrailEst** given in Section 2.1.1, we would have to try every candidate $b^{[2]}$ for $a^{[2]}$, namely the 24 patterns in

$$\{(b_0^{[2]}, 0, 0, b_3^{[2]}) \mid b_0^{[2]} \in \{8, 9, A, C, D, F\}, b_3^{[2]} \in \{2, 4, 8, E\}\}.$$

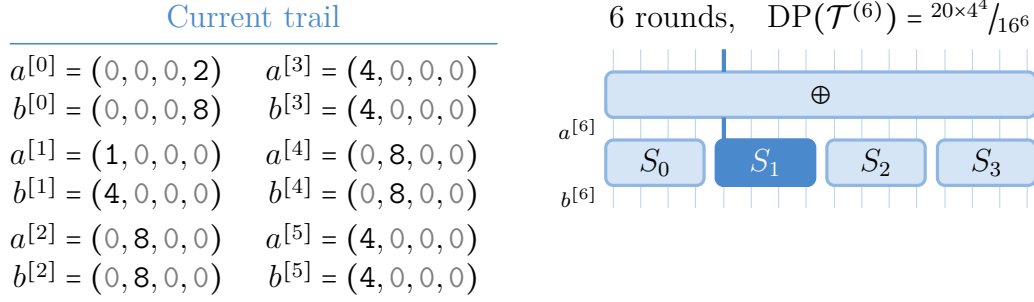
In this optimized version, we have considered only the two complete output patterns $(A, 0, 0, 8)$, $(A, 0, 0, 2)$ and the two half patterns $(F, 0, 0, ?)$, $(2, 0, 0, ?)$. Moreover, it is worth observing that all the sorted output candidates over each S-box with their respective differential probabilities should be computed and stored before starting the search. Similarly, all the rank- s bounds $p_e^{(r)} / p_o^{(r-s)}$ are updated and stored after each modification of the estimation. Finally, the probability of the current trail is computed recursively. Therefore, each pruning test requires at most two multiplications and one comparison.

2.2.3. Search Algorithm for the Last Round

To finish this example, we should consider the function **LastRound**. Nevertheless, this function is called only few times during the execution of **OptTrailEst** and is already very efficient. Assume that the current 6-round trail $\mathcal{T}^{(6)}$ is as in Figure 2.12. The corresponding input pattern of this last round is

$$a^{[6]} = \pi(b^{[5]}) = \pi(4, 0, 0, 0) = (0, 8, 0, 0).$$

Therefore there is only one S-box activated in the last round. Its best output candidate is also obtained by choosing the best output candidate for each active S-box. In this case, we must choose an element $b_1^{[6]}$ maximizing the probability $DP_{S_1}(8, b_1^{[6]})$. According to the difference probability matrix of S_1 given in Figure 2.3, the only choice is $b_1^{[6]} = 7$. Thus, the final output pattern $b^{[6]}$ is equal to

Figure 2.12: Example of Run for `LastRound()`

$(0, 7, 0, 0)$. It remains to compute the differential probability of this 7-round trail and compare it with the estimation. We have

$$DP(\mathcal{T}^{(7)}) = DP(\mathcal{T}^{(6)}) \times DP_{\sigma}(a^{[6]}, b^{[6]}) = \frac{20 \times 4^4}{16^6} \times \frac{6}{16} = \frac{120 \times 4^4}{16^7}.$$

Therefore, its probability is equal to the estimation, so this trail is save in $\mathcal{T}_o^{(7)}$. The estimation $p_e^{(7)}$ is then updated even if in this case its value does not change. The call to the function `LastRound` and the algorithm continues the execution of `Round(6)`.

2.3. Optimizations

This section is dedicated to give a theoretical framework of the optimizations introduced in the previous example. First observe that the first loop of `OptTrailEst` requires to call the function `FirstRound` for all non-zero output differences $b^{[0]}$. Since there are $2^{nm} - 1$ such differences, we can lower-bound its complexity by 2^{64} or 2^{128} for real-sized substitution-permutation networks. Therefore, this algorithm must be optimized for any practical execution.

2.3.1. Construction of the First Output Pattern

As we have said above, the number of calls to the function `FirstRound()` is a problem that must be solved. To optimize this step, a partition of the set of all non-zero differences is defined. Then, we give an effective way to test whether no difference in one part can be the beginning of an optimal trail.

For each integer w such that $1 \leq w \leq m$, we denote by $DP_{(w)}^{\max}$ the maximal probability of any 1-round trail activating w S-boxes. In other words,

$$DP_{(w)}^{\max} = \max\{DP_{\sigma}(a, b) \mid a, b \in (\mathbb{F}_2^n)^m \text{ such that } w_n(a) = w\},$$

where $w_n(a)$ denotes the *bundle weight* of a . Then, let us sort the differential probabilities $DP_{S_i}^{\max}$ in the decreasing order. This is equivalent to define a permutation τ of $\llbracket 0, m \rrbracket$ such that for each $i < m - 1$, it holds that

$$DP_{S_{\tau(i)}}^{\max} \geq DP_{S_{\tau(i+1)}}^{\max}.$$

Proposition 2.7. Let w be an integer such that $1 \leq w \leq m$. Then,

$$\text{DP}_{(w)}^{\max} = \prod_{i=0}^{w-1} \text{DP}_{S_{\tau(i)}}^{\max}.$$

Proof. Let a be an input pattern activating w S-boxes and b be an output pattern. For each $0 \leq i < m$, denote by p_i the differential probability $\text{DP}_{S_i}(a_i, b_i)$. Let ρ be a permutation of $\llbracket 0, m \rrbracket$ such that $p_{\rho(i)} \geq p_{\rho(i+1)}$ for all $i < m-1$. Since the pattern a activates w S-boxes, it must be the case that $p_{\rho(i)} = 0$ for each $i \geq w$. Thus,

$$\text{DP}_{\sigma}(a, b) = \prod_{i=0}^{m-1} \text{DP}_{S_i}(a_i, b_i) = \prod_{i=0}^{m-1} p_i = \prod_{i=0}^{m-1} p_{\rho(i)} = \prod_{i=0}^{w-1} p_{\rho(i)} \leq \prod_{i=0}^{w-1} \text{DP}_{S_{\rho(i)}}^{\max}.$$

By definition of τ , $\text{DP}_{S_{\rho(i)}}^{\max} \leq \text{DP}_{S_{\tau(i)}}^{\max}$ for each $0 \leq i < w$. Therefore,

$$\text{DP}_{\sigma}(a, b) \leq \prod_{i=0}^{w-1} \text{DP}_{S_{\tau(i)}}^{\max}. \quad (2.3)$$

As this inequality holds for every a and b in $(\mathbb{F}_2^m)^n$ such that $w_n(a) = w$, it follows that

$$\max\{\text{DP}_{\sigma}(a, b) \mid a, b \in (\mathbb{F}_2^m)^n, w_n(a) = w\} \leq \prod_{i=0}^{w-1} \text{DP}_{S_{\tau(i)}}^{\max}.$$

Clearly, there exists a pair (a, b) of input/output patterns with $w_n(a) = w$ such that $\text{DP}_{\sigma}(a, b)$ meets the bound (2.3) with equality, proving our proposition. ■

Remark 2.8. It goes without saying that $\text{DP}_{(1)}^{\max} \geq \dots \geq \text{DP}_{(m)}^{\max}$ hold. Thus, the probability of an optimal one-round trail is

$$p_o^{(1)} = \max\{\text{DP}_{\sigma}(a, b) \mid a, b \in (\mathbb{F}_2^{nm})^*\} = \text{DP}_{(1)}^{\max} = \text{DP}_{S_{\tau(0)}}^{\max}.$$

Of course, the differential probability matrices DP_{S_i} and the probabilities $\text{DP}_{S_i}^{\max}$ and $\text{DP}_{(i)}^{\max}$ are computed and stored before starting the search.

Theorem 2.9. Let w and w' be two integers such that $1 \leq w \leq w' \leq m$. If $\text{DP}_{(w)}^{\max}$ is less than the rank-one bound, then there exists no r -round trail activating w' S-boxes in the first round with probability greater than or equal to $p_e^{(r)}$.

Proof. Assume that $\text{DP}_{(w)}^{\max}$ is less than the rank-one bound. Let \mathcal{T} be a one-round trail activating w' S-boxes. By definition, $\text{DP}(\mathcal{T}) \leq \text{DP}_{(w')}^{\max}$. Then, the inequality $\text{DP}_{(w')}^{\max} \leq \text{DP}_{(w)}^{\max}$ obviously holds, and thus $\text{DP}(\mathcal{T}) \leq \text{DP}_{(w)}^{\max}$. Therefore, $\text{DP}(\mathcal{T})$ is less than the rank-one bound and Lemma 2.5 ensures that there does not exist any r -round trail extending \mathcal{T} with probability greater than or equal to $p_e^{(r)}$. This concludes the proof. ■

This theorem states that whenever $\text{DP}_{(w)}^{\max}$ is less than the rank-one bound, we only have to test the output differences $b^{[0]}$ activating at most $(w-1)$ S-boxes. There are

$$\sum_{i=1}^{w-1} \binom{m}{w} (2^n - 1)^i$$

Algorithm 2 – OptTrailEst

```

1  $\mathcal{T}_o^{(r)} \leftarrow ()$ 
2 For  $w$  from 1 to  $m$  do
3   If  $\text{DP}_{(w)}^{\max}$  is lower than the rank-one bound then
4     Exit the loop
5   Else
6     For each output pattern  $b^{[0]}$  activating  $w$  S-boxes do
7       Call FirstRound( $b^{[0]}$ )
8 Return  $\mathcal{T}_o^{(r)}$ 

```

Figure 2.13: First optimization – construction of the first difference

such differences, compared to $2^{nm} - 1$ without this optimization.

We have run the final algorithm with several SPN having a bit permutation as linear layer. With $m = 16$ and $n = 4$, $\text{DP}_{(4)}^{\max}$ was always less than the rank-one bound, and hence there was at most 2^{21} difference patterns $b^{[0]}$ to test instead of 2^{64} . With $m = 16$ and $n = 8$, the gap is even larger since $\text{DP}_{(3)}^{\max}$ was always less than the rank-one bound, yielding 2^{21} difference patterns to test instead of 2^{128} . The algorithm optimized with Theorem 2.9 is described in Figure 2.13.

2.3.2. The Round Function

Following Matsui's algorithm [76], the output candidates of the function **Round** are constructed recursively. Let a denote the input difference of the current round. According to Propositions 1.24 and 1.36, any candidate b for a can be constructed by selecting an output pattern for each S-box activated by a . The following theorem establishes that the pruning mechanism can be applied bundle by bundle.

Theorem 2.10. Let s be an integer such that $1 \leq s \leq r$ and \mathcal{T} be an s -round trail. Denote by $x_0 < \dots < x_{w-1}$ the indices of the S-boxes activated by $a^{[s-1]}$ where $w = w_n(a^{[s-1]})$. Let v be an integer satisfying $1 \leq v \leq w$. If

$$\text{DP}(\mathcal{T}^{[0,s-2]}) \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max}$$

is less than the s -rank bound, then for every pattern c satisfying:

- $c_{x_i} = b_{x_i}^{[s-1]}$ for each $i < v - 1$, and
- $\text{DP}_{S_{x_{v-1}}}(a_{x_{v-1}}^{[s-1]}, c_{x_{v-1}}) \leq \text{DP}_{S_{x_{v-1}}}(a_{x_{v-1}}^{[s-1]}, b_{x_{v-1}}^{[s-1]})$,

there does not exist any r -round trail extending $\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)$ with probability greater than or equal to $p_e^{(r)}$.

Proof. Let c be an output pattern satisfying the required conditions. If c is not a candidate for $a^{[s-1]}$, then $\text{DP}_\sigma(a^{[s-1]}, c) = 0$ and any trail extending the current trail $\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)$ has also zero probability. Therefore, we assume that c is a

candidate for $a^{[s-1]}$ in the following. By hypothesis,

$$\begin{aligned} \text{DP}_\sigma(a^{[s-1]}, c) &= \prod_{i=0}^{w-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, c_{x_i}) \\ &= \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, c_{x_i}) \right) \times \left(\prod_{i=v}^{w-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, c_{x_i}) \right), \end{aligned}$$

And thus

$$\begin{aligned} \text{DP}_\sigma(a^{[s-1]}, c) &\leq \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \left(\prod_{i=v}^{w-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, c_{x_i}) \right) \\ &\leq \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max}. \end{aligned}$$

Next, we have the inequality

$$\begin{aligned} \text{DP}(\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)) &= \text{DP}(\mathcal{T}^{[0,s-2]}) \times \text{DP}_\sigma(a^{[s-1]}, c) \\ &\leq \text{DP}(\mathcal{T}^{[0,s-2]}) \times \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max}. \end{aligned}$$

Consequently, the probability of $\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)$ is less than the s -rank bound. The result then is a consequence of Lemma 2.5. ■

2.3.3. Active S-Boxes in the Next Round

Throughout this part, the linear layer π is assumed to be a bit permutation. Denote by L_{ASB} the mapping from $(\mathbb{F}_2^n)^m$ to \mathbb{F}_2^m which maps a pattern c to the m -bit vector $L_{\text{ASB}}(c) = (x_i)_{i < m}$ where x_i is equal to one if and only if the bundle c_i is nonzero. In other words, $L_{\text{ASB}}(c)$ is a compact representation of the S-boxes activated by the pattern c and L_{ASB} should be read “*List of the Active S-Boxes*”.

Given two elements L and L' of \mathbb{F}_2^m , we denote by $L \vee L'$ their bitwise OR. Moreover, we say that two patterns c and c' seen as elements of \mathbb{F}_2^{nm} are disjoint if for all bit index $i \leq nm$, the equation $c_i = c'_i$ implies that $c_i = c'_i = 0$. It should be noted that if two disjoint patterns c and c' are seen as elements of $(\mathbb{F}_2^n)^m$, then $c_i = c'_i$ also implies that $c_i = c'_i = 0_n$ for each $i < m$. Let c be a pattern and $i < m$ be a nonnegative integer. By $b|_i$ we mean the element of $(\mathbb{F}_2^n)^m$ where all bundles are zero, except the one of index i which is equal to b_i . In other words,

$$(x_j)_{j < m} = b|_i \iff \begin{cases} x_i = b_i & \text{and} \\ x_j = 0_n & \text{if } j \neq i. \end{cases}$$

Before stating and proving the pruning condition involving the active S-boxes in the next round, we introduce two preliminary results.

Lemma 2.11. Let c^0, \dots, c^{w-1} be w pairwise mutually disjoint patterns. Then

$$L_{\text{ASB}}\left(\sum_{i=0}^{w-1} c^i\right) = \bigvee_{i=0}^{w-1} L_{\text{ASB}}(c^i).$$

Proof. The result is certainly true when $w = 1$, so assume that $w = 2$. Denote by L^0 , L^1 and L the lists of S-boxes activated by c^0 , c^1 and $(c^0 + c^1)$ respectively. Let $i < m$ be an integer. Next, we have the following equivalences

$$L_i = 0 \Leftrightarrow c_i^0 + c_i^1 = 0_n \Leftrightarrow c_i^0 = c_i^1 = 0_n \Leftrightarrow L_i^0 = L_i^1 = 0.$$

Therefore, $L = L^0 \vee L^1$. The result follows by induction on w as c^{w-1} and $(\sum_{i=0}^{w-2} c^i)$ are clearly mutually disjoint. ■

Corollary 2.12. Let b be an output pattern. Let $1 \leq w \leq m$ be an integer and let $0 \leq x_0 < \dots < x_{w-1} < m$ be w indices. Then,

$$L_{\text{ASB}}\left(\pi\left(\sum_{i=0}^{w-1} b|_{x_i}\right)\right) = \bigvee_{i=0}^{w-1} L_{\text{ASB}}(\pi(b|_{x_i})).$$

Proof. Since the diffusion layer π is linear, it holds that

$$\pi\left(\sum_{i=0}^{w-1} b|_{x_i}\right) = \sum_{i=0}^{w-1} \pi(b|_{x_i}).$$

Clearly, the patterns $b|_{x_i}$ are mutually disjoint. Since π is a bit permutation, it must be the case that the $\pi(b|_{x_i})$ are also disjoint. Finally, the relation

$$L_{\text{ASB}}\left(\sum_{i=0}^{w-1} \pi(b|_{x_i})\right) = \bigvee_{i=0}^{w-1} L_{\text{ASB}}(\pi(b|_{x_i}))$$

follows from Lemma 2.11, which concludes the proof. ■

Theorem 2.13. We use the same notations as in Theorem 2.10 except that $s < r-1$. Let w' denote the Hamming weight of $\bigvee_{i=0}^{v-1} L_{\text{ASB}}(\pi(b^{[s-1]}|_{x_i}))$. If

$$\left[\text{DP}(\mathcal{T}^{[0,s-2]}) \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max} \right] \times \text{DP}_{(w')}^{\max}$$

is less than the rank- $(s+1)$ bound, then for every output pattern c satisfying

$$c_{x_i} = b_{x_i}^{[s-1]} \text{ for each } i < v,$$

there does not exist any r -round trail extending $\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)$ with probability greater than or equal to $p_e^{(r)}$.

Proof. Following the proof of Theorem 2.10, we can assume that c is a candidate for $a^{[s-1]}$ and deduce the upper bound

$$\text{DP}(\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)) \leq \text{DP}(\mathcal{T}^{[0,s-2]}) \times \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max}.$$

Define $a^{[s]} = \pi(c)$. Let $b^{[s]}$ be any output pattern. Similarly, we can assume that $b^{[s]}$ is a candidate for $a^{[s]}$. Let w'' denote the bundle weight of $a^{[s]}$ which is clearly

equal to the Hamming weight of $L_{\text{ASB}}(a^{[s]})$. According to Corollary 2.12,

$$\begin{aligned} L_{\text{ASB}}(a^{[s]}) &= L_{\text{ASB}}(\pi(b^{[s-1]})) = L_{\text{ASB}}\left(\pi\left(\sum_{i=0}^{w-1} b^{[s-1]}|_{x_i}\right)\right) = \bigvee_{i=0}^{w-1} L_{\text{ASB}}(\pi(b^{[s-1]}|_{x_i})) \\ &= \left(\bigvee_{i=0}^{v-1} L_{\text{ASB}}(\pi(b^{[s-1]}|_{x_i}))\right) \vee \left(\bigvee_{i=v}^{w-1} L_{\text{ASB}}(\pi(b^{[s-1]}|_{x_i}))\right). \end{aligned}$$

As a consequence,

$$w'' = w(L_{\text{ASB}}(a^{[s]})) \leq w\left(\bigvee_{i=0}^{v-1} L_{\text{ASB}}(\pi(b^{[s-1]}|_{x_i}))\right) = w',$$

and thus $\text{DP}_{(w'')}^{\max} \geq \text{DP}_{(w')}^{\max}$. Eventually,

$$\begin{aligned} \text{DP}(\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c) \parallel (a^{[s]}, b^{[s]})) &= \text{DP}(\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c)) \times \text{DP}_{\sigma}(a^{[s]}, b^{[s]}) \\ &\leq \left[\text{DP}(\mathcal{T}^{[0,s-2]}) \times \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max} \right] \times \text{DP}_{(w'')}^{\max} \\ &\leq \left[\text{DP}(\mathcal{T}^{[0,s-2]}) \times \left(\prod_{i=0}^{v-1} \text{DP}_{S_{x_i}}(a_{x_i}^{[s-1]}, b_{x_i}^{[s-1]}) \right) \times \text{DP}_{(w-v)}^{\max} \right] \times \text{DP}_{(w')}^{\max}. \end{aligned}$$

Therefore, the probability of $\mathcal{T}^{[0,s-2]} \parallel (a^{[s-1]}, c) \parallel (a^{[s]}, b^{[s]})$ is less than the rank- $(s+1)$ bound and there exists no r -round trail extending it with probability greater than or equal to $p_e^{(r)}$. Using the fact that this property holds for all $b^{[s]}$, the desired result is proven. ■

The search procedure **Round** optimized with Theorems 2.10 and 2.13 is described in Figure 2.14.

2.3.4. Test on the Bound

All the previous results can be preserved while strengthening the condition on the bound. Suppose we have found a trail with probability greater than or equal to $p_e^{(r)}$. The estimation $p_e^{(r)}$ is then equal the differential probability of this trail. Now, assume that the probability of the current s -round trail \mathcal{T} satisfies $\text{DP}(\mathcal{T}) \cdot p_o^{(r-s)} = p_e^{(r)}$. In this case, the probability $\text{DP}(\mathcal{T})$ is not less than the rank- r bound and the algorithm tries all its possible extensions. However, the previous equality implies that in the best-case scenario, we find an r -round trail with probability $p_e^{(r)}$. Because such a trail is already known, the extension of \mathcal{T} can be aborted. This discussion proves that Definition 2.3 can be enhanced as follows.

Definition 2.14 (rank- s bound). Let \mathcal{T} be a s -round trail with $s < r$. Its probability is *less than the rank- s bound* if

$$\left(\mathcal{T}_o^{(r)} = () \text{ and } \text{DP}(\mathcal{T}) < \frac{p_e^{(r)}}{p_o^{(r-s)}} \right) \quad \text{or} \quad \left(\mathcal{T}_o^{(r)} \neq () \text{ and } \text{DP}(\mathcal{T}) \leq \frac{p_e^{(r)}}{p_o^{(r-s)}} \right).$$

Algorithm 3 – Round($s, \mathcal{T}^{(s-1)}, p^{(s-1)}$)

Input. $\mathcal{T} = ((a^{[0]}, b^{[0]}), \dots, (a^{[s-2]}, b^{[s-2]}))$

```

1   $a \leftarrow \mathcal{A}(a^{[s-2]})$ 
2   $b \leftarrow \mathcal{B}_{nm}(a)$ 
3   $p \leftarrow p^{(s-1)}$ 
4   $\mathcal{T}^{(s)} \leftarrow \mathcal{T}^{(s-1)} \parallel (a^{[s-1]}, b^{[s-1]})$ 
5   $w \leftarrow w_n(a^{[s-1]})$ 
6  Denote  $x_0 < \dots < x_{w-1}$  the indices of the S-boxes activated by  $a^{[s-1]}$ .
7   $X \leftarrow (x_0, \dots, x_{w-1})$ 
8   $L^{(0)} \leftarrow 0_m$ 
9  Call RoundRec( $s, 1, \mathcal{T}^{(s)}, p^{(s,0)}, L^{(0)}, X$ )

Function RoundRec( $s, v, \mathcal{T}^{(s)}, p^{(s,v-1)}, L^{(v-1)}, X$ )
10 If  $v = w$  then
11    $p^{(s)} \leftarrow p^{(s,w-1)}$ 
12   If  $s + 1 < r$  then
13     Call Round( $s + 1, \mathcal{T}^{(s)}, p^{(s)}$ )
14   Else
15     Call LastRound( $\mathcal{T}^{(s)}, p^{(s)}$ )
16 Else
17    $x \leftarrow x_{v-1}$ 
18   For each  $b_x^{[s-1]}$  sorted in decreasing order according to
19      $\text{DP}_{S_x}(a_x^{[s-1]}, \cdot)$  do
20      $p^{(s,v)} \leftarrow p^{(s,v-1)} \times \text{DP}_{S_x}(a_x^{[s-1]}, b_x^{[s-1]})$ 
21     If  $p^{(s,v)} \times \text{DP}_{(w-s)}^{\max}$  is less than the rank- $s$  bound then Theorem 2.10
22       Exit the loop
23     If  $\pi$  is a bit permutation then
24        $L^{(v)} \leftarrow L^{(v-1)} \vee L_{\text{ASB}}(\pi(b^{[s-1]}|_x))$ 
25        $w' \leftarrow w(L^{(v)})$ 
26       If  $p^{(s,v)} \times \text{DP}_{(w-s)}^{\max} \times \text{DP}_{(w')}^{\max}$  is not less than the rank- $(s+1)$ 
27         bound then Theorem 2.13
28         Call RoundRec( $s, v + 1, \mathcal{T}^{(s)}, p^{(s,v)}, L^{(v)}, X$ )
29       Else
30         Call RoundRec( $s, v + 1, \mathcal{T}^{(s)}, p^{(s,v)}, L^{(v)}, X$ )

```

Figure 2.14: Second optimization – the search function Round

Algorithm 4 – OptTrail($r, (p_o^{(i)})_{1 \leq i < r}$)

Input. The current number r of rounds and the probabilities $(p_o^{(i)})_{1 \leq i < r}$

Output. An optimal r -round trail $\mathcal{T}_o^{(r)}$

```

1   $\mathcal{T}_o^{(r)} \leftarrow ()$ 
2   $p_e^{(r)} \leftarrow p_o^{(r-1)}$ 
3  While  $\mathcal{T}_o^{(r)}$  is empty do
4     $p_e^{(r)} \leftarrow p_e^{(r)} / 2$ 
5     $\mathcal{T}_o^{(r)} \leftarrow \text{OptTrailEst}(r, (p_o^{(i)})_{1 \leq i < r}, p_e^{(r)})$ 
6  Return  $\mathcal{T}_o^{(r)}$ 
    
```

Figure 2.15: Automatic estimation management

2.3.5. Automatic Management of the Estimation

As explained in Section 2.1.1, the estimation $p_e^{(r)}$ determines the complexity of the algorithm **OptTrailEst**. Several methods yield good estimations of $p_o^{(r)}$. For instance, an iterative trail can be used. Following an idea of Ohta, Moriai and Aoki [87], let us introduce the algorithm **OptTrail**. The latter has two main advantages. Firstly, the estimation management is completely automatic, that is to say, no knowledge is required on the SPN. Secondly, its complexity has the same order of magnitude as **OptTrailEst** runs with $p_e^{(r)} = p_o^{(r)} / 2$.

The algorithm **OptTrail** is presented in Figure 2.15. To understand how it works, it is worth recalling that **OptTrailEst** finds no trail whenever $p_e^{(r)} > p_o^{(r)}$ as ensured by Theorem 2.6. In this case, the best trail $\mathcal{T}_o^{(r)}$ remains empty at the end of the execution of this algorithm. Since $p_o^{(r)} \leq p_o^{(r-1)}$, we begin by running **OptTrailEst** with the estimation $p_e^{(r)} = p_o^{(r-1)} / 2$. Then, this estimation is divided by two after each execution **OptTrailEst** until an optimal trail is found. This happens whenever the condition $p_e^{(r)} \leq p_o^{(r)}$ becomes true. It is not hard to see that we have the following proposition.

Proposition 2.15. The complexity of **OptTrailEst** decreases as the input $p_e^{(r)}$ increases.

In addition, we have observed experimentally that the complexity of the algorithm **OptTrailEst** executed with $p_e^{(r)} \geq 2^4 \cdot p_o^{(R)}$, is negligible compared to its complexity when running with $p_e^{(r)} = p_o^{(R)} / 2$. This discussion justifies that **OptTrail** has roughly the same complexity as **OptTrailEst**.

2.4. Results

Experiments and simulations have been performed by a *AMD Phenom II X4 965 Black Edition 3.4 GHz* processor. The running time for a R -round cipher includes the precomputations and $R - 1$ calls to **OptTrail**, as explained in Section 2.1.1.

To prove the practical security of PRESENT [17] against differential cryptanalysis, the authors have shown that the probability of any 5-round trail is upper-bounded by 2^{-20} and had exhibited a 5-round trail of probability 2^{-21} . The algorithm presented here allows us to prove in 0.3 second that this upper bound is met with equality. They have then deduced that any 25-round trail probability is upper-bounded by 2^{-100} . Our algorithm shows that the optimal trail probability is 2^{-110} in 0.5 second. The number of rounds is not a problem since an optimal 64-round trail is computed in just 2 seconds. Note that PRESENT has 32 rounds.

The permutation used in SMALLPRESENT [69] (and in PRESENT) can be generalized for all positive integers n and m . Denote by $\phi_{n,m}$ the permutation of $\llbracket 0, nm \rrbracket$ defined by the rule

$$\phi(i) = m(i \bmod n) + \left\lfloor \frac{i}{m} \right\rfloor.$$

We have constructed a 128-bit SPN on the same model as PRESENT to test our algorithm efficiency. Define π to be the bit permutation associated with $\phi_{8,16}$ and the S-boxes to be all equal to the AES S-box [39]. Using this algorithm, an optimal 13-round differential trail with probability 2^{-89} was obtained in 7.1 seconds.

To analyze PUFFIN security against differential cryptanalysis, Cheng et al [36] have upper-bounded the probability of an optimal 31-round trail by 2^{-62} . In 0.02 second, we have computed a trail meeting this bound with equality.

Finally, we have tested our algorithm on ICEBERG [95]. However, its diffusion layer is not a bit permutation so the optimization presented in Section 2.3.3 is no longer applicable. The authors have upper-bounded the probability of an optimal 16-round differential trail by 2^{-160} . We proved that it is in fact $2^{-171.6}$ in 2.3 seconds. All these results are outlined in Figure 2.16.

	Block size	Round number	Upper- bound	Best probability	Running time
PRESENT	64	5	2^{-20}	2^{-20}	0.3 s
PRESENT	64	25	2^{-100}	2^{-110}	0.5 s
PRESENT-like	128	13	—	2^{-89}	7.1 s
PUFFIN	64	31	2^{-62}	2^{-62}	0.02 s
ICEBERG	64	16	2^{-160}	$2^{-171.6}$	2.3 s

Figure 2.16: Summary of Results

To conclude, we have presented in this chapter a generic algorithm that computes an optimal differential or linear trail in an SPN. Running this algorithm may allow to prove the practical security of the block cipher. In the opposite case of a weak cipher, the returned trail gives rise to a powerful differential or linear cryptanalysis of the cipher. Especially optimized for SPN whose diffusion layer is a bit permutation, we are able to find an optimal differential trail of PRESENT and PUFFIN within one second. Block cipher designers have then a powerful tool which can be run several times in order to improve their cipher primitives.

Partition-Based Backdoor Ciphers

One of the first backdoor ciphers was created in 1997 by Rijmen and Preneel [89]. Their S-boxes are constructed to have one high correlation between the zero mapping and a sum of certain output bits. The knowledge of this correlation yields a high potential linear trail which is used to recover a part of the key with linear cryptanalysis. Such a weakness is generally pointed out by the first line of the S-boxes' correlation matrices. Yet, if the output size of the S-boxes is large enough, their computation is too expensive. Relying on this fact, the authors claimed that their backdoor is undetectable, even if one knows its global design. Nevertheless, Wu and al. [100] disproved this by discovering a way to recover the backdoor. It is worthwhile to mention that in practice, if a real cipher containing a backdoor is given, the presence of the backdoor will certainly not be revealed.

More recently in [2], the authors created non-surjective S-boxes embedding a parity check to create a backdoor cipher. The message space is thus divided into cosets and leads to create an attack on this DES-like cipher in less than 2^{23} operations. The security of the whole algorithm, particularly against linear and differential cryptanalysis is not given and the authors admit that their attack is dependent on the first and last permutation of the cipher. Finally, the non-surjective S-boxes may lead to detect easily the backdoor by simply calculating the image of each input vector. This problem is naturally avoided in a Substitution-Permutation Network in which S-boxes are bijective by definition.

Our approach is mainly a generalization of the ideas presented by Paterson in [88]. In this article, a DES-like backdoor cipher exploiting a weakness induced by the round functions is presented. The group generated by the round functions acts imprimitively on the message space. In other words, the round function preserves a partition of the message space no matter the round keys used, and hence the same applies to the full cipher. This partition forms the backdoor. Paterson then introduced a backdoor cipher composed of 32 rounds and using an 80-bit cipher key. The backdoor can seriously compromise the cipher security using 2^{32} chosen plaintexts. Moreover, when combined with a carefully chosen key schedule, the backdoor enables recovery of the key using 2^{41} operations and a few known plaintexts. Even if the mathematical material to build the backdoor is given, no general algorithm details the S-boxes' construction. As the author acknowledges, the S-boxes of his backdoor cipher are incomplete: half of the ciphertext bits are independent of half of the plaintext bits and the security against a differential attack is *not as high*

as one might expect. Moreover, the author wondered whether the partition of the message space had to be linear, that is to say made up with every coset of a linear subspace. Caranti and al. [31] answered Paterson’s question by proving that if the group generated by the round functions is imprimitive, then the partition of the message space must be linear.

In his thesis [50], Harpes considered backdoor ciphers mapping a partition of the plaintexts to a partition of the ciphertexts independently of the cipher key used. As these partitions are not necessarily equal, this family generalizes Paterson’s one. These ciphers are called *partition-based backdoor ciphers* throughout this thesis. When the input and output partitions are equal, we speak of *imprimitve backdoor ciphers* to fit Paterson’s work. More generally, a *probabilistic partition-based backdoor cipher* is a cipher which behaves like a partition-based cipher with high probability. Harpes suggested using such a backdoor with its partitioning cryptanalysis [52] to recover some bits of the cipher key using known or chosen plaintext/ciphertext pairs.

Along a similar line to Paterson’s imprimitive ciphers, the group generated by the round functions has required much attention. This group was first studied by Coppersmith and Grossman in [38]. Then, Kaliski et al. asked whether the DES is a group and provided strong evidence that it is not the case [56]. This group was proved to be the alternating group later in [97]. The next standard block cipher, namely the AES, was proven to generate also the alternating group in [94, 98]. Even if a secure cipher must generate a large group, it has been shown that this condition is not sufficient in [78]. Indeed, the authors described a very weak block cipher generating the symmetric group. More recently, Caranti et al. [31] introduced a class of block ciphers for which it is easier to prove that the group generated by the round functions is primitive. To demonstrate the efficiency of their framework, they applied it to the AES. Their results were then improved in [30, 32, 5, 4] and can then be used to prove that this group is either the alternating group or the symmetric group. Finally, we should mention another active area of research about backdoor ciphers which considers the so-called *hidden sum* [24, 25, 19]. This family of backdoor ciphers relies on an alternative vector space structure which can be used to break the cipher.

The backdoor ciphers covered by this thesis, namely imprimitive and partition-based ciphers with their probabilistic variants, are introduced formally in the next section. We also recall several ways to exploit the backdoors of imprimitive ciphers but these attacks can easily be extended to partition-based backdoors. A cryptanalysis of a probabilistic backdoor cipher will be detailed later in Chapter 5.

The remainder of this chapter focuses only on non-probabilistic partition-based backdoor substitution-permutation networks. More precisely, we study the structure of such ciphers when the backdoor holds no matter the round keys used, that is to say independently of the key schedule. We explore in Section 3.2 how the partition of the message space evolves through each step of the encryption process and prove that the study of the whole cipher can be reduced to the study of its substitution layer. Then, we spend quite a bit of time in Section 3.3 showing that this study can be restricted further to that of one single S-box. Lastly, our results are summarized

in Section 3.4 which concludes this chapter. The content of this chapter was first published in [9] and then developed in [12].

3.1. Partition-Based Backdoor Ciphers

This section introduces every family of backdoor ciphers covered by this thesis. To detail these backdoors, we may recall some classical results and definitions in Section 3.1.1. Readers acquainted with basic facts on imprimitive groups may jump immediately to Section 3.1.2 which presents imprimitive backdoor ciphers. Then Section 3.1.3 recalls how to take advantage of such a backdoor while Section 3.1.4 deals with its generalizations. To conclude its introduction, other closely related attacks are given in Section 3.1.5.

3.1.1. Imprimitive Group Actions

The *symmetric group* on X , denoted by $\text{Sym}(X)$, is the set of all permutations of X together with the operation of composition. Even if it is quite common to define the composition of two permutations σ, τ in $\text{Sym}(X)$ by $\sigma \circ \tau : x \mapsto \tau(\sigma(x))$ when studying permutation groups, we will keep the convention that $\sigma \circ \tau(x) = \sigma(\tau(x))$ throughout this and the following chapters. In other words, permutations are still evaluated from right to left in a composition.

Definition 3.1 (Group Action). Let G be a group and let X be a set. A *(left) group action* is a mapping $G \times X \rightarrow X$, $(g, x) \mapsto g \cdot x$ such that the following statements hold:

- $e \cdot x = x$ for any x in X , (e denotes the identity element of G);
- $g \cdot (h \cdot x) = (gh) \cdot x$ for all g, h in G and all x in X .

Alternatively, a group action can be defined as a group homomorphism ϕ from G to the symmetric group $\text{Sym}(X)$.

Let us explain the equivalence between these two definitions of a group action. Let G be a group acting on X . Define the mapping ϕ from G to $\text{Sym}(X)$ which maps an element g of G to the permutation

$$\phi_g : X \rightarrow X, \quad x \mapsto g \cdot x.$$

Let g be an element of G . It is easily seen that $\phi_{g^{-1}} \circ \phi_g = \phi_g \circ \phi_{g^{-1}} = \text{Id}_X$ and hence ϕ_g is a permutation of X , ensuring that ϕ is well-defined. It remains to prove that ϕ is a homomorphism. Let h be an element of G . For any x in X , it holds that

$$(\phi_g \circ \phi_h)(x) = \phi_g(\phi_h(x)) = \phi_g(h \cdot x) = g \cdot (h \cdot x) = (gh) \cdot x = \phi_{gh}(x).$$

Thus, the action of G on X yields a homomorphism ϕ from G to $\text{Sym}(X)$.

Conversely, let ϕ be a homomorphism from a group G to the symmetric group on a set X . Define the mapping \cdot from $G \times X$ to X by the rule $g \cdot x = \phi_g(x)$. Let g

and h be two elements of G . Then, for every x in X , we have

$$\begin{aligned} e \cdot x &= \phi_e(x) = \text{Id}_X(x) = x \quad \text{and} \\ g \cdot (h \cdot x) &= \phi_g(\phi_h(x)) = (\phi_g \circ \phi_h)(x) = \phi_{gh}(x) = (gh) \cdot x. \end{aligned}$$

This discussion establishes the equivalence between the two definitions.

A *permutation group* on X is a subgroup of $\text{Sym}(X)$. Permutation groups are closely tied to group actions. Indeed, a permutation group G on X naturally acts on X by $g \cdot x = g(x)$ for all g in G and all x in X . In this case, the corresponding homomorphism from G to $\text{Sym}(X)$ is simply the inclusion mapping.

Inversely, let G be a group acting on X and let ϕ denote the corresponding homomorphism. Then, the image $\phi(G)$ is a permutation group on X called the permutation group *induced* on X by G . Moreover, if ϕ is one-to-one, G is isomorphic to $\phi(G)$ and the action of G on X is said to be *faithful*. In such a case, the notions of permutation groups and group actions are the same.

Before defining imprimitive group actions, we need to introduce the following two definitions.

Definition 3.2 (Transitivity). The action of a group G on a set X is said to be *transitive* if for all x_1 and x_2 in X , there exists an element g of G such that $g \cdot x_1 = x_2$.

Definition 3.3 (G-invariant Partition). Let G be a group acting on a set X . A partition \mathcal{B} of X is said to be a *G-invariant partition* (or a *block system of G*) if every element g of G preserves \mathcal{B} , that is to say, if $\mathcal{B} = \{g \cdot B \mid B \in \mathcal{B}\}$ where $g \cdot B$ denotes the set $\{g \cdot x \mid x \in B\}$.

Any group G acting on a set X has at least two G -invariant partitions, namely $\mathcal{B} = \{X\}$ and $\mathcal{B} = \{\{x\} \mid x \in X\}$. These partitions are said to be *trivial*.

Definition 3.4 (Imprimitivity). Let G be a group acting transitively on X . The action of G on X is said to be *imprimitive* if there exists a non-trivial G -invariant partition of X . Otherwise, the group is said to act *primitively*.

A permutation group G on X is naturally said to be *imprimitive* when its induced action on X is imprimitive. Moreover, it should be noted that any subgroup of an imprimitive permutation group is also imprimitive.

Lemma 3.5. Let G be a group acting imprimitively on a set X and let \mathcal{B} be a non-trivial G -invariant partition. For all parts B_1 and B_2 in \mathcal{B} , there exists g in G such that $g \cdot B_1 = B_2$.

Proof. Let B_1 and B_2 be two parts of \mathcal{B} . Let x_1 and x_2 be elements of B_1 and B_2 respectively. As the action of G on X is transitive, there exists an element g of G such that $g \cdot x_1 = x_2$. Thus, x_2 belongs to both $g \cdot B_1$ and B_2 . Note that $g \cdot B_1$ is a part of \mathcal{B} because \mathcal{B} is G -invariant. It follows that $g \cdot B_1 = B_2$ since these two parts have a non-empty intersection. ■

Remark 3.6. Any part B of a G -invariant partition \mathcal{B} is called a *block*. Generally, permutation group books [42, 55, 99] deal with blocks rather than G -invariant partitions because such partitions are uniquely determined by any of their blocks. Indeed, Lemma 3.5 implies that $\mathcal{B} = \{g \cdot B \mid g \in G\}$. Alternatively, a block B of G can be defined to be a non-empty subset of X such that for every g in G , the subsets $g \cdot B$ and B are either disjoint or equal. For this reason, it is common to define an imprimitive group action to be a transitive action which has a non-trivial block. Finally, note that G -invariant partitions are called *block systems* in [6, 42, 55], *imprimitive systems* in [90] and *complete block systems* in [88, 99].

Let G be a group acting on X and assume that \mathcal{B} is a G -invariant partition. Denote by B a fixed block in \mathcal{B} . Given any block B' , Lemma 3.5 ensures that there exists an element g of G such that $g \cdot B = B'$. Denoting by ϕ the homomorphism associated with the action of G on X , we know that the mapping $\phi_g : x \mapsto g \cdot x$ is a permutation of X . As a consequence, B and B' have the same cardinality. This discussion proves the following corollary.

Corollary 3.7. Let G be a group acting imprimitively on a set X and let \mathcal{B} be a non-trivial G -invariant partition. Every part of \mathcal{B} have the same cardinality. Assuming that X is finite, we have the relation $\#X = \#B \times \#\mathcal{B}$ where B is any block of \mathcal{B} .

In mathematics, we generally ties together different objects which are similar when considering their structures. These similar objects are then said to be *isomorphic*. Let G and H be two groups acting respectively on X and Y . For these actions to be isomorphic, the groups G and H must have the same structure, namely being isomorphic as groups. However, this condition cannot be sufficient since it disregards how the groups G and H act on their respective sets. For instance, the group $\text{Sym}(\{1, 2\})$ acts naturally on $\{1, 2\}$ but can also act trivially on $\{1, 2\}$ by always fixing every element. These two actions are very different while they have the same group and set. For this reason, we introduce the following stronger notion.

Definition 3.8 (Permutation Isomorphism). Let G act on X and let H act on Y . The action of G on X is *permutation isomorphic* to H on Y if there exists an isomorphism $\varphi : G \rightarrow H$ and a bijection $\lambda : X \rightarrow Y$ satisfying for every g in G and every x in X the relation

$$\lambda(g \cdot x) = \varphi(g) \cdot \lambda(x).$$

Remark 3.9. The condition that the relation $\lambda(g \cdot x) = \varphi(g) \cdot \lambda(x)$ holds for every x in X is equivalent to saying that the diagram in Figure 3.1 commutes. The terminology *permutation isomorphism* is also used in [42, pp. 17] and [6]. However, the same notion is called a *G -space isomorphism* in [26, pp. 6], and more simply an *isomorphism* in [90, pp. 282].

Assume that the action of G on X is permutation isomorphic to H on Y . Then, this action is uniquely determined by the other action. Indeed, for any g in G and any x in X , we have

$$g \cdot x = \lambda^{-1}(\varphi(g) \cdot \lambda(x)).$$

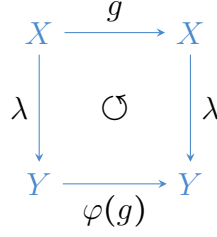


Figure 3.1: Diagrammatic representation of a permutation isomorphism (see Definition 3.8). Note that g denotes the mapping $X \rightarrow X$, $x \mapsto g \cdot x$ and similarly, $\varphi(g)$ denotes $Y \rightarrow Y$, $y \mapsto \varphi(g) \cdot y$.

Let \mathcal{B} be any G -invariant partition. For every h in H , we have

$$h \cdot \lambda(\mathcal{B}) = \varphi(\varphi^{-1}(h)) \cdot \lambda(\mathcal{B}) = \lambda(\varphi^{-1}(h) \cdot \mathcal{B}) = \lambda(\mathcal{B})$$

Thus, $\lambda(\mathcal{B})$ is a H -invariant partition. Consequently, if the action of G on X is imprimitive, then so is the action of H on Y .

3.1.2. Imprimitive Backdoor ciphers

Backdoors based on imprimitive permutation groups were introduced by Paterson in [88]. We restate here their theoretical framework using our notations. This family of backdoor ciphers belongs to the class of iterated block ciphers. We may recall that the encryption process of an iterated block cipher (see Definition 1.4) consists of the composition of round functions applied to the plaintext with different round keys. More formally, the message, cipher key and round key spaces are respectively \mathbb{F}_2^n , \mathbb{F}_2^κ and \mathbb{F}_2^l , where n is the block size, κ the cipher key length and l the round key length. The round function is a family $(F_k)_{k \in \mathbb{F}_2^l}$ of keyed permutations of the message space \mathbb{F}_2^n . Then, the r -round encryption function E associated with the round keys $k^{[0]}, \dots, k^{[r-1]}$ is given by

$$E_{k^{[0]}, \dots, k^{[r-1]}} = F_{k^{[r-1]}} \circ \dots \circ F_{k^{[0]}}.$$

In practice, the round keys $(k^{[i]})_{i < r}$ are derived from a cipher key K using a key schedule. Nevertheless, the key schedule is disregarded by the main framework of imprimitive backdoor ciphers, which considers only independent round keys. As will be seen at the end of Chapter 4, a carefully designed key schedule can remarkably improve the backdoor, but for now it is simpler to ignore this part of the cipher.

By the *group generated by the round functions*, we mean the subgroup G of $\text{Sym}(\mathbb{F}_2^n)$ defined to be

$$G = \langle F_k \mid k \in \mathbb{F}_2^l \rangle.$$

Being a permutation group on \mathbb{F}_2^n , it naturally acts on the message space by the rule $g \cdot x = g(x)$ for every g in G and x in \mathbb{F}_2^n . Even if we will only consider this group in the remainder of this section, we may introduce two other similar groups to

stress its relevance. First, let G_r be the group generated by the r -round encryption functions with independent round keys, namely

$$G_r = \langle E_{k^{[0]}, \dots, k^{[r-1]}} \mid k^{[0]}, \dots, k^{[r-1]} \in \mathbb{F}_2^l \rangle.$$

Then G_{cipher} is defined in the same way as G_r except that the round keys are derived from all possible cipher keys using the cipher's key schedule. In other words, G_{cipher} is generated by all the encryption functions of the cipher. In [54, Lemma 1], it has been proven that G_{cipher} is a subgroup of G_r , itself being a subgroup of G . As a consequence, if G is an imprimitive permutation group, then so are G_r and G_{cipher} .

An *imprimitive backdoor cipher* is an iterated block cipher whose permutation group G generated by its round functions is imprimitive. Naturally, the cipher's designer has to be aware of this property, otherwise we should not talk about backdoor. Now, suppose that G is an imprimitive permutation group on \mathbb{F}_2^n . Then, there exists by definition a non-trivial G -invariant partition \mathcal{B} of \mathbb{F}_2^n . Corollary 3.7 establishes that the number of blocks in \mathcal{B} divides the cardinality of \mathbb{F}_2^n , that is 2^n . Thus, the partition \mathcal{B} contains 2^d blocks with $1 < d < n$ and we can write

$$\mathcal{B} = \{B_0, \dots, B_{2^d-1}\}.$$

Furthermore, each block B_i has cardinality 2^{n-d} . Let g be an element of G . Since the partition \mathcal{B} is G -invariant, the image of any block under g is still a block. Equivalently, the permutation g of \mathbb{F}_2^n induces a permutation \bar{g} of $\llbracket 0, 2^d \rrbracket$ which maps i to the unique index j satisfying $g(B_i) = B_j$. Using this notation, a block B_i is mapped to $B_{\bar{g}(i)}$ by g .

3.1.3. Exploiting the backdoor

There are several ways to take advantage of this backdoor. We begin with the most basic, but also the one which works for every imprimitive backdoor cipher. As explained above, G_{cipher} is a subgroup of G , so every encryption function lies in G . Let K be a cipher key and let g denote its associated encryption function E_K . This means that when several plaintexts lying in a same block B_i are encrypted with g , their corresponding ciphertexts lie in the same block $B_{\bar{g}(i)}$. Such a property can be used by the following chosen-plaintext attack. For each index $0 \leq i < 2^d$, choose a plaintext p_i in B_i and request their corresponding ciphertexts c_i . With those data, we can recover the induced permutation \bar{g} . Indeed, the image of any index i under \bar{g} is the index of the block containing c_i .

Now, assume that we are given a ciphertext c whose corresponding plaintext is unknown. First, we must find the index j of the block B_j containing c . Next, we know that the plaintext lies in the block $B_{\bar{g}^{-1}(j)}$, that is to say, in a subset of size 2^{n-d} . If we know that p is a meaningful message, our uncertainty on p can be further restricted by canceling the meaningless messages in $B_{\bar{g}^{-1}(j)}$. If d is large, typically when $\frac{n}{2} < d < n$, then this cryptanalysis requires a huge amount of chosen plaintexts but also gives precise information on the plaintext. Similarly, if d is small, this attack gives little information on the plaintext but only needs a few chosen plaintexts.

Relying only on this basic cryptanalysis, the cipher designer needs to choose between quantity of data required and efficiency of its backdoor. However, such a choice can be avoided if the imprimitive cipher has several non-trivial G -invariant partitions. Assume that

$$\mathcal{A} = \{A_0, \dots, A_{2^a-1}\} \quad \text{and} \quad \mathcal{B} = \{B_0, \dots, B_{2^b-1}\}$$

are two G -invariant partitions. Up to a rearrangement of the blocks, it can be assumed that 0_n lies in both A_0 and B_0 . It is well-known that a non-empty intersection of two blocks of G is still a block (see for instance [99, Proposition 6.1]). Then, $A_0 \cap B_0$ is a block and

$$\mathcal{A} \cap \mathcal{B} = \{g(A_0 \cap B_0) \mid g \in G\} = \{A \cap B \mid A \in \mathcal{A}, B \in \mathcal{B}\} \setminus \{\emptyset\}$$

is a (possibly trivial) G -invariant partition. This result will be generalized later in Proposition 3.20. Denote by 2^d the number of elements in the intersection $A_0 \cap B_0$. Then, we know that each block of $\mathcal{A} \cap \mathcal{B}$ has also cardinality 2^d .

The basic cryptanalysis detailed above can be enhanced as follows. Again, denote by g the encryption function associated with the unknown cipher key. Using 2^a chosen plaintexts and their corresponding ciphertexts, recover the permutation \bar{g}_A induced by g on the partition \mathcal{A} . Similarly, recover the permutation \bar{g}_B induced on \mathcal{B} with 2^b other chosen plaintexts. Next, given a ciphertext c , find the indices j_A and j_B such that c lies in both A_{j_A} and B_{j_B} . Finally, the plaintext corresponding to c lies in

$$A_{i_A} \cap B_{i_B} \quad \text{where} \quad i_A = (\bar{g}_A)^{-1}(j_A) \text{ and } i_B = (\bar{g}_B)^{-1}(j_B).$$

To summarize, this cryptanalysis requires $2^a + 2^b \leq 2^{\max(a,b)+1}$ chosen plaintexts and yields 2^d possible plaintexts for each ciphertext.

Maybe the most interesting set of parameters for this cryptanalysis are $a = b = \frac{n}{2}$ and $d = 0$. In this case, the partitions \mathcal{A} and \mathcal{B} consist of $2^{\frac{n}{2}}$ blocks, each of cardinality $2^{\frac{n}{2}}$. The intersection $A_0 \cap B_0$ contains only $2^d = 1$ element, so the partition $\mathcal{A} \cap \mathcal{B}$ is equal to $\{\{x\} \mid x \in \mathbb{F}_2^n\}$. Once the permutations \bar{g}_A and \bar{g}_B have been recovered with $2^{\frac{n}{2}+1}$ chosen plaintexts, the cryptanalyst can decrypt any ciphertexts. In other words, the cryptanalyst has an alternative decryption algorithm as he does not recover the cipher key. Using the common vocabulary introduced in [62, 63], this attack performs a *global deduction*. Intuitively, the messages in \mathbb{F}_2^n are arranged in a $2^{\frac{n}{2}} \times 2^{\frac{n}{2}}$ matrix. Each block A of \mathcal{A} represents a row and each block B of \mathcal{B} a column of this matrix. The mappings \bar{g}_A and \bar{g}_B describe how g permutes the rows and columns respectively. Given a ciphertext, this attack recovers the row and the column of the plaintext, and hence the plaintext itself. An example of such a backdoor cipher is given in [88, Section 3.3].

Finally, let us explain the key schedule dependent attack outlined by Paterson in his article. Even if this attack can be generalized using several G -invariant partitions, we consider hereinafter only one G -invariant partition \mathcal{B} for simplicity. The main idea is to design a key schedule such that every induced permutation $\overline{E_K}$ of \mathcal{B} is

uniquely determined by a part of the cipher key K . Equivalently, we require the existence of a (non-trivial) partition $\mathcal{K} = \{I_0, \dots, I_{m-1}\}$ of \mathbb{F}_2^κ such that all the cipher keys belonging to a same part of \mathcal{K} induce the same permutation of \mathcal{B} , that is to say

$$\forall K, K' \in \mathbb{F}_2^\kappa, \quad (\exists I \in \mathcal{K}, K \in I \text{ and } K' \in I) \implies (\overline{E_K} = \overline{E_{K'}}).$$

Such a property can be used to carry out a key recovery attack. Let K be an unknown cipher key. Assume that the cryptanalyst has a few plaintext/ciphertext pairs (p_i, c_i) . It is worthwhile to mention that this attack does not require a lot of data, only two or three pairs could be sufficient. Denote by $[x]$ the block of \mathcal{B} containing the message x in \mathbb{F}_2^n . Then, proceed as follows.

- For each class I in \mathcal{K} , choose a cipher key \tilde{K} in I and test whether the equalities $\overline{E_{\tilde{K}}}([p_i]) = [c_i]$ hold for all pairs (p_i, c_i) . Observe that $\overline{E_{\tilde{K}}}([p_i]) = [c_i]$ holds if and only if $E_{\tilde{K}}(p_i)$ lies in the same block as c_i . This equivalent statement is more convenient for a real implementation.
- Then, for each candidate class I , check for every cipher key \tilde{K} in I if $E_{\tilde{K}}(p_i) = c_i$ hold for all pairs (p_i, c_i) .

Although this cryptanalysis was sketched by Paterson, no real example was given in his paper. In [9, Section 6], we introduced a toy imprimitive backdoor cipher vulnerable to this key schedule cryptanalysis. The cipher key space is divided into $2^{\frac{\kappa}{2}}$ classes, each containing $2^{\frac{\kappa}{2}}$ keys. When this attack is performed with two plaintext/ciphertext pairs, the first step requires at most $2 \times 2^{\frac{\kappa}{2}}$ encryptions. Generally, only one candidate class has to be tested in the second step, thereby requiring at most $2 \times 2^{\frac{\kappa}{2}}$ encryptions. Thus, the average-case complexity of this attack is $\mathcal{O}(2^{\frac{\kappa}{2}})$, compared with the exhaustive search which requires 2^κ encryptions. In Section 4.3, we will detail a toy backdoor cipher combining several G -invariant partitions with a key schedule dependent cryptanalysis.

3.1.4. Generalizations

Now we turn our attention to generalizations of imprimitive backdoor ciphers proposed by Harpes in his thesis [50]. So far, we have considered backdoor ciphers preserving a partition \mathcal{B} of the message space. More generally, a *Partition-Based Backdoor Cipher* is a cipher mapping a partition of the plaintext space to a partition of the ciphertext space, no matter the cipher key used. An imprimitive cipher is then a partition-based cipher whose input and output partitions are equal. More formally, we introduce the following definition.

Definition 3.10 (Partition-Based Backdoor Cipher). An iterated n -bit block cipher E is called a *partition-based backdoor cipher* if there exist two partitions \mathcal{A} and \mathcal{B} of \mathbb{F}_2^n such that for every cipher key K in \mathbb{F}_2^κ the following relationship holds:

$$\{E_K(A) \mid A \in \mathcal{A}\} = \mathcal{B}.$$

Since E_K must be a permutation of \mathbb{F}_2^n to allow decryption, it is easily seen that the partitions \mathcal{A} and \mathcal{B} necessarily have the same number of parts. Such backdoor

ciphers are the focus of this and the next chapter. The toy backdoor cipher given at the end of Chapter 4 will also illustrate this generalization.

Partitioning Cryptanalysis is an attack on iterated block ciphers introduced by Harpes in [52]. As differential cryptanalysis uses a pair (a, b) of difference patterns, partitioning cryptanalysis considers a pair of partitions $(\mathcal{A}, \mathcal{B})$, where \mathcal{A} is a partition of the plaintexts and \mathcal{B} a partition of the set of inputs of the last round. A pair $(\mathcal{A}, \mathcal{B})$ is *effective* if for almost all cipher keys, the inputs of the last round function are non-uniformly distributed over the blocks of \mathcal{B} when the plaintexts are uniformly chosen among one fixed block A of \mathcal{A} . Then, the attack exploits this non-uniform behavior to recover information on the last round key, in the same way as linear and differential cryptanalysis do.

In the light of this attack, we should relax the definition of partition-based backdoor ciphers to include any iterated cipher designed to be vulnerable to partitioning cryptanalysis. To avoid confusion, we suggest the following definition.

Definition 3.11 (Probabilistic Partition-Based Backdoor Cipher).

An r -round iterative block cipher $E : \mathbb{F}_2^\kappa \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ is said to be a *probabilistic partition-based backdoor cipher* if there exists a pair $(\mathcal{A}, \mathcal{B})$ of partitions of \mathbb{F}_2^n satisfying the following property: for almost all cipher keys K in \mathbb{F}_2^κ and for each part A of \mathcal{A} , there exists a part $B_{A,K}$ of \mathcal{B} such that for every other part B it holds that

$$\mathbb{P}_{x \in A}(E_K^{(r-1)}(x) \in B_{A,K}) \gg \mathbb{P}_{x \in A}(E_K^{(r-1)}(x) \in B).$$

In other words, for almost all cipher keys K in \mathbb{F}_2^κ and for each part A of \mathcal{A} , the $(r-1)$ -round encryption function $E_K^{(r-1)}$ maps a significant proportion of the plaintexts lying in A to a part $B_{A,K}$ of \mathcal{B} and the remaining plaintexts in A should be spread over the other parts of \mathcal{B} . Again, this property must be intended by the designer to call it a backdoor. Chapter 5 will be dedicated to BEA-1, our backdoor cipher inspired by Paterson and Harpes' work and by the theory developed in this and the next chapters.

3.1.5. Links With Other Attacks

Before addressing the formal treatment of partition-based backdoor ciphers, we may digress a little from backdoors and expose a cryptanalysis closely related to our topic. In [70], Leander et al. developed a new cryptanalysis, called *invariant subspace attack*, breaking the PRINTCIPHER [61] for a significant fraction of its keys. Its efficiency has then been proven on several ciphers [21, 49, 71]. The general idea of this attack can be outlined as follows. Let F denote the SP-layer of a Substitution-Permutation network, that is, the round function without the key addition. Then, assume that F maps a coset of a given subspace V to another coset of V . In other words, there exist a and b such that $F(a + V) = b + V$. Here, the addition is made in \mathbb{F}_2^n , and hence corresponds with the XOR operation. The round function associated with the round key k is then defined by $F_k : x \mapsto F(x + k)$. If the round key k belongs

to the coset $a + b + V$, then it holds that

$$F_k(b + V) = F(b + k + V) = F(a + V) = b + V,$$

hence the name of *invariant subspace*. Therefore, if every round key lies in this particular coset, the affine subspace $b + V$ is preserved by the full encryption process. Such a property enables a very efficient distinguisher. As additional results, they also showed that the invariant subspace attack

- implies a truncated differential attack to be possible (the probability of the truncated differential characteristic is however highly key-dependent);
- implies the existence of strongly biased linear approximations for weak keys (independently of the number of rounds).

This attack was generalized in 2015 by Leander, Minaud and Rønjom [71]. They proposed a generic algorithm that is able to detect invariant subspaces. Indeed, their initial invariant subspaces on PRINTCIPHER were found empirically.

Following the idea of the invariant subspace attack, Grassi et al. [47, 48] introduced the *subspace trail cryptanalysis*. Given $r + 1$ subspaces $V^{[0]}, \dots, V^{[r]}$, it is assumed that the image of any coset of $V^{[i]}$ under the SP-network is included in a coset of $V^{[i+1]}$. That is to say, for each $a^{[i]}$, there exists $a^{[i+1]}$ such the following inclusion holds

$$F(a^{[i]} + V^{[i]}) \subseteq a^{[i+1]} + V^{[i+1]}.$$

In this case, it is easy to see the all the round functions F_k inherit such a property. The family of subspaces $(V^{[i]})_{i \leq r}$ is said to be a *subspace trail*. Naturally, the dimension of $V^{[i]}$ must be less than or equal to the dimension of $V^{[i+1]}$. In contrast to the invariant subspace attack, Grassi et al. relaxed the assumption that the coset has to be invariant. Here, the considered subset becomes the coset of possibly different increasingly dimensional subspaces throughout the encryption. However, the authors also required this property to hold for each coset of $V^{[0]}$ instead of one. Therefore, this cryptanalysis is not a generalization, but a variation of the invariant subspace attack. As will become clear in the next section, the family of backdoors covered in this thesis is closely related to constant-dimensional subspace trails.

3.2. Substitution-Permutation Networks and Partitions

This section aims at studying an SPN which maps a partition of the plaintexts to a partition of the ciphertexts. When the cipher key K is fixed, the encryption function E_K is just a permutation of the message space. Therefore, any partition \mathcal{A} of the plaintexts is mapped to the partition $E_K(\mathcal{A})$ of the ciphertexts. Nonetheless, to exploit the backdoor, the designer needs to know the pair of partitions $(\mathcal{A}, E_K(\mathcal{A}))$. The problem is that the output partition $E_K(\mathcal{A})$ depends *a priori* on the cipher key K , which is unknown to the attacker. The simplest way to solve this problem is to require that the partitions $E_K(\mathcal{A})$ are independent of the cipher keys K . In other words, we want all the partitions $E_K(\mathcal{A})$ to be equal to a fixed partition \mathcal{B} .

As with differential and linear cryptanalysis, taking account of the exact effect of the key schedule seems to be a challenging problem. Therefore, the key schedule will deliberately be omitted throughout this chapter. This amounts to consider an SPN mapping a partition \mathcal{A} to a fixed partition \mathcal{B} , independently of the round keys used. In the following subsection, we introduce some definitions and preliminary results.

3.2.1. Linear Partitions

Since we are concerned with ciphers which associate a partition of the ciphertext space to another partition of the plaintext space, let us introduce the following definition.

Definition 3.12. Let f be a permutation of E and \mathcal{A}, \mathcal{B} be two partitions of E . Let $f(\mathcal{A})$ denote the set $\{f(A) \mid A \in \mathcal{A}\}$. We say that f maps \mathcal{A} to \mathcal{B} if $f(\mathcal{A}) = \mathcal{B}$. If $\mathcal{A} = \mathcal{B}$, we say that f *preserves* the partition \mathcal{A} .

The two partitions $\{\{x\} \mid x \in E\}$ and $\{E\}$ are called the *trivial partitions* of E . Observe that, for any permutation f of E ,

$$f(\{\{x\} \mid x \in E\}) = \{\{x\} \mid x \in E\} \quad \text{and} \quad f(\{E\}) = \{E\}.$$

That is, every permutation preserves the two trivial partitions. Moreover it should be highlighted that if f maps \mathcal{A} to \mathcal{B} and if \mathcal{A} is non-trivial, then so is \mathcal{B} .

Example 3.13. Let E denote the set $\llbracket 0, 8 \rrbracket$ and consider the two partitions \mathcal{A}, \mathcal{B} of E defined to be $\mathcal{A} = \{\{0, 1, 4\}, \{2, 6\}, \{3, 7\}, \{5\}\}$ and $\mathcal{B} = \{\{0, 2, 7\}, \{1\}, \{3, 5\}, \{4, 6\}\}$. Let f be the permutation of E defined as follows:

$$0 \mapsto 7, \quad 1 \mapsto 0, \quad 2 \mapsto 3, \quad 3 \mapsto 6, \quad 4 \mapsto 2, \quad 5 \mapsto 1, \quad 6 \mapsto 5, \quad 7 \mapsto 4.$$

By definition,

$$\begin{aligned} f(\mathcal{A}) &= \{f(A) \mid A \in \mathcal{A}\} = \{f(\{0, 1, 4\}), f(\{2, 6\}), f(\{3, 7\}), f(\{5\})\} \\ &= \{\{7, 0, 2\}, \{3, 5\}, \{6, 4\}, \{1\}\}. \end{aligned}$$

The equality $f(\mathcal{A}) = \mathcal{B}$ holds, and thus f maps the partition \mathcal{A} to \mathcal{B} . ▲

Lemma 3.14. Let f be a permutation of E and \mathcal{A}, \mathcal{B} be two partitions of E . If for any part A of \mathcal{A} , $f(A)$ is a part of \mathcal{B} , then f maps \mathcal{A} to \mathcal{B} .

Proof. Suppose that for all A in \mathcal{A} , $f(A)$ lies in \mathcal{B} . By hypothesis, $f(\mathcal{A})$ is included in \mathcal{B} . It remains to show that \mathcal{B} is a subset of $f(\mathcal{A})$. Let B be a part of \mathcal{B} and let y be an element of B . Since f is onto, there exists x in E such that $f(x) = y$. Furthermore, there exists a unique part A of \mathcal{A} which contains x as \mathcal{A} is a partition de E . Then, y belongs to $f(A)$ and B . Observe that $f(A)$ and B are two non-disjoint parts of \mathcal{B} . Consequently, $f(A) = B$ and B belongs to $f(\mathcal{A})$. The result follows. ■

		.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
$f(x)$	0.	1E	08	04	13	0F	18	14	10	19	15	0E	0D	03	1C	07	17
	1.	12	11	0B	1B	09	05	1F	00	0A	01	02	1A	06	0C	1D	16

 Figure 3.2: The permutation f of Example 3.17.

In this chapter, we will consider a special kind of partitions which is composed of all the cosets of a linear subspace. Such partitions have already been introduced by Harpes [50, Definition 4.4] and are recalled below.

Definition 3.15 (linear partition). Let \mathcal{A} be a partition of \mathbb{F}_2^n . Let V denote its part containing 0_n . The partition \mathcal{A} is said to be *linear* if V is a subspace of \mathbb{F}_2^n and if every part of \mathcal{A} is a coset of V in \mathbb{F}_2^n , in other words, if

$$\mathcal{A} = \{x + V \mid x \in \mathbb{F}_2^n\} = \mathbb{F}_2^n / V.$$

We denote by $\mathcal{L}(V)$ such a partition.

Remark 3.16. It turns out that the linear partitions associated with the two trivial subspaces of \mathbb{F}_2^n , that is $\{0_n\}$ and \mathbb{F}_2^n , correspond with the two trivial partitions of \mathbb{F}_2^n . Moreover, if V is a non-trivial subspace of \mathbb{F}_2^n , then the linear partition $\mathcal{L}(V)$ is also non-trivial.

Example 3.17. Consider the subspaces V and W of \mathbb{F}_2^5 defined to be

$$V = \text{span}(07, 1A) = \{00, 07, 1A, 1D\} \quad \text{and} \quad W = \text{span}(0E, 12) = \{00, 0E, 12, 1C\}.$$

Since both V and W are 2-dimensional subspaces of \mathbb{F}_2^5 , the quotient spaces $\mathcal{L}(V) = \mathbb{F}_2^5/V$ and $\mathcal{L}(W) = \mathbb{F}_2^5/W$ are 3-dimensional. In other words, the two linear partitions $\mathcal{L}(V)$ and $\mathcal{L}(W)$ have $2^3 = 8$ parts. It can be verified that

$$\begin{aligned} \mathcal{L}(V) &= \{V, 01 + V, 02 + V, 03 + V, 08 + V, 09 + V, 0A + V, 0B + V\}, \\ \mathcal{L}(W) &= \{W, 01 + W, 02 + W, 03 + W, 04 + W, 05 + W, 06 + W, 07 + W\}. \end{aligned}$$

For instance, the part $0B + V$ of the linear partition $\mathcal{L}(V)$ is the coset of V with respect to $0B$. Explicitly, it is equal to

$$0B + V = \{0B + 00, 0B + 07, 0B + 1A, 0B + 1D\} = \{0B, 0C, 11, 16\}.$$

Now, consider the permutation f of \mathbb{F}_2^5 given in Figure 3.2. The image of $0B + V$ under f is

$$\begin{aligned} f(0B + V) &= f(\{0B, 0C, 11, 16\}) = \{0D, 03, 11, 1F\} \\ &= \{03 + 0E, 03 + 00, 03 + 12, 03 + 1F\} = 03 + W. \end{aligned}$$

Observe that $f(0B + V)$ is a coset of W so a part of $\mathcal{L}(W)$. The images of all cosets of V under f are displayed in Figure 3.3. Since any of them is a part of $\mathcal{L}(W)$, the permutation f maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. It is worthwhile to observe that a permutation

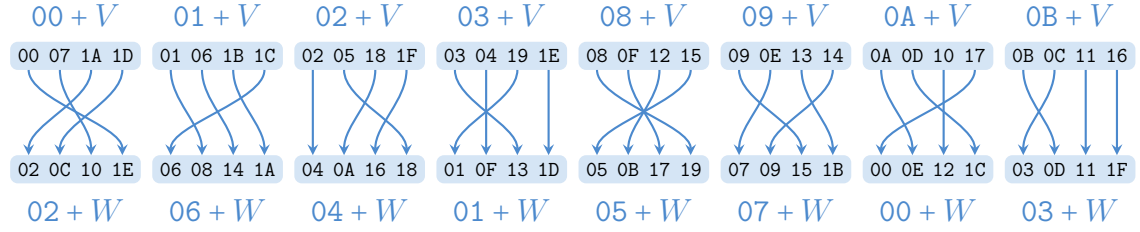


Figure 3.3: The permutation f mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$ where $V = \text{span}(07, 1A)$ and $W = \text{span}(0E, 12)$.

mapping a linear partition to another one does not need to be itself linear or even affine. Indeed, f is certainly not linear as $f(00) = 1E \neq 00$. By contradiction, suppose that f is an affine transformation. Then, there exist a linear mapping $L : \mathbb{F}_2^5 \rightarrow \mathbb{F}_2^5$ and an element c of \mathbb{F}_2^5 such that $f(x) = L(x) + c$ holds for all x in \mathbb{F}_2^5 . Therefore,

$$f(x) + f(y) + f(z) = L(x) + c + L(y) + c + L(z) + c = L(x + y + z) + c = f(x + y + z)$$

for all x, y and z in \mathbb{F}_2^5 . Observe that

$$f(00) + f(01) + f(02) = 1E + 08 + 04 = 12 \neq 13 = f(00 + 01 + 02).$$

Thus, f is not an affine transformation. ▲

Lemma 3.18. Let V, W be two subspaces of \mathbb{F}_2^n and f be a permutation of \mathbb{F}_2^n which maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. For any x in \mathbb{F}_2^n , f maps $x + V$ to $f(x) + W$.

Proof. Let x be an element of \mathbb{F}_2^n . By hypothesis, there exists y in \mathbb{F}_2^n such that $f(x + V) = y + W$. Observe that x lies in $x + V$, so $f(x)$ lies in both $y + W$ and $f(x) + W$. Since $y + W$ and $f(x) + W$ are two non-disjoint parts of $\mathcal{L}(W)$, they must be equal. Thus, $f(x + V) = f(x) + W$. ■

Example 3.19. In Example 3.17, we have seen that $f(0B + V) = 03 + W$. Since f maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, the previous lemma states that $f(0B + V) = f(0B) + W = 0D + W$. There is however no contradiction here because $0D$ belongs to $03 + W$. Consequently, the cosets $03 + W$ and $0D + W$ are equal. ▲

The following two propositions are interesting properties of linear partitions which will be used in the rest of this chapter.

Proposition 3.20. Let V_1, V_2, W_1, W_2 be four subspaces of \mathbb{F}_2^n and f be a permutation of \mathbb{F}_2^n which maps $\mathcal{L}(V_1)$ to $\mathcal{L}(W_1)$ and $\mathcal{L}(V_2)$ to $\mathcal{L}(W_2)$. Then f maps $\mathcal{L}(V_1 \cap V_2)$ to $\mathcal{L}(W_1 \cap W_2)$.

Proof. Let $x + (V_1 \cap V_2)$ be a part $\mathcal{L}(V_1 \cap V_2)$. Observe that $x + (V_1 \cap V_2) = (x + V_1) \cap (x + V_2)$. Now,

$$f(x + (V_1 \cap V_2)) = f((x + V_1) \cap (x + V_2)) = f(x + V_1) \cap f(x + V_2)$$

as f is one-to-one. Then, Lemma 3.18 ensures that $f(x + V_1) = f(x) + W_1$ and $f(x + V_2) = f(x) + W_2$. Next,

$$f(x + (V_1 \cap V_2)) = (f(x) + W_1) \cap (f(x) + W_2) = f(x) + (W_1 \cap W_2).$$

This show that the image of any part of $\mathcal{L}(V_1 \cap V_2)$ under f lies in $\mathcal{L}(W_1 \cap W_2)$. The result is then a consequence of Lemma 3.14. ■

Proposition 3.21. Let V, W be two subspaces of \mathbb{F}_2^n and f be a permutation of \mathbb{F}_2^n which maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. There exists an automorphism L of \mathbb{F}_2^n such that $L(V) = W$. In particular, V and W are isomorphic.

Proof. By definition, $f(V)$ belongs to $\mathcal{L}(W)$. Thus, there exists an element x of \mathbb{F}_2^n such that $f(V) = x + W$. Consequently, V and W have the same finite cardinality. Hence, V and W have the same dimension denoted by d . Let $(v_i)_{0 \leq i < d}$ and $(w_i)_{0 \leq i < d}$ be two bases of V and W respectively. According to the incomplete basis theorem, there exist two families $(v_i)_{d \leq i < n}$ and $(w_i)_{d \leq i < n}$ such that $\mathcal{B}_V = (v_i)_{0 \leq i < n}$ et $\mathcal{B}_W = (w_i)_{0 \leq i < n}$ are two bases of \mathbb{F}_2^n . Denoting by L the linear mapping which maps v_i to w_i for all $0 \leq i < n$, we get an automorphism of \mathbb{F}_2^n satisfying the equality $L(V) = W$. ■

Example 3.22. Consider again the permutation f of \mathbb{F}_2^5 defined as in Figure 3.9. As seen in the previous example, the permutation maps the linear partition $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Then, Proposition 3.21 ensures that there exists a linear permutation L of \mathbb{F}_2^5 such that $L(V) = W$. Following its proof, consider the bases $(07, 1A)$ and $(0E, 12)$ of V and W respectively and complete them into the following bases of \mathbb{F}_2^5

$$\mathcal{B}_V = (v_i)_{i < 5} = (07, 1A, 01, 02, 08) \quad \text{and} \quad \mathcal{B}_W = (w_i)_{i < 5} = (0E, 12, 01, 02, 04).$$

Then, the mapping L can be defined by the rule $L(v_i) = w_i$ for each $i < 5$. This linear transformation will be used in the next chapter. ■

3.2.2. The Key Addition and Diffusion Layer

Before tackling the full SPN, we look at its basic operations and primitives. Recall that the round function is made up of a *key addition*, a *substitution layer* and a *diffusion layer*. The attacker knows the specifications of the substitution and diffusion layers but he does not know the round key used in the key addition. Therefore, the key addition should not be considered as one operation but rather as a family of permutations. To get back to the subject at hand, we must first determine the partitions \mathcal{A} which are mapped to a unique partition under the action of all round keys.

The next proposition explains the fundamental property of linear partitions according to the key addition. This result was introduced by Harpes in [50, Lemma 4.3] and [52, Theorem 4]. Later, Caranti et al. gave a similar result expressed for imprimitive groups in [31]. For convenience, we restate and prove this result with our own notations.

Proposition 3.23. Let n be a positive integer. Let \mathcal{A} and \mathcal{B} be two partitions of \mathbb{F}_2^n . For each k in \mathbb{F}_2^n , let α_k denote the permutation of \mathbb{F}_2^n defined by the rule $\alpha_k(x) = x + k$. Then, the permutation α_k maps \mathcal{A} to \mathcal{B} for any k in \mathbb{F}_2^n if and only if $\mathcal{A} = \mathcal{B}$ and \mathcal{A} is a linear partition.

Proof. Firstly, suppose that $\alpha_k(\mathcal{A}) = \mathcal{B}$ for any k in \mathbb{F}_2^n . Especially, choosing $k = 0_n$ gives $\alpha_{0_n}(\mathcal{A}) = \mathcal{B}$, and thus $\mathcal{A} = \mathcal{B}$ since α_{0_n} is the identity mapping. Let V denote the part of \mathcal{A} containing 0_n . It is sufficient to show that V is a subgroup of \mathbb{F}_2^n because any subgroup of \mathbb{F}_2^n is also a \mathbb{F}_2 -linear subspace of \mathbb{F}_2^n . Let v_1 and v_2 be two elements of V . Since $\alpha_{v_1}(0_n) = v_1$, the intersection $\alpha_{v_1}(V) \cap V$ is non-empty. We know that α_{v_1} maps \mathcal{A} to \mathcal{A} , so $\alpha_{v_1}(V)$ lies in \mathcal{A} . Thus, $\alpha_{v_1}(V) = V$ since \mathcal{A} is a partition. It follows that $\alpha_{v_1}(v_2) = v_1 + v_2$ is an element of V . Therefore, the subset V of \mathbb{F}_2^n is closed under the operation of addition and because every element of \mathbb{F}_2^n is its own inverse, V is a subgroup of \mathbb{F}_2^n . Furthermore, for any x in \mathbb{F}_2^n , $\alpha_x(V) = x + V$ must be a part of \mathcal{A} . Thus, \mathcal{A} is linear.

Conversely, suppose that the partition \mathcal{A} is linear and that $\mathcal{A} = \mathcal{B}$. Let V denote the part of \mathcal{A} containing 0_n and let x be an element of \mathbb{F}_2^n . Then,

$$\alpha_x(\mathcal{A}) = \alpha_x(\{y + V \mid y \in \mathbb{F}_2^n\}) = \{(x + y) + V \mid y \in \mathbb{F}_2^n\} = \mathcal{A}.$$

The result is proven. ■

Even if this result was easily obtained, it has maybe the most important impact on our study. Due to this result and its generalization given later in the next section, only linear partitions will be considered. By definition, the linear partitions are quotient spaces, and hence highly structured algebraic objects. Consequently, the apparent combinatorial aspect of our study is reduced to an algebraic problem. This result is indeed quite restrictive since the linear partitions account for a small proportion of all partitions.

Example 3.24. Let n and k be non-negative integers and q be a prime power. The q -binomial (or Gaussian) coefficient is defined to be

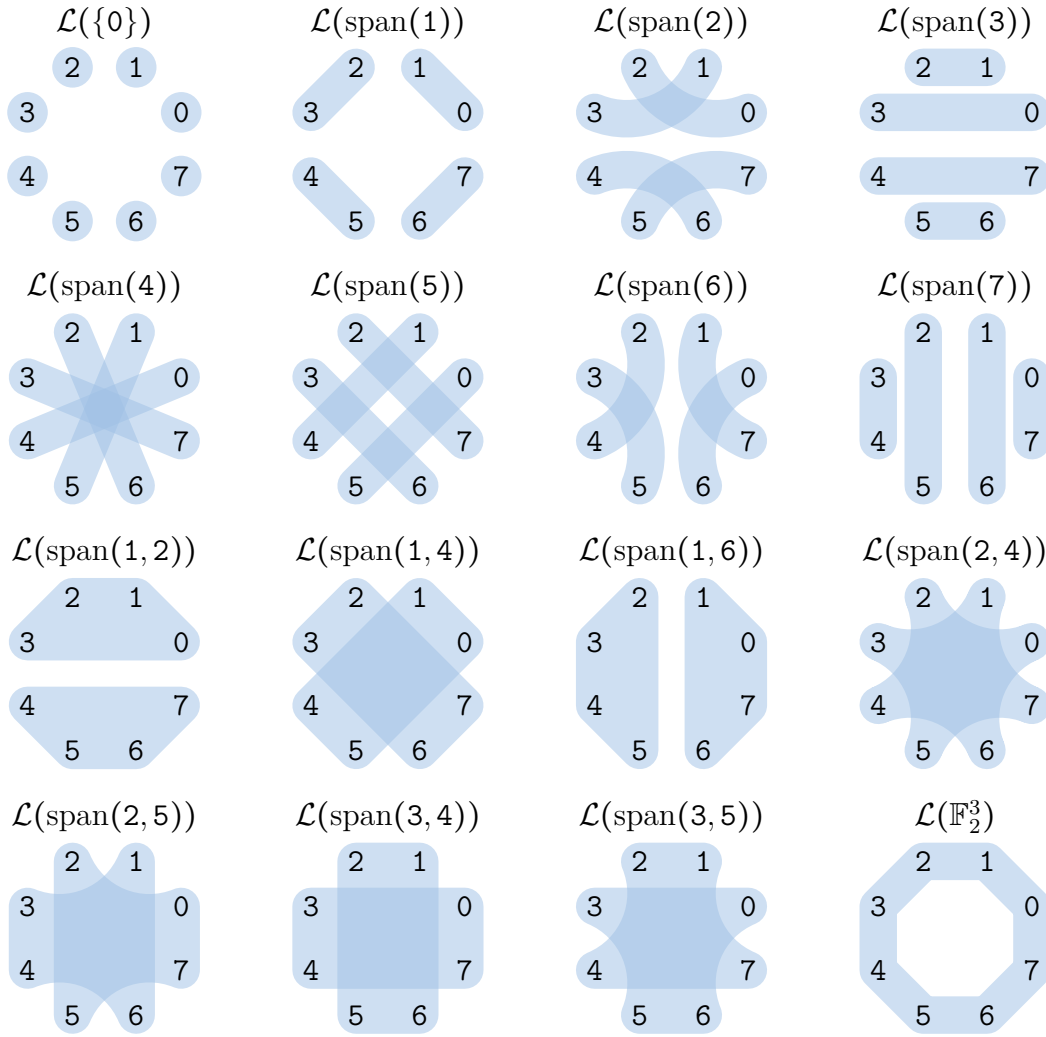
$$\begin{bmatrix} n \\ d \end{bmatrix}_q = \prod_{i=1}^d \frac{1 - q^{n-i+1}}{1 - q^i}.$$

It can be proved that this coefficient counts the number of d -dimensional subspaces of an n -dimensional vector space over the finite field \mathbb{F}_q [46]. Therefore, the number of subspaces of \mathbb{F}_2^3 is given by

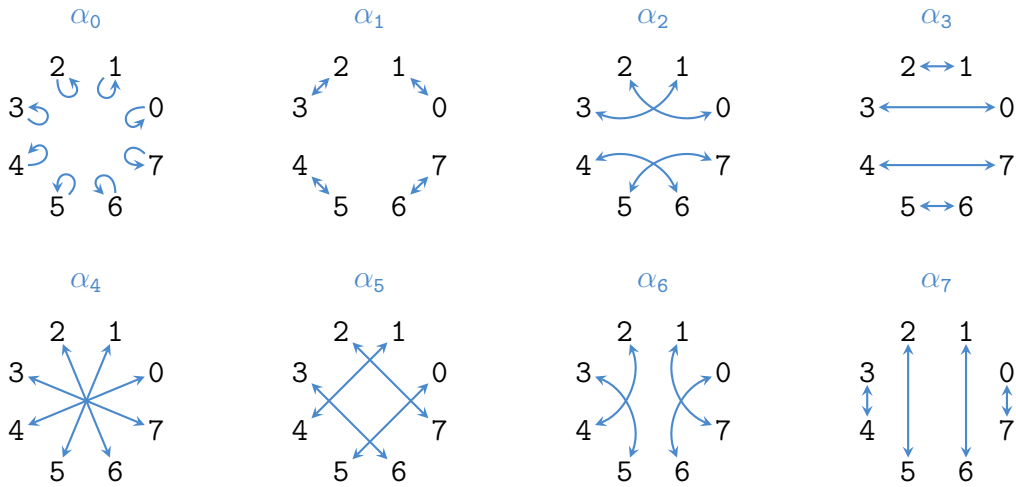
$$\begin{aligned} \sum_{d=0}^3 \begin{bmatrix} 3 \\ d \end{bmatrix}_2 &= 1 + \frac{1 - 2^3}{1 - 2} + \frac{(1 - 2^3)(1 - 2^2)}{(1 - 2)(1 - 2^2)} + \frac{(1 - 2^3)(1 - 2^2)(1 - 2^1)}{(1 - 2)(1 - 2^2)(1 - 2^3)} \\ &= 1 + 7 + 7 + 1 = 16. \end{aligned}$$

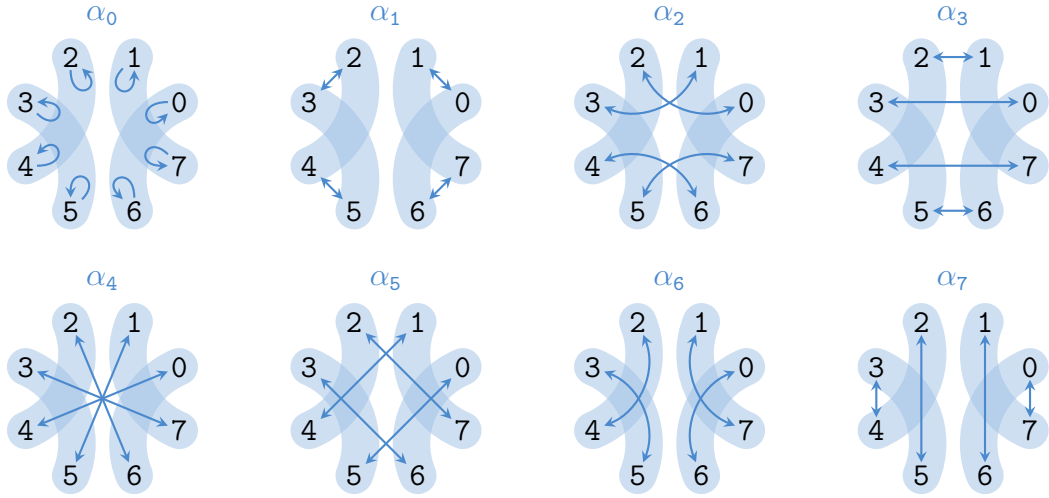
Since a linear partition of \mathbb{F}_2^3 is uniquely determined by a subspace of \mathbb{F}_2^3 , there are exactly 16 linear partitions. All these partitions are represented graphically at the top of Figure 3.4. For instance, the linear partition associated with the subspace $\text{span}(2, 4) = \{0, 2, 4, 6\}$ is $\mathcal{L}(\text{span}(2, 4)) = \{\{0, 2, 4, 6\}, \{1, 3, 5, 7\}\}$. The On-line Encyclopedia of Integer Sequences (OEIS [93]) includes almost all known integer sequences. The 2-binomial coefficients are given in the sequence A022166 and their sums are given in A006116.

Proposition 3.23 states that, among the set of all the partitions of \mathbb{F}_2^n , only the linear ones yield a unique output partition for every key. The Bell number B_m

Every linear partitions over \mathbb{F}_2^3


All the key additions


 Figure 3.4: All linear partitions and key additions in \mathbb{F}_2^3 .


 Figure 3.5: The key additions preserving the partition $\mathcal{L}(\text{span}(6))$.

counts the number of partitions of a set of size m (see sequence A000110). Thus, the number of partitions of \mathbb{F}_2^n is B_{2^n} . For $n = 3$, there are $B_8 = 4140$ partitions in all. Hence, the linear partitions represent a fraction of $16/B_8 \approx 2^{-8.0}$. This ratio falls greatly as n increase. In fact, for $n = 4$, only $67/B_{16} \approx 2^{-27.2}$ are linear and for $n = 5$, this ratio becomes $374/B_{32} \approx 2^{-78.2}$. This underlines how Proposition 3.23 is restrictive.

All the key additions are given at the bottom of Figure 3.4. The reverse implication of Proposition 3.23 states that any linear partition is preserved by all the key additions. For instance,

$$\begin{aligned} \alpha_2(\mathcal{L}(\text{span}(6))) &= \{f(\{0, 6\}), f(\{1, 7\}), f(\{2, 4\}), f(\{3, 5\})\} \\ &= \{ \{2, 4\}, \{3, 5\}, \{0, 6\}, \{1, 7\} \} = \mathcal{L}(\text{span}(6)). \end{aligned}$$

Thus, the permutation α_2 preserves $\mathcal{L}(\text{span}(6))$. Figure 3.5 illustrates graphically that this linear partition is preserved by all the key additions. It is then not hard to check that the same holds for every linear partition given in Figure 3.4. ▀

Now that we know linear partitions are of major importance, we focus on how the diffusion layer deals with these partitions.

Proposition 3.25. Let n be a positive integer. Let L be an automorphism of \mathbb{F}_2^n and V a subspace of \mathbb{F}_2^n . Then, $L(\mathcal{L}(V)) = \mathcal{L}(L(V))$. In particular, L maps a linear partition to another one.

Proof. Since L is an automorphism, we have

$$\begin{aligned} L(\mathcal{L}(V)) &= L(\{x + V \mid x \in \mathbb{F}_2^n\}) = \{L(x + V) \mid x \in \mathbb{F}_2^n\} \\ &= \{L(x) + L(V) \mid x \in \mathbb{F}_2^n\} = \{x' + L(V) \mid x' \in \mathbb{F}_2^n\}. \end{aligned}$$

Moreover, $L(V)$ is a subspace of \mathbb{F}_2^n because L is a linear mapping. Consequently, $L(\mathcal{L}(V)) = \mathcal{L}(L(V))$. ■

If V and W are two subspaces of \mathbb{F}_2^n , it is straightforward to design a linear permutation L of \mathbb{F}_2^n mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Indeed, Proposition 3.25 establishes that L maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ if and only if $L(V) = W$. In other words, we only need to consider the image of V and not the whole linear partition $\mathcal{L}(V)$.

3.2.3. From the Encryption Function to the Substitution Layer

Along with the two results of the previous section, we can now address our main issue. For the rest of this chapter, we consider a generic SPN whose parameters are defined as follows.

Let m, n and r be positive integers.

Let S_0, \dots, S_{m-1} be n -bit S-boxes.

- The *addition* of the round key k is denoted by $\alpha_k : \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$, $x \mapsto x + k$.
- The *substitution layer* is denoted by σ and maps $(x_i)_{0 \leq i < m}$ to $(S_i(x_i))_{0 \leq i < m}$.
- The *diffusion layer* is a linear permutation denoted by $\pi : \mathbb{F}_2^{nm} \rightarrow \mathbb{F}_2^{nm}$.

The round function F_k associated with the round key k is defined to be $F_k = \pi \sigma \alpha_k$. The *encryption function* associated with the round keys $K = (k^{[0]}, \dots, k^{[r]})$ in $(\mathbb{F}_2^{nm})^{r+1}$ is defined to be

$$E_K = \alpha_{k^{[r]}} F_{k^{[r-1]}} \dots F_{k^{[0]}}.$$

We can now prove the following result.

Theorem 3.26. Let \mathcal{A} and \mathcal{B} be two partitions of \mathbb{F}_2^{nm} . Suppose for any $(r+1)$ -tuples of round keys $K = (k^{[0]}, \dots, k^{[r]})$ in $(\mathbb{F}_2^{nm})^{r+1}$ that the encryption function E_K maps \mathcal{A} to \mathcal{B} . Define $\mathcal{A}^{[0]} = \mathcal{A}$ and for all $1 \leq i \leq r$, $\mathcal{A}^{[i]} = (\pi \sigma)^i(\mathcal{A})$. Then,

- $\mathcal{A}^{[r]} = \mathcal{B}$;
- for any $0 \leq i < r$ and for any $k^{[i]}$ in \mathbb{F}_2^{nm} , $F_{k^{[i]}}(\mathcal{A}^{[i]}) = \mathcal{A}^{[i+1]}$;
- for any $0 \leq i \leq r$, $\mathcal{A}^{[i]}$ is a linear partition.

Proof. Observe that for the round key $k = 0_{nm}$, the key addition $\alpha_{0_{nm}}$ is the identity mapping on \mathbb{F}_2^{nm} , and thus $F_{0_{nm}} = \pi \sigma \alpha_{0_{nm}} = \pi \sigma$. Now, choosing $K = (k^{[0]}, \dots, k^{[r]}) = (0_{nm}, \dots, 0_{nm})$ gives

$$\begin{aligned} \mathcal{B} &= E_K(\mathcal{A}^{[0]}) = \alpha_{k^{[r]}} F_{k^{[r-1]}} \dots F_{k^{[0]}}(\mathcal{A}^{[0]}) = \alpha_{0_{nm}} (F_{0_{nm}})^r(\mathcal{A}^{[0]}) \\ &= (\pi \sigma)^r(\mathcal{A}^{[0]}) = \mathcal{A}^{[r]}. \end{aligned}$$

Let $0 \leq i < r$ be an integer. Let $k^{[i]}$ be any element of \mathbb{F}_2^{nm} . Define $k^{[j]} = 0_{nm}$ for all $0 \leq j \leq r$ such that $j \neq i$. By hypothesis, the equality $\alpha_{k^{[r]}} F_{k^{[r-1]}} \dots F_{k^{[0]}}(\mathcal{A}^{[0]}) = \mathcal{A}^{[r]}$ holds. Thus,

$$F_{k^{[i]}} \dots F_{k^{[0]}}(\mathcal{A}^{[0]}) = (\alpha_{k^{[r]}} F_{k^{[r-1]}} \dots F_{k^{[i+1]}})^{-1}(\mathcal{A}^{[r]}).$$

On one hand,

$$\begin{aligned} F_{k^{[i]}} \dots F_{k^{[0]}}(\mathcal{A}^{[0]}) &= F_{k^{[i]}}(F_{k^{[i-1]}} \dots F_{k^{[0]}})(\mathcal{A}^{[0]}) = F_{k^{[i]}}(F_{0_{nm}})^i(\mathcal{A}^{[0]}) \\ &= F_{k^{[i]}}(\pi \sigma)^i(\mathcal{A}^{[0]}) = F_{k^{[i]}}(\mathcal{A}^{[i]}). \end{aligned}$$

On the other hand,

$$\begin{aligned} (\alpha_{k[r]} F_{k[r-1]} \dots F_{k[i+1]})^{-1}(\mathcal{A}^{[r]}) &= (\alpha_{0_{nm}} (F_{0_{nm}})^{r-(i+1)})^{-1}(\mathcal{A}^{[r]}) \\ &= ((\pi\sigma)^{r-(i+1)})^{-1}(\mathcal{A}^{[r]}) = \mathcal{A}^{[i+1]}. \end{aligned}$$

Therefore, $F_{k[i]}(\mathcal{A}^{[i]}) = \mathcal{A}^{[i+1]}$, or equivalently $\alpha_{k[i]}(\mathcal{A}^{[i]}) = (\pi\sigma)^{-1}(\mathcal{A}^{[i+1]})$. Since this equality holds for every $k[i]$, Proposition 3.23 states that the partition $\mathcal{A}^{[i]}$ is linear.

It remains to show that $\mathcal{A}^{[r]}$ is linear as the previous argument holds only for $i < r$. Let $k^{[r]}$ be an element of \mathbb{F}_2^{nm} . Define $k^{[i]} = 0_{nm}$ for each $0 \leq i < r$. Then,

$$\mathcal{A}^{[r]} = \alpha_{k[r]} F_{k[r-1]} \dots F_{k[0]}(\mathcal{A}^{[0]}) = \alpha_{k[r]} (F_{0_{nm}})^r(\mathcal{A}^{[0]}) = \alpha_{k[r]}(\mathcal{A}^{[r]}).$$

Again, Proposition 3.23 implies that $\mathcal{A}^{[r]}$ is linear and the result is proven. ■

This theorem can be restated in the following way. Firstly, the input partition \mathcal{A} and the output partition \mathcal{B} must be linear. This result generalizes Proposition 3.23 in the sense that it applies to the full cipher and not only to the key addition. As was pointed in Example 3.24, linear partitions are very specific partitions. This means that our combinatorial hypothesis implies to consider only algebraic objects.

Secondly, we have only supposed that the encryption function maps \mathcal{A} to \mathcal{B} after r rounds. Nevertheless, Theorem 3.26 ensures that each iteration of the round function also maps a fixed linear partition to another one. As a consequence, the study of the full cipher is reduced to the study of the round function. Additionally, this result can be strengthened as follows.

Corollary 3.27. Keep the notations of Theorem 3.26. For all $0 \leq i \leq r$, let $V^{[i]}$ denote the part of $\mathcal{A}^{[i]}$ containing 0. According to Theorem 3.26, $\mathcal{A}^{[i]} = \mathcal{L}(V^{[i]})$. Let $0 \leq i < r$ be an integer. Then,

$$\sigma(\mathcal{L}(V^{[i]})) = \mathcal{L}(W^{[i]}).$$

where $W^{[i]}$ denotes the subspace $\pi^{-1}(V^{[i+1]})$. In particular, the substitution layer must at least map one linear partition to another one.

Proof. By definition, $\pi\sigma(\mathcal{A}^{[i]}) = \mathcal{A}^{[i+1]}$ or, equivalently, $\sigma(\mathcal{A}^{[i]}) = \pi^{-1}(\mathcal{A}^{[i+1]})$. This equality can be restated as

$$\sigma(\mathcal{L}(V^{[i]})) = \pi^{-1}(\mathcal{L}(V^{[i+1]})).$$

As π is an automorphism of \mathbb{F}_2^{nm} , then so π^{-1} is. Next, Proposition 3.25 ensures that $\pi^{-1}(\mathcal{L}(V^{[i+1]})) = \mathcal{L}(\pi^{-1}(V^{[i+1]}))$. The result follows. ■

A diagrammatic representation of Theorem 3.26 and Corollary 3.27 is given in Figure 3.6. This highlights that the input partition is always transformed in the same way through each basic operation of the encryption process. The results obtained so far can be summarized as follows: if an SPN maps a partition \mathcal{A} of the plaintext space to a partition \mathcal{B} of the ciphertext space no matter the round keys used, then the substitution layer has to map at least one linear partition to another one. This shows that our study can be reduced to the substitution layer without loss of generality.

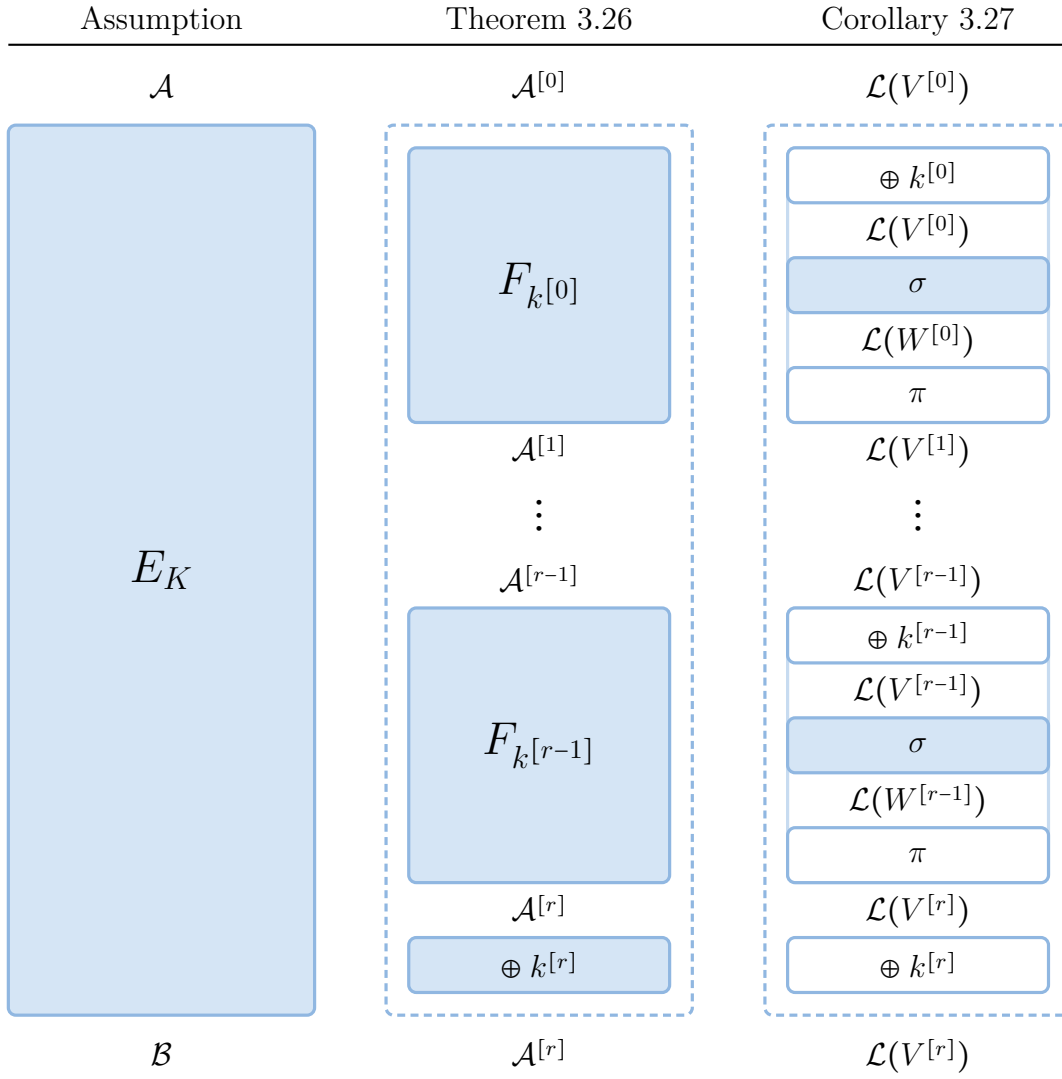


Figure 3.6: Results of Section 3.2.3.

		.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
$S_0(x)$	0.	1F	19	03	05	1D	1B	01	07	14	12	1C	1A	16	10	1E	18
	1.	0E	08	09	0F	0C	0A	0B	0D	04	02	17	11	06	00	15	13
$S_1(x)$	0.	02	19	11	14	1B	0E	0C	07	15	0A	01	00	0D	1C	1D	12
	1.	06	1E	10	16	05	13	17	1F	18	04	09	0B	1A	08	0F	03
$S_2(x)$	0.	1E	08	04	13	0F	18	14	10	19	15	0E	0D	03	1C	07	17
	1.	12	11	0B	1B	09	05	1F	00	0A	01	02	1A	06	0C	1D	16
$S_3(x)$	0.	03	0A	10	1A	15	04	1C	0E	12	18	02	0B	06	14	0C	1D
	1.	1B	09	11	00	0F	05	1F	16	08	19	01	13	1E	17	0D	07

Figure 3.7: Specification of the S-boxes used throughout Section 3.3.

3.3. Structure of the Substitution Layer

In the remainder of this chapter, V and W will denote two subspaces of $(\mathbb{F}_2^n)^m$.

As explained in the previous section, it remains to understand how the substitution layer can map the linear partition $\mathcal{L}(V)$ to $\mathcal{L}(W)$. This problem is far more complex for the substitution layer than it was for the diffusion layer. The reasons for this are twofold. First, the substitution layer is non-linear. It is even the only part of the SPN which is not affine. As a consequence, to map the linear partition $\mathcal{L}(V)$ to $\mathcal{L}(W)$, we have to consider all the parts of both partitions and not only the subspaces V and W , as was the case for the diffusion layer (see Proposition 3.25).

Secondly, the substitution layer should not be considered as a whole, but as the parallel application of its S-boxes. Therefore our problem becomes the following. Given two subspaces V and W , what are the necessary and/or sufficient conditions on the S-boxes for the substitution layer to map $\mathcal{L}(V)$ to $\mathcal{L}(W)$.

Before going any further, let us introduce an example that we will continue throughout this section.

Example 3.28. Consider the substitution layer made up of the four 5-bit S-boxes S_0 , S_1 , S_2 and S_3 described in Figure 3.7. Its parameters are then $m = 4$ and $n = 5$. Observe that the S-box S_2 was previously studied in Example 3.17. Define the two families $\mathcal{E}_V = (v_i)_{0 \leq i < 7}$ and $\mathcal{E}_W = (w_i)_{0 \leq i < 7}$ of elements of $(\mathbb{F}_2^5)^4$ as follows:

$$\begin{aligned}
 v_0 &= (10, 00, 00, 17), & v_3 &= (02, 00, 00, 1C), & w_0 &= (10, 00, 00, 15), & w_3 &= (02, 00, 00, 08), \\
 v_1 &= (08, 00, 00, 17), & v_4 &= (01, 00, 00, 1C), & w_1 &= (08, 00, 00, 1D), & w_4 &= (01, 00, 00, 00), \\
 v_2 &= (04, 00, 00, 0B), & v_5 &= (00, 00, 1A, 00), & w_2 &= (04, 00, 00, 15), & w_5 &= (00, 00, 12, 00), \\
 & & v_6 &= (00, 00, 07, 00). & & & w_6 &= (00, 00, 0E, 00).
 \end{aligned}$$

Finally, define V and W to be the subspaces spanned by \mathcal{E}_V and \mathcal{E}_W respectively. Note that the family \mathcal{E}_V is linearly independent because it is echelonized. Hence, \mathcal{E}_V

is a basis of V . The same applies for \mathcal{E}_W and W . As a consequence, V and W are both 7-dimensional subspaces of $(\mathbb{F}_2^5)^4$.

We claim that the substitution layer σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Naturally, we will not verify this statement by hand because it requires to check for each of the 2^{13} cosets of V that the 2^7 images of its elements under σ lies in the same coset of W . However, the reader who is reluctant to accept this claim is encouraged to check it with a computer. ▀

3.3.1. Truncating the substitution layer

To understand how the substitution layer can maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, we will adopt a *divide and conquer* strategy. That is to say, we want to break down this problem into several independent sub-problems, each involving less S-boxes than the full substitution layer. The first idea is to truncate the substitution layer and the subspaces V and W to get a local view of what happens on some S-boxes.

Definition 3.29 (Truncation and Substitution Layer). Let E be any non-empty subset of $\llbracket 0, m \rrbracket$ and define the following mappings

$$\begin{aligned} T_E : (\mathbb{F}_2^n)^m &\longrightarrow (\mathbb{F}_2^n)^E & \sigma_E : (\mathbb{F}_2^n)^E &\longrightarrow (\mathbb{F}_2^n)^E \\ (x_i)_{0 \leq i < m} &\longmapsto (x_i)_{i \in E} & (x_i)_{i \in E} &\longmapsto (S_i(x_i))_{i \in E}. \end{aligned}$$

If E has cardinality p , then we identify $(\mathbb{F}_2^n)^E$ with $(\mathbb{F}_2^n)^p$.

The mapping T_E allows to shorten a vector of $(\mathbb{F}_2^n)^m$ to keep only the coordinates whose indices belongs to E . The application σ_E is a substitution layer truncated to the S-boxes whose indices lie in E .

Remark 3.30. Note that T_E is a linear mapping. Observe that $\sigma_{\llbracket 0, m \rrbracket}$ is the substitution layer of the SPN. Moreover, the truncated substitution layer $\sigma_{\{i\}}$ and the S-box S_i are equal for all $0 \leq i < m$.

Proposition 3.31 (Truncating to a few S-boxes). Suppose that σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Let E be a non-empty subset of $\llbracket 0, m \rrbracket$. Then, the permutation σ_E maps $\mathcal{L}(T_E(V))$ to $\mathcal{L}(T_E(W))$.

Proof. Let $x = (x_i)_{i \in E}$ be an element of $(\mathbb{F}_2^n)^E$. Let y be the element of $(\mathbb{F}_2^n)^m$ defined by the rule $y_i = x_i$ if i belongs to E and $y_i = 0_n$ otherwise. Thus, $T_E(y) = x$. By hypothesis, σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Hence, Lemma 3.18 implies that, $\sigma(y + V) = \sigma(y) + W$. Next,

$$T_E(\sigma(y + V)) = T_E(\sigma(y)) + T_E(W)$$

since T_E is a linear mapping. Furthermore,

$$\begin{aligned} T_E(\sigma(y + V)) &= T_E\sigma(\{y + v \mid v \in V\}) = \{T_E\sigma(y + v) \mid v \in V\} \\ &= \{\sigma_E(T_E(y + v)) \mid v \in V\} = \sigma_E(\{T_E(y + v) \mid v \in V\}) \\ &= \sigma_E(\{T_E(y) + T_E(v) \mid v \in V\}) = \sigma_E(T_E(y) + T_E(V)). \end{aligned}$$

$(07, 03) + T_{\{0,3\}}(V) \longrightarrow (07, 1A) + T_{\{0,3\}}(W)$	$(07, 03) + T_{\{0,3\}}(V) \longrightarrow (07, 1A) + T_{\{0,3\}}(W)$
$(07, 03) + (00, 00) \mapsto (07, 1A) + (00, 00)$	$(07, 03) + (10, 17) \mapsto (07, 1A) + (0A, 15)$
$(07, 03) + (01, 1C) \mapsto (07, 1A) + (06, 1D)$	$(07, 03) + (11, 0B) \mapsto (07, 1A) + (0C, 08)$
$(07, 03) + (02, 1C) \mapsto (07, 1A) + (1C, 1D)$	$(07, 03) + (12, 0B) \mapsto (07, 1A) + (0D, 08)$
$(07, 03) + (03, 00) \mapsto (07, 1A) + (1A, 00)$	$(07, 03) + (13, 17) \mapsto (07, 1A) + (0B, 15)$
$(07, 03) + (04, 0B) \mapsto (07, 1A) + (02, 08)$	$(07, 03) + (14, 1C) \mapsto (07, 1A) + (08, 1D)$
$(07, 03) + (05, 17) \mapsto (07, 1A) + (04, 15)$	$(07, 03) + (15, 00) \mapsto (07, 1A) + (0E, 00)$
$(07, 03) + (06, 17) \mapsto (07, 1A) + (1E, 15)$	$(07, 03) + (16, 00) \mapsto (07, 1A) + (0F, 00)$
$(07, 03) + (07, 0B) \mapsto (07, 1A) + (18, 08)$	$(07, 03) + (17, 1C) \mapsto (07, 1A) + (09, 1D)$
$(07, 03) + (08, 17) \mapsto (07, 1A) + (1F, 15)$	$(07, 03) + (18, 00) \mapsto (07, 1A) + (14, 00)$
$(07, 03) + (09, 0B) \mapsto (07, 1A) + (19, 08)$	$(07, 03) + (19, 1C) \mapsto (07, 1A) + (12, 1D)$
$(07, 03) + (0A, 0B) \mapsto (07, 1A) + (17, 08)$	$(07, 03) + (1A, 1C) \mapsto (07, 1A) + (07, 1D)$
$(07, 03) + (0B, 17) \mapsto (07, 1A) + (11, 15)$	$(07, 03) + (1B, 00) \mapsto (07, 1A) + (01, 00)$
$(07, 03) + (0C, 1C) \mapsto (07, 1A) + (1D, 1D)$	$(07, 03) + (1C, 0B) \mapsto (07, 1A) + (16, 08)$
$(07, 03) + (0D, 00) \mapsto (07, 1A) + (1B, 00)$	$(07, 03) + (1D, 17) \mapsto (07, 1A) + (10, 15)$
$(07, 03) + (0E, 00) \mapsto (07, 1A) + (15, 00)$	$(07, 03) + (1E, 17) \mapsto (07, 1A) + (05, 15)$
$(07, 03) + (0F, 1C) \mapsto (07, 1A) + (13, 1D)$	$(07, 03) + (1F, 0B) \mapsto (07, 1A) + (03, 08)$

 Figure 3.8: $\sigma_{\{0,3\}}$ mapping a coset of $T_{\{0,3\}}(V)$ to a coset of $T_{\{0,3\}}(W)$.

Therefore, $\sigma_E(x + T_E(V)) = T_E(\sigma(y)) + T_E(W)$. In other words, the image of any part of $\mathcal{L}(T_E(V))$ under σ_E lies in $\mathcal{L}(T_E(W))$. The result is a consequence of Lemma 3.14. ■

Example 3.32. By choosing $E = \{0, 3\}$, the previous proposition ensures that the truncated substitution layer $\sigma_{\{0,3\}}$ maps $\mathcal{L}(T_{\{0,3\}}(V))$ to $\mathcal{L}(T_{\{0,3\}}(W))$. First, it is easy to see that

$$\begin{aligned} T_{\{0,3\}}(V) &= \text{span}((10, 17), (08, 17), (04, 0B), (02, 1C), (01, 1C)), \\ T_{\{0,3\}}(W) &= \text{span}((10, 15), (08, 1D), (04, 15), (02, 08), (01, 00)). \end{aligned}$$

Again, we will not explicitly check that $\sigma_{\{0,3\}}$ maps $\mathcal{L}(T_{\{0,3\}}(V))$ to $\mathcal{L}(T_{\{0,3\}}(W))$ but limit ourselves to prove that the coset $(07, 03) + T_{\{0,3\}}(V)$ is mapped to one coset of $T_{\{0,3\}}(W)$. Its image can be found using Lemma 3.18 as follow

$$\begin{aligned} \sigma_{\{0,3\}}((07, 03) + T_{\{0,3\}}(V)) &= \sigma_{\{0,3\}}((07, 03)) + T_{\{0,3\}}(W) \\ &= (07, 1A) + T_{\{0,3\}}(W). \end{aligned}$$

The images of every element of this coset are given in Figure 3.8. For instance,

$$\begin{aligned} \sigma_{\{0,3\}}((07, 03) + (01, 1C)) &= \sigma_{\{0,3\}}(06, 1F) = (S_0(06), S_3(1F)) = (01, 07) \\ &= (07, 1A) + (06, 1D). \end{aligned}$$

This explains the second image. ▲

Choosing $E = \{i\}$ in Proposition 3.31 gives that the S-box S_i maps $\mathcal{L}(T_{\{i\}}(V))$ to $\mathcal{L}(T_{\{i\}}(W))$. As this result holds for each index i in $\llbracket 0, m \rrbracket$, we deduce that

$$\sigma(\mathcal{L}(V)) = \mathcal{L}(W) \implies \forall i \in \llbracket 0, m \rrbracket, S_i(\mathcal{L}(T_{\{i\}}(V))) = \mathcal{L}(T_{\{i\}}(W)). \quad (3.1)$$

3.3 – STRUCTURE OF THE SUBSTITUTION LAYER

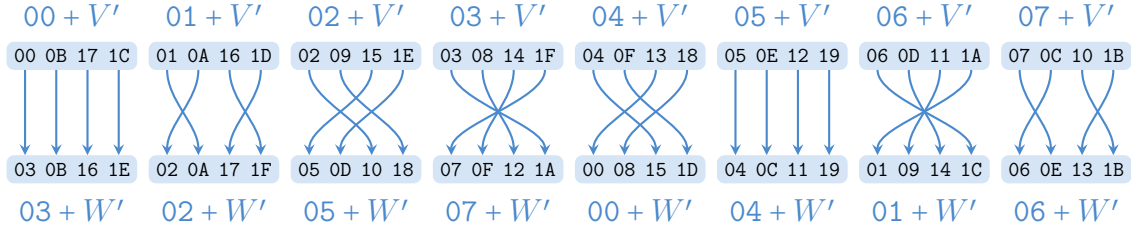


Figure 3.9: The S-box S_3 mapping $\mathcal{L}(V')$ to $\mathcal{L}(W')$ where $V' = \text{span}(0B, 17)$ and $W' = \text{span}(08, 15)$.

However, the equivalence does not hold in general. Hence, this only gives a necessary condition on each S-box. In other words, this means that we can lose information when considering each S-box independently. The next example stresses this fact.

Example 3.33. In our example, the truncated subspaces $T_{\{i\}}(V)$ and $T_{\{i\}}(W)$ are the following:

$$T_{\{0\}}(V) = \mathbb{F}_2^5, \quad T_{\{1\}}(V) = \{00\}, \quad T_{\{2\}}(V) = \text{span}(07, 1A), \quad T_{\{3\}}(V) = \text{span}(0B, 17), \\ T_{\{0\}}(W) = \mathbb{F}_2^5, \quad T_{\{1\}}(W) = \{00\}, \quad T_{\{2\}}(W) = \text{span}(0B, 17), \quad T_{\{3\}}(W) = \text{span}(08, 15).$$

First, observe that the truncated subspaces for S_0 and S_1 are trivial. Hence, the associated linear partitions are also trivial and no information on S_0 or S_1 can be drawn from (3.1). Yet, the last two truncated subspaces are non-trivial and (3.1) gives the following equalities:

$$S_2(\mathcal{L}(\text{span}(07, 1A))) = \mathcal{L}(\text{span}(0B, 17)), \\ S_3(\mathcal{L}(\text{span}(0B, 17))) = \mathcal{L}(\text{span}(08, 15)).$$

The first property has already been highlighted in Example 3.17 and in Figure 3.3. The second one is represented in Figure 3.9.

Let us now show that the converse of Implication 3.1 does not hold in general. Consider the substitution layer σ' made up of the four S-boxes S'_0 , S'_1 , S'_2 and S'_3 where

$$S'_0 = S_1, \quad S'_1 = S_1, \quad S'_2 = S_2, \quad S'_3 = S_3.$$

Thus, this new substitution layer differs from σ by only one S-box. Recall that the linear partition associated with $T_{\{0\}}(V) = T_{\{0\}}(W)$ is trivial. Therefore, S'_0 necessarily preserves this partition. As the other S-boxes remain the same, the right side of 3.1 still holds for σ' , that is

$$\forall i \in \llbracket 0, 4 \rrbracket, \quad S'_i(\mathcal{L}(T_{\{i\}}(V))) = \mathcal{L}(T_{\{i\}}(W)).$$

However, we will prove that σ' does not map $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Suppose by contradiction that it does. Then Proposition 3.31 ensures that $\sigma'_{\{0,3\}}$ maps $\mathcal{L}(T_{\{0,3\}}(V))$ to $\mathcal{L}(T_{\{0,3\}}(W))$. By Lemma 3.18,

$$\begin{aligned} \sigma'_{\{0,3\}}((07, 03) + T_{\{0,3\}}(V)) &= \sigma'_{\{0,3\}}(07, 03) + T_{\{0,3\}}(W) \\ &= (S'_0(07), S'_3(03)) + T_{\{0,3\}}(W) \\ &= (S_1(07), S_3(03)) + T_{\{0,3\}}(W) = (07, 1A) + T_{\{0,3\}}(W). \end{aligned}$$

Then

$$\begin{aligned}\sigma'_{\{0,3\}}((07, 03) + (01, 1C)) &= \sigma'_{\{0,3\}}(06, 1F) = (S'_0(06), S'_3(1F)) = (S_1(06), S_3(1F)) \\ &= (0C, 07) = (07, 1A) + (0B, 1D).\end{aligned}$$

This is a contradiction since $(0B, 1D)$ does not belong to $T_{\{0,3\}}(W)$ as it can be seen in Figure 3.8. As a consequence, the substitution layer σ' does not map $\mathcal{L}(V)$ to $\mathcal{L}(W)$. ▲

As shown in the previous example, truncating the substitution layer and the subspaces V and W to each S-box independently of the others is too restrictive in general. This suggests that some S-boxes can in a way be linked together. That is to say, considering them independently results in a loss of information on the subspaces V and W . Recall that we are interested in splitting the problem of finding all the substitution layers σ mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$ into several independent smaller problems. Taking into account that some S-boxes can be linked together, we require the following:

- a sub-problem can involve several S-boxes;
- the same S-box cannot be involved in two different sub-problems (in other words, the sub-problems are independent);
- each S-box is involved in one sub-problem (possibly trivial).

This is naturally formalized by a partition \mathcal{I} of $\llbracket 0, m \rrbracket$. Each part I of \mathcal{I} represents a sub-problem and its elements are the indices of the S-boxes involved in. By virtue of Proposition 3.31, it holds that

$$\sigma(\mathcal{L}(V)) = \mathcal{L}(W) \implies \forall I \in \mathcal{I}, \sigma_I(\mathcal{L}(T_I(V))) = \mathcal{L}(T_I(W)). \quad (3.2)$$

The next section aims to find a sufficient condition on the partition \mathcal{I} to obtain the equivalence. In such a case, this means that combining the solutions of these sub-problems yields a substitution layer mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$ and vice versa.

3.3.2. Structure of the Subspaces V and W

With the aim of finding partitions for which the converse of 3.2 holds, let us introduce a few definitions and notations.

Definition 3.34 (Wall, V_E and W_E). Let E be a subset of $\llbracket 0, m \rrbracket$. The *wall* associated with E , denoted by Wall_E , is defined to be

$$\text{Wall}_E = \{x \in (\mathbb{F}_2^n)^m \mid \forall i \in E^c, x_i = 0_n\}.$$

Moreover, we denote by V_E the intersection of V and Wall_E , that is $V_E = V \cap \text{Wall}_E = \{v \in V \mid \forall i \in E^c, v_i = 0_n\}$. The subspace W_E is defined in the same way.

Remark 3.35. The notion of wall was introduced by Aragona and Calderini [4, 22]. It is easily seen that

$$\text{Wall}_E = \prod_{i=0}^{m-1} \text{Wall}_E^{[i]} \quad \text{with} \quad \text{Wall}_E^{[i]} = \begin{cases} \{0_n\} & \text{if } i \in E^c, \\ \mathbb{F}_2^n & \text{if } i \in E. \end{cases}$$

3.3 – STRUCTURE OF THE SUBSTITUTION LAYER

$$\begin{aligned}
 v_0 &= (15, 00, 00, 00), & v_3 &= (04, 00, 00, 0B), & w_0 &= (14, 00, 00, 00), & w_3 &= (04, 00, 00, 15), \\
 v_1 &= (0D, 00, 00, 00), & v_4 &= (01, 00, 00, 1C), & w_1 &= (0E, 00, 00, 00), & w_4 &= (02, 00, 00, 08), \\
 v_2 &= (03, 00, 00, 00), & v_5 &= (00, 00, 1A, 00), & w_2 &= (01, 00, 00, 00), & w_5 &= (00, 00, 12, 00), \\
 & & v_6 &= (00, 00, 07, 00). & & & w_6 &= (00, 00, 0E, 00).
 \end{aligned}$$

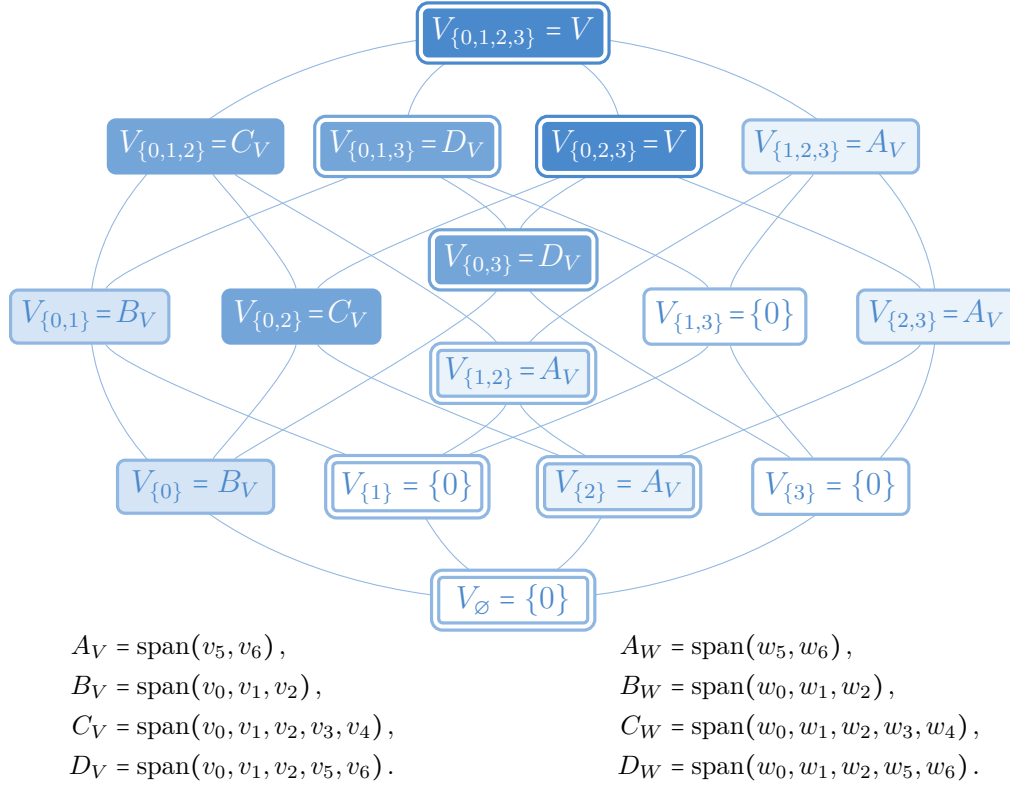


Figure 3.10: The subspaces V_E , W_E for each subset E of $\{0, 1, 2, 3\}$.

Thus, a wall is the Cartesian product of trivial spaces for each S-box. Additionally, if $E \subseteq F$, then $\text{Wall}_E \subseteq \text{Wall}_F$ and hence $V_E \subseteq V_F$ and $W_E \subseteq W_F$.

The subspaces Wall_E are essential in the study of the substitution layer because the latter always preserves the partition $\mathcal{L}(\text{Wall}_E)$ regardless of its S-boxes. This result, together with Proposition 3.20, establishes the following corollary.

Corollary 3.36. Let E be a subset of $\llbracket 0, m \rrbracket$. If σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, then σ also maps $\mathcal{L}(V_E)$ to $\mathcal{L}(W_E)$.

Example 3.37. All the subspaces V_E are graphically represented in Figure 3.10. For instance,

$$V_{\{0\}} = \text{span}((15, 00, 00, 00), (0D, 00, 00, 00), (03, 00, 00, 00)).$$

Additionally, this figure also highlights the expected inclusions given by Remark 3.35. Observe that $\mathcal{B}_V = (v_i)_{0 \leq i < 7}$ is a basis of V . This new basis is more convenient than the standard basis \mathcal{E}_V previously introduced in Example 3.28 since all the V_E

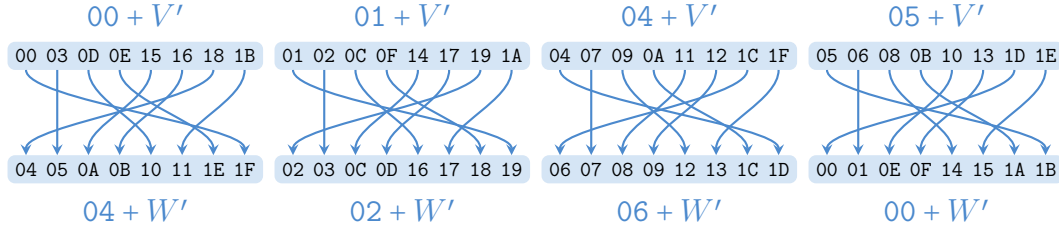


Figure 3.11: The S-box S_0 mapping $\mathcal{L}(V')$ to $\mathcal{L}(W')$ where $V' = \text{span}(03, 0D, 15)$ and $W' = \text{span}(01, 0E, 14)$.

are then easily described. It is worth noting that the same picture remains valid for the subspace W . For example,

$$W_{\{0\}} = \text{span}((14, 00, 00, 00), (0E, 00, 00, 00), (01, 00, 00, 00)).$$

This emphasizes that when the substitution layer maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, the subspaces V and W have the same structure.

According to corollary 3.36, the substitution layer maps $\mathcal{L}(V_{\{0\}})$ to $\mathcal{L}(W_{\{0\}})$. Next, truncate to $E = \{0\}$ using Proposition 3.31 to obtain

$$S_0(\mathcal{L}(\text{span}(03, 0D, 15))) = \mathcal{L}(\text{span}(01, 0E, 14)).$$

This property is depicted in Figure 3.11. Finally, it should be underlined that with Proposition 3.31 alone, no property can be established on the S-box S_0 (see Example 3.33). ▲

Definition 3.38 (Projection P_E). Let E be a subset of $\llbracket 0, m \rrbracket$. The *projection* P_E from $(\mathbb{F}_2^n)^m$ onto Wall_E is defined to be $P_E(x_0, \dots, x_{m-1}) = (y_0, \dots, y_{m-1})$ where $y_i = x_i$ if i belongs to E and $y_i = 0_n$ otherwise.

Remark 3.39. It is not hard to see that P_E is a linear mapping and that V_E is always a subspace of $P_E(V)$. Moreover, it holds that $T_E(V) = T_E(P_E(V))$.

The next lemma gives some relations between the previous definitions. It is quite important and will be used several times by the end of the current chapter.

Lemma 3.40. Let \mathcal{I} be a partition of $\llbracket 0, m \rrbracket$. Then V equals the internal direct sum $\bigoplus_{I \in \mathcal{I}} V_I$ if and only if $V_I = P_I(V)$ for any part I of \mathcal{I} . In this case, the decomposition of an element v of V is $v = \sum_{I \in \mathcal{I}} P_I(v)$.

Proof. Suppose that $V = \bigoplus_{I \in \mathcal{I}} V_I$. Let I be a part of \mathcal{I} . Since V_I is always included in $P_I(V)$, only $P_I(V) \subseteq V_I$ needs to be verified. Let $v = (v_0, \dots, v_{m-1})$ be an element of V . We must prove that $P_I(v)$ lies in V_I . By hypothesis, v can be uniquely written as $\sum_{J \in \mathcal{I}} v_J$ where v_J belongs to V_J . For every i in I , we have

$$(P_I(v))_i = v_i = \sum_{J \in \mathcal{I}} (v_J)_i = (v_I)_i,$$

since $(v_J)_i = 0_n$ for all part J of \mathcal{I} distinct from I . As $P_I(v)_i = 0_n = (v_I)_i$ for each i in I^c , we have $P_I(v) = v_I$. Thus, $P_I(v)$ is included in V_I .

Conversely, suppose that $V_I = P_I(V)$ for all I in \mathcal{I} . Let v be an element of V . Clearly, $v = \sum_{I \in \mathcal{I}} P_I(v)$. By hypothesis, $P_I(v)$ belongs to V_I for any I in \mathcal{I} . The uniqueness of this decomposition directly follows from the definition of the V_I . Therefore, $V = \bigoplus_{I \in \mathcal{I}} V_I$. \blacksquare

Remark 3.41. Suppose that \mathcal{I} is a partition of $\llbracket 0, m \rrbracket$ such that $V = \bigoplus_{I \in \mathcal{I}} V_I$. The previous lemma, together with Remark 3.39, establishes that $T_I(V) = T_I(V_I)$ for each part I of \mathcal{I} .

Proposition 3.42 (Substitution layer structure). Let \mathcal{I} be a partition of $\llbracket 0, m \rrbracket$ satisfying both $V = \bigoplus_{I \in \mathcal{I}} V_I$ and $W = \bigoplus_{I \in \mathcal{I}} W_I$. The permutation σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ if and only if σ_I maps $\mathcal{L}(T_I(V))$ to $\mathcal{L}(T_I(W))$ for any I in \mathcal{I} .

Proof. The implication follows from Proposition 3.31. Conversely, suppose that σ_I maps $\mathcal{L}(T_I(V))$ to $\mathcal{L}(T_I(W))$ for any I in \mathcal{I} . First, let us prove that V and W have the same number of elements. Let I be a part of \mathcal{I} . Since σ_I maps $\mathcal{L}(T_I(V))$ to $\mathcal{L}(T_I(W))$, Proposition 3.21 states that $T_I(V)$ and $T_I(W)$ are isomorphic. Then $T_I(V_I)$ and $T_I(W_I)$ are isomorphic by Remark 3.41. It is not hard to see that the restriction of T_I to V_I is one-to-one. Therefore, $T_I(V_I)$ is isomorphic to V_I and similarly, $T_I(W_I)$ is isomorphic to W_I . Gathering together these results, we deduce that V_I and W_I are isomorphic for each part I of \mathcal{I} . Consequently, $V = \bigoplus_{I \in \mathcal{I}} V_I$ and $W = \bigoplus_{I \in \mathcal{I}} W_I$ have the same dimension, and thus the same number of elements.

To prove that σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, it is sufficient to show that the equality $\sigma(x + V) = \sigma(x) + W$ holds for any element x of \mathbb{F}_2^{nm} thanks to Lemma 3.14. Hence, let x belong to \mathbb{F}_2^{nm} . The discussion above implies that $\sigma(x + V)$ and $\sigma(x) + W$ have the same cardinality. Thus, we just need to verify that $\sigma(x + V) \subseteq \sigma(x) + W$. Let v be any element of V . By hypothesis and by Lemma 3.18, for each part I of \mathcal{I} , there exists t_I in $T_I(W)$ such that

$$\sigma_I(T_I(x) + T_I(v)) = \sigma_I(T_I(x)) + t_I.$$

Observe that for any index $0 \leq i < m$, denoting by $[i]$ the unique part of \mathcal{I} containing i , we have the following:

$$\begin{aligned} \sigma(x + v)_i &= \sigma_{[i]}(T_{[i]}(x) + T_{[i]}(v))_i = \sigma_{[i]}(T_{[i]}(x))_i + (t_{[i]})_i \\ &= \sigma(x)_i + (t_{[i]})_i. \end{aligned}$$

Then, define $w = (w_0, \dots, w_{m-1})$ by $w_i = (t_{[i]})_i$. This yields the equality

$$\sigma(x + v) = \sigma(x) + w.$$

It remains to explain why w lies in W . By hypothesis, $W = \bigoplus_{I \in \mathcal{I}} W_I$.

Because t_I is in $T_I(W)$, there exists w'_I in W such that $T_I(w'_I) = t_I$. As $T_I = T_I \circ P_I$, we have $t_I = T_I(P_I(w'_I)) = T_I(w_I)$ with $w_I = P_I(w'_I)$. Next, Lemma 3.40 states that $W_I = P_I(W)$ for any part I of \mathcal{I} thus w_I belongs to W_I . Finally, it is easy to see that $w = \sum_{I \in \mathcal{I}} w_I$, and hence w belongs to W . Summarizing, this proves that $\sigma(x + V)$ is included in $\sigma(x) + W$, as desired. \blacksquare

The preceding proposition establishes that the converse of Implication 3.2 (page 88) holds whenever the partition \mathcal{I} satisfies both $V = \bigoplus_{I \in \mathcal{I}} V_I$ and $W = \bigoplus_{I \in \mathcal{I}} W_I$. For such a partition, the problem of finding all the substitution layers σ mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$ can equivalently be broken down into the independent sub-problems of finding all the σ_I mapping $\mathcal{L}(T_I(V))$ to $\mathcal{L}(T_I(W))$ for each part I of \mathcal{I} .

3.3.3. Linked and Independent S-Boxes

Of course, there may be several partitions \mathcal{I} such that $V = \bigoplus_{I \in \mathcal{I}} V_I$ and $W = \bigoplus_{I \in \mathcal{I}} W_I$, each yielding a different decomposition of the substitution layer. A few of these decompositions are certainly more interesting or easier to solve. The purpose of this section is to study such partitions. Let us begin with the following lemma.

Lemma 3.43. Suppose that σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. For every partition \mathcal{I} of $\llbracket 0, m \rrbracket$, $V = \bigoplus_{I \in \mathcal{I}} V_I$ if and only if $W = \bigoplus_{I \in \mathcal{I}} W_I$.

Proof. Let \mathcal{I} be a partition of $\llbracket 0, m \rrbracket$. Suppose that $V = \bigoplus_{I \in \mathcal{I}} V_I$. Firstly, let us prove that $W = \sum_{I \in \mathcal{I}} W_I$. Since the W_I are subspaces of W , the inclusion $\sum_{I \in \mathcal{I}} W_I \subseteq W$ clearly holds. Now, let us prove the converse inclusion. Let w be an element of W . Define $x = \sigma^{-1}(0_{nm}) = (S_i^{-1}(0_n))_{0 \leq i < m}$. According to Lemma 3.18, we have

$$\sigma(x + V) = \sigma(x) + W = \sigma(\sigma^{-1}(0_{nm})) + W = W.$$

Hence, there exists an element v of V satisfying the equality $\sigma(x + v) = w$. Then, Lemma 3.40 ensures that $v = \sum_{I \in \mathcal{I}} P_I(v)$. For any $0 \leq i < m$, we have

$$\sigma(x + P_I(v))_i = S_i(x_i + P_I(v)_i) = \begin{cases} S_i(x_i + v_i) & \text{if } i \in I, \\ S_i(x_i + 0_n) = 0_n & \text{if } i \in I^c. \end{cases}$$

Consequently, $\sigma(x + P_I(v))$ lies in Wall_I and W , so in W_I . Note that

$$w = \sigma(x + v) = \sum_{I \in \mathcal{I}} \sigma(x + P_I(v))$$

since \mathcal{I} is a partition of $\llbracket 0, m \rrbracket$. The inclusion $W \subseteq \sum_{I \in \mathcal{I}} W_I$ follows. Finally, the definition of the W_I implies that $W = \bigoplus_{I \in \mathcal{I}} W_I$.

Conversely, suppose that $W = \bigoplus_{I \in \mathcal{I}} W_I$. Following the previous reasoning with σ^{-1} instead of σ gives the equality $V = \bigoplus_{I \in \mathcal{I}} V_I$, as desired. ■

The contrapositive of Lemma 3.43 is the following: if there exists a partition \mathcal{I} such that $V = \bigoplus_{I \in \mathcal{I}} V_I$ and $W \neq \bigoplus_{I \in \mathcal{I}} W_I$ or such that $V \neq \bigoplus_{I \in \mathcal{I}} V_I$ and $W = \bigoplus_{I \in \mathcal{I}} W_I$, then there exists no substitution layer mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Because we intend to study the substitution layers mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$, Lemma 3.43 suggests to assume the following.

Assumption 3.44. For the remainder of this section, we assume that for any partition \mathcal{I} of $\llbracket 0, m \rrbracket$, it holds that

$$V = \bigoplus_{I \in \mathcal{I}} V_I \iff W = \bigoplus_{I \in \mathcal{I}} W_I.$$

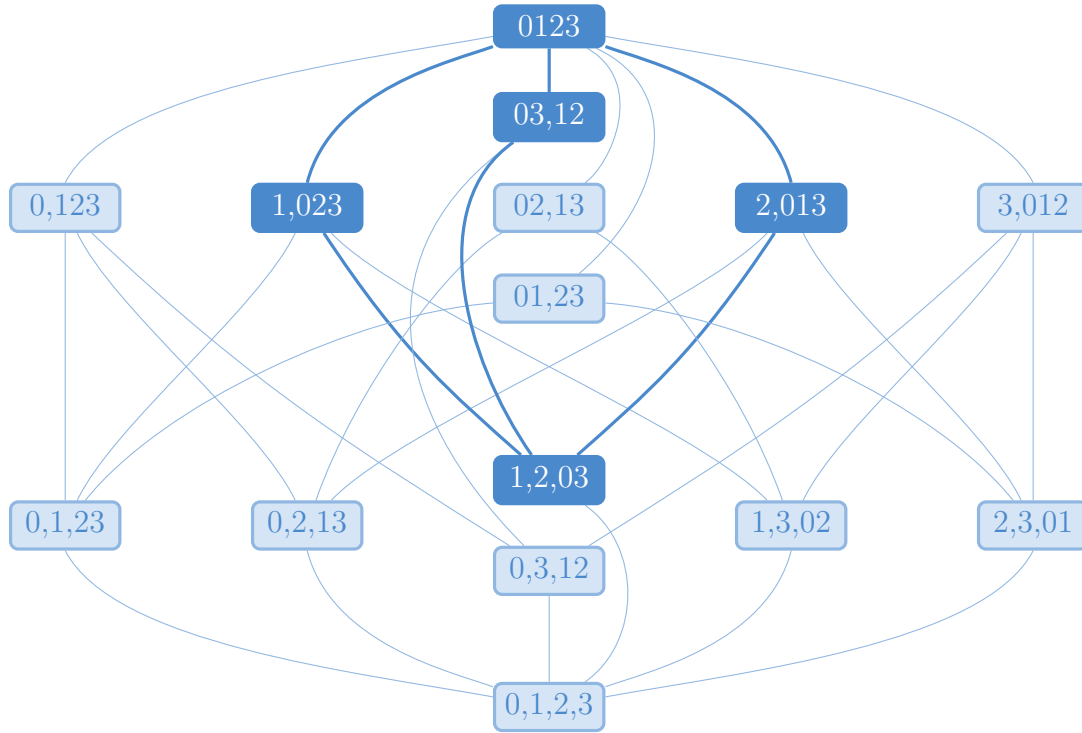


Figure 3.12: The partitions \mathcal{I} of $\{0, 1, 2, 3\}$ such that $V = \bigoplus_{I \in \mathcal{I}} V_I$.

Proposition 3.42, together with the preceding assumption, suggests the following definition.

Definition 3.45 (Decomposition Partition). A *decomposition partition* (with respect to V and W) is a partition of $\llbracket 0, m \rrbracket$ such that $V = \bigoplus_{I \in \mathcal{I}} V_I$.

Remark 3.46 (Partial Order on Partitions). Recall that if \mathcal{I} and \mathcal{J} are two partitions of $\llbracket 0, m \rrbracket$, then the partition \mathcal{I} is said to be *finer* than \mathcal{J} if for any part I in \mathcal{I} , there exists a part J in \mathcal{J} such that $I \subseteq J$.

Example 3.47. The purpose of this example is to find all the decomposition partitions with regards to V and W . By virtue of Lemma 3.40, the subspace V can be decomposed as $\bigoplus_{I \in \mathcal{I}} V_I$ if and only if V_I is equal to $P_I(V)$ for each part I of \mathcal{I} . The eight framed subspaces in the middle of Figure 3.10 are exactly those which satisfy $V_E = P_E(V)$. Hence, the decomposition partitions are the partitions whose parts are selected from the following:

$$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{0, 3\}, \{0, 1, 3\}, \{0, 2, 3\}, \{0, 1, 2, 3\}.$$

It is then easy to check that the decomposition partitions of V are:

$$\begin{aligned} &\{\{1\}, \{2\}, \{0, 3\}\}, \quad \{\{1\}, \{0, 2, 3\}\}, \quad \{\{2\}, \{0, 1, 3\}\}, \\ &\quad \{\{0, 3\}, \{1, 2\}\} \quad \text{and} \quad \{\{0, 1, 2, 3\}\}. \end{aligned}$$

In Figure 3.12, all the partitions of $\llbracket 0, 4 \rrbracket$ are ordered by the “*finer-than*” relation and the decomposition partitions are emphasized. What stands out is that the decomposition partition $\{\{1\}, \{2\}, \{0, 3\}\}$ is finer than all other decomposition partitions. ▀

The existence of this least decomposition partition in the example above is a very welcome and non-trivial property. This means that all the truncated substitution layers obtained using Proposition 3.42 are the smallest possible. Thus, such a partition should be preferred to any other decomposition partition. We will now prove that this least decomposition partition always exists.

Notation 3.48 (Partition Intersection). Let \mathcal{I} and \mathcal{J} be two partitions of $\llbracket 0, m \rrbracket$. We denote by $\mathcal{I} \cap \mathcal{J}$ the set $\{I \cap J \mid (I, J) \in \mathcal{I} \times \mathcal{J} \setminus \{\emptyset\}\}$. Note that $\mathcal{I} \cap \mathcal{J}$ is a partition of $\llbracket 0, m \rrbracket$ finer than \mathcal{I} and \mathcal{J} .

Lemma 3.49. Let \mathcal{I} and \mathcal{J} be two partitions of $\llbracket 0, m \rrbracket$ such that $V = \bigoplus_{I \in \mathcal{I}} V_I = \bigoplus_{J \in \mathcal{J}} V_J$. Then, $V = \bigoplus_{K \in \mathcal{I} \cap \mathcal{J}} V_K$.

Proof. Let K be a part of $\mathcal{I} \cap \mathcal{J}$. According to Lemma 3.40, we only have to prove that $P_K(V) = V_K$. Clearly, $V_K \subseteq P_K(V)$. Thus, it remains to show that $P_K(V) \subseteq V_K$. Since or more precisely that $P_K(V) \subseteq V_K$. By definition, there exists parts I and J of \mathcal{I} and \mathcal{J} such that $K = I \cap J$. Let v be an element of V . Since $V = \bigoplus_{I' \in \mathcal{I}} V_{I'}$, Lemma 3.40 ensures that $P_I(v)$ lies in V_I , hence in V . In the same way, using the relation $V = \bigoplus_{J' \in \mathcal{J}} V_{J'}$, we deduce that $P_J(P_I(v))$ lies in V_J , so in V . Note that $P_J(P_I(v)) = P_{I \cap J}(v) = P_K(v)$. Therefore, $P_K(v)$ belongs to $V \cap \text{Wall}_K = V_K$. ■

Proposition 3.50. The set of the partitions \mathcal{I} of $\llbracket 0, m \rrbracket$ satisfying $V = \bigoplus_{I \in \mathcal{I}} V_I$ has a least element denoted \mathcal{I}_{ld} .

Proof. Let \mathcal{P} denote the set of all the partitions \mathcal{I} of $\llbracket 0, m \rrbracket$ satisfying $V = \bigoplus_{I \in \mathcal{I}} V_I$. By virtue of Lemma 3.49, the set \mathcal{P} is closed under the operation of intersection. Then, it is sufficient to define \mathcal{I}_{ld} to be the intersection of all the elements of \mathcal{P} . ■

Consequently, the only decomposition partition that will be considered in the remainder of this chapter is the least decomposition partition \mathcal{I}_{ld} . The following definition is inspired by Proposition 3.42 and Proposition 3.50.

Definition 3.51 (Linked and independent S-boxes). Suppose that σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Let I be a part of \mathcal{I}_{ld} .

- If $I = \{i\}$, the S-box S_i is said to be *independent* of the other S-boxes. Moreover, if $V_{\{i\}} = \{0_{nm}\}$ or $V_{\{i\}} = \text{Wall}_{\{i\}}$, the S-box S_i is said to be *inactive*. Otherwise, S_i is *active*.
- If $\#I \geq 2$, then the S-boxes whose indices lie in I are said to be *linked together*.

Remark 3.52. Let $0 \leq i < m$ be an integer. We have already noted that the substitution layer σ always preserves $\mathcal{L}(\{0_{nm}\})$ and $\mathcal{L}(\text{Wall}_{\{i\}})$. In addition, Proposition 3.42 ensures that σ maps $\mathcal{L}(V_{\{i\}})$ to $\mathcal{L}(W_{\{i\}})$. Consequently, if $V_{\{i\}} = \{0_{nm}\}$ or if $V_{\{i\}} = \text{Wall}_{\{i\}}$, then $V_{\{i\}} = W_{\{i\}}$.

Suppose that the S-box S_i is independent with regards to the subspaces V and W . As established by Proposition 3.42 and Remark 3.41, if S_i is replaced with another S-box S'_i , then this new substitution layer still maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ provided that S'_i maps $\mathcal{L}(T_{\{i\}}(V_{\{i\}}))$ to $\mathcal{L}(T_{\{i\}}(W_{\{i\}}))$.

Suppose further that S_i is active. By definition, $\{0_{nm}\} \not\subseteq V_{\{i\}} \not\subseteq \text{Wall}_{\{i\}}$. Observe that the restriction of $T_{\{i\}}$ to $\text{Wall}_{\{i\}}$ is one-to-one, hence

$$\{0_n\} = T_{\{i\}}(\{0_{nm}\}) \not\subseteq T_{\{i\}}(V_{\{i\}}) \not\subseteq T_{\{i\}}(\text{Wall}_{\{i\}}) = \mathbb{F}_2^n.$$

Thus, $T_{\{i\}}(V_{\{i\}})$ is a non-trivial subspace of \mathbb{F}_2^n and the requirement that S'_i maps $\mathcal{L}(T_{\{i\}}(V_{\{i\}}))$ to $\mathcal{L}(T_{\{i\}}(W_{\{i\}}))$ is also non-trivial. Therefore, an independent active S-box can be chosen independently of the other S-boxes but has to respect the structure of the subspaces V and W .

Now suppose that S_i is inactive. By definition, $V_{\{i\}} = \{0_{nm}\}$ or $V_{\{i\}} = \text{Wall}_{\{i\}}$. Then the equality $V_{\{i\}} = W_{\{i\}}$ follows from Remark 3.52 and we have

$$T_{\{i\}}(V_{\{i\}}) = T_{\{i\}}(W_{\{i\}}) = \{0_n\} \quad \text{or} \quad T_{\{i\}}(V_{\{i\}}) = T_{\{i\}}(W_{\{i\}}) = \mathbb{F}_2^n.$$

In either case, the condition that S'_i maps $\mathcal{L}(T_{\{i\}}(V_{\{i\}}))$ to $\mathcal{L}(T_{\{i\}}(W_{\{i\}}))$ is trivial and any S-box fulfills it. As a consequence, an independent inactive S-box can be freely chosen. In other words, such an S-box has no impact on the fact that σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$.

Finally, suppose that some S-boxes are linked together. If only one of these S-boxes is replaced independently of the others, then the desired property of the substitution layer may not hold.

Example 3.53. As we have seen in Example 3.47 and Figure 3.12, the least decomposition partition with regards to the subspaces V and W is $\mathcal{I}_{\text{ld}} = \{\{1\}, \{2\}, \{0, 3\}\}$. By Proposition 3.42, the substitution layer maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ if and only if the following equalities hold:

$$\sigma_{\{0,3\}}(\mathcal{L}(T_{\{0,3\}}(V))) = \mathcal{L}(T_{\{0,3\}}(W)), \quad \begin{aligned} S_1(\mathcal{L}(T_{\{1\}}(V))) &= \mathcal{L}(T_{\{1\}}(W)), \\ S_2(\mathcal{L}(T_{\{2\}}(V))) &= \mathcal{L}(T_{\{2\}}(W)). \end{aligned}$$

Thus, the S-box S_1 is independent of the other S-boxes, the same applies to S_2 and the S-boxes S_0 and S_3 are linked together. As it was already noted in Figure 3.10, we have

$$V_{\{1\}} = \{(00, 00, 00, 00)\} \quad \text{and} \quad V_{\{2\}} = \text{span}((00, 00, 1A, 00), (00, 00, 07, 00)).$$

Therefore, the S-box S_2 is active while S_1 is inactive. ▀

3.3.4. The Forbidden Case

Throughout this section, we assume that the substitution layer σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. In order to prove the last main theorem of this chapter, we need to consider the following particular case.

Proposition 3.54. Let \mathcal{I} be a decomposition partition. Let I be a part of \mathcal{I} such that $\#I \geq 2$ and let E be a non-empty proper subset of I . Suppose that $V_E = V_{I \setminus E} = \{0_{nm}\}$ and $P_E(V) = \text{Wall}_E$. Then, for all i in E , S_i is an affine mapping.

If the subspace V satisfies the assumption of the proposition above, then at least one of S-boxes has to be affine. Nowadays, an SPN whose substitution layer has an affine S-box cannot be taken seriously. Additionally, such a cipher is likely to be very weak to differential and linear cryptanalysis. This discussion explains the title of this section.

Example 3.55. As seen in Example 3.47, the least decomposition partition is $\mathcal{I}_{\text{ld}} = \{\{1\}, \{2\}, \{0, 3\}\}$. Its only part of cardinality greater than or equal to 2 is $I = \{0, 3\}$. The non-empty proper subsets of I are the $E = \{0\}$ and $E = \{1\}$. According to Figure 3.10, we have $V_{\{0\}} \neq \{0_{20}\}$. Consequently, Proposition 3.54 does not apply to this example and this is good news because none of the S-boxes is affine. Otherwise, this would have disproved the contrapositive of Proposition 3.54.

Now let us introduce another example. Consider a substitution layer σ' made up of two 3-bit S-boxes S'_0 and S'_1 , hence its parameters are $m = 2$ and $n = 3$. Define the subspaces V' and W' of $(\mathbb{F}_2^3)^2$ to be

$$V' = W' = \text{span}((4, 4), (2, 2), (1, 1)) = \{(x, x) \mid x \in \mathbb{F}_2^3\}.$$

Finally, suppose that σ' maps $\mathcal{L}(V')$ to $\mathcal{L}(W')$. It is easily seen that

$$\begin{aligned} V'_\emptyset &= \{(0, 0)\}, & V'_{\{0\}} &= \{(0, 0)\}, & V'_{\{1\}} &= \{(0, 0)\}, & V'_{\{0,1\}} &= V, \\ P_\emptyset(V') &= \text{Wall}_\emptyset, & P_{\{0\}}(V') &= \text{Wall}_{\{0\}}, & P_{\{1\}}(V') &= \text{Wall}_{\{1\}}, & P_{\{0,1\}}(V') &= V. \end{aligned}$$

Thus, the least decomposition partition with regards to V' and W' is $\{\{0, 1\}\}$. The S-boxes S'_0 and S'_1 are then linked together. Choosing $E = \{0\}$ in Proposition 3.54 ensures that S'_0 must be affine. Similarly, we can prove that S'_1 must also be affine by considering $E = \{1\}$. As a result, any substitution layer σ' mapping $\mathcal{L}(V')$ to $\mathcal{L}(W')$ is necessarily affine. These subspaces are thus completely prohibited as the whole cipher is then affine. ▀

The rest of this section is devoted to the proof of Proposition 3.54.

Lemma 3.56. Let E be a subset of $\llbracket 0, m \rrbracket$. Suppose that $V_E = V_{E^c} = \{0_{nm}\}$ and $P_E(V) = \text{Wall}_E$. Then $W_E = W_{E^c} = \{0_{nm}\}$ and $T_E(V) = T_E(W) = (\mathbb{F}_2^n)^p$ with $p = \#E$.

Proof. Recall that σ maps $\mathcal{L}(V_E)$ to $\mathcal{L}(W_E)$. Then, Proposition 3.21 states that V_E and W_E are isomorphic, so $W_E = \{0_{nm}\}$. By a similar argument, we obtain the equality $W_{E^c} = \{0_{nm}\}$. Now, it is easy to see that $T_E = T_E \circ P_E$. Hence, $T_E(V) = T_E(P_E(V)) = T_E(\text{Wall}_E) = (\mathbb{F}_2^n)^p$ where p denotes $\#E$. By Proposition 3.31, σ_E maps $\mathcal{L}(T_E(V))$ to $\mathcal{L}(T_E(W))$. It follows that $T_E(V)$ and $T_E(W)$ are isomorphic and $T_E(W)$ is also equal to $(\mathbb{F}_2^n)^p$. ■

Lemma 3.57. Let E be a subset of $\llbracket 0, m \rrbracket$. Then $\#V = \#T_E(V) \times \#V_{E^c}$.

Proof. Let p denote $\#E$. Consider the restriction of the linear mapping T_E to V . Its kernel is

$$\text{Ker}(T_E) = \{v \in V \mid T_E(v) = 0_{np}\} = \{v \in V \mid \forall i \in E, v_i = 0_n\} = V_{E^c}.$$

From the first isomorphism theorem, the quotient space V/V_{E^c} is isomorphic to the image $T_E(V)$. Particularly, the equality $\#V/\#V_{E^c} = \#T_E(V)$ holds. \blacksquare

Lemma 3.58. Let $E = \llbracket 0, p \rrbracket$ with $0 \leq p < m$. Suppose that $V_E = V_{E^c} = \{0_{nm}\}$ and $T_E(V) = (\mathbb{F}_2^n)^p$. There exist two isomorphisms $\varphi : T_E(V) \rightarrow T_{E^c}(V)$ and $\psi : T_E(W) \rightarrow T_{E^c}(W)$ such that

$$V = \{[y \parallel \varphi(y)] \mid y \in (\mathbb{F}_2^n)^p\} \quad \text{and} \quad W = \{[z \parallel \psi(z)] \mid z \in (\mathbb{F}_2^n)^p\}.$$

Proof. Lemma 3.57 ensures that $\#V = \#T_E(V) \times \#V_{E^c}$. By hypothesis, $V_{E^c} = \{0_{nm}\}$, so $\#V_{E^c} = 1$. It follows that $\#V = \#T_E(V)$. Therefore, V and $T_E(V)$ have the same dimension d . Let $\mathcal{B} = (b^{[i]})_{0 \leq i < d}$ be a basis of $T_E(V)$. By definition, there exists a family $(c^{[i]})_{0 \leq i < d}$ of vectors in V such that $T_E(c^{[i]}) = b^{[i]}$. That is, $c^{[i]} = [b^{[i]} \parallel T_{E^c}(c^{[i]})]$. Note that the vectors $c^{[0]}, \dots, c^{[d-1]}$ are linearly independent as the $b^{[i]}$ are, and thus $(c^{[i]})_{0 \leq i < d}$ is a basis of V . Define the linear mapping $\varphi : T_E(V) \rightarrow T_{E^c}(V)$ by the equalities $\varphi(b^{[i]}) = T_{E^c}(c^{[i]})$ for every $0 \leq i < d$. Let v be an element of V . Since $(c^{[i]})_{0 \leq i < d}$ is a basis of V , the vector v can be written as $v = \sum_{i=0}^{d-1} \lambda_i c^{[i]}$ where the λ_i are elements of \mathbb{F}_2 . Next,

$$\begin{aligned} v &= \sum_{i=0}^{d-1} \lambda_i c^{[i]} = \sum_{i=0}^{d-1} \lambda_i [b^{[i]} \parallel T_{E^c}(c^{[i]})] = \sum_{i=0}^{d-1} \lambda_i [b^{[i]} \parallel \varphi(b^{[i]})] \\ &= \left[\sum_{i=0}^{d-1} \lambda_i b^{[i]} \parallel \sum_{i=0}^{d-1} \lambda_i \varphi(b^{[i]}) \right] = \left[\sum_{i=0}^{d-1} \lambda_i b^{[i]} \parallel \varphi\left(\sum_{i=0}^{d-1} \lambda_i b^{[i]}\right) \right] = [y \parallel \varphi(y)] \end{aligned}$$

where y denotes the element $\sum_{i=0}^{d-1} \lambda_i b^{[i]}$ of $T_E(V)$. Consequently, every element of V can be written in the desired form. As the converse inclusion is clear from the definition of φ , the equality $V = \{[y \parallel \varphi(y)] \mid y \in (\mathbb{F}_2^n)^p\}$ follows. Hence, the mapping φ is onto. Applying Lemma 3.57 with the subset E^c gives $\#V = \#T_{E^c}(V) \times \#V_E = \#T_{E^c}(V)$, and thus $T_{E^c}(V)$ is also a d -dimensional subspace. Therefore, φ is an isomorphism. Because of Lemma 3.56, our assumptions about V are also hold for W . Thus, the same argument yields an isomorphism $\psi : T_E(W) \rightarrow T_{E^c}(W)$ satisfying $W = \{[z \parallel \psi(z)] \mid z \in (\mathbb{F}_2^n)^p\}$. \blacksquare

Lemma 3.59. Let p be a non-negative integer and let $f : (\mathbb{F}_2^n)^p \rightarrow (\mathbb{F}_2^n)^p$. Suppose that there exists a mapping $g : (\mathbb{F}_2^n)^p \rightarrow (\mathbb{F}_2^n)^p$ satisfying $f(x + y) = f(x) + g(y)$ for all x and y in $(\mathbb{F}_2^n)^p$. Then f is an affine mapping.

Proof. Let y be an element of $(\mathbb{F}_2^n)^p$. Choosing $x = 0_{np}$ yields $f(0_{np} + y) = f(0_{np}) + g(y)$, and thus $g(y) = f(y) + f(0_{np})$. Therefore, the equalities

$$\begin{aligned} f(x + y) + f(0_{np}) &= f(x) + g(y) + f(0_{np}) = f(x) + (f(y) + f(0_{np})) + f(0_{np}) \\ &= (f(x) + f(0_{np})) + (f(y) + f(0_{np})) \end{aligned}$$

hold for all x and y in $(\mathbb{F}_2^n)^p$. This proves that the mapping $x \mapsto f(x) + f(0_{np})$ is linear. The result follows. \blacksquare

Lemma 3.60. Let I be a part of a decomposition partition and let E be a subset of I . The following equalities hold:

$$P_E(T_I(V)) = T_I(P_E(V)) \quad \text{and} \quad (T_I(V))_E = T_I(V_E).$$

Remark 3.61. The statement of the lemma above is an abuse of notation. The domain of the projection P_E on the left side of the first equality is $(\mathbb{F}_2^n)^I$ whereas the domain of P_E on the other side is $(\mathbb{F}_2^n)^m$. Similarly, $(T_I(V))_E$ denotes the set $T_I(V) \cap \text{Wall}_E^I$ where $\text{Wall}_E^I = \{x \in (\mathbb{F}_2^n)^I \mid \forall i \in I \setminus E, x_i = 0_n\}$ is the wall associated with E in $(\mathbb{F}_2^n)^I$.

Proof. Let x be an element of $(\mathbb{F}_2^n)^m$. We have

$$T_I(P_E(x)) = T_I((\delta_{i \in E} \cdot x_i)_{i < m}) = (\delta_{i \in E} \cdot x_i)_{i \in I} = P_E((x_i)_{i \in I}) = P_E(T_I(x)).$$

Thus, $T_I \circ P_E = P_E \circ T_I$ and the first equality follows.

Observe that $\text{Wall}_E \subseteq \text{Wall}_I$ because $E \subseteq I$. Thus, $\text{Wall}_I \cap \text{Wall}_E = \text{Wall}_E$. Since I is a part of a decomposition partition, Lemma 3.40 implies that $P_I(V) = V_I$. Hence, we have

$$\begin{aligned} V_E &= V \cap \text{Wall}_E = V \cap (\text{Wall}_I \cap \text{Wall}_E) = (V \cap \text{Wall}_I) \cap \text{Wall}_E = V_I \cap \text{Wall}_E \\ &= P_I(V) \cap \text{Wall}_E. \end{aligned}$$

Note that $P_I(V)$ and Wall_E are two subsets of Wall_I . In addition, the restriction of T_I to Wall_I is clearly one-to-one. Therefore,

$$T_I(V_E) = T_I(P_I(V) \cap \text{Wall}_E) = T_I(P_I(V)) \cap T_I(\text{Wall}_E) = T_I(V) \cap \text{Wall}_E^I = T_I(V)_E,$$

The result is proven. \blacksquare

We have now all the tools needed to prove Proposition 3.54. For convenience, we recall its statement.

Let \mathcal{I} be a decomposition partition. Let I be a part of \mathcal{I} such that $\#I \geq 2$ and let E be a non-empty proper subset of I . Suppose that $V_E = V_{I \setminus E} = \{0_{nm}\}$ and $P_E(V) = \text{Wall}_E$. Then, for all i in E , S_i is an affine mapping.

Proof (of Proposition 3.54). Denote m' the cardinality of I . Define $\sigma' = \sigma_I$, $V' = T_I(V)$ and $W' = T_I(W)$. Proposition 3.31 establishes that σ' maps $\mathcal{L}(V')$ to $\mathcal{L}(W')$. Then, Lemma 3.60 states that

$$V'_E = (T_I(V))_E = T_I(V_E) = T_I(\{0_{nm}\}) = \{0_{nm'}\}.$$

Similarly, $V'_{I \setminus E} = \{0_{nm'}\}$ and $P_E(V') = \text{Wall}_E$. Consequently, we can assume without loss of generality that $\mathcal{I} = \llbracket 0, m \rrbracket$ and $I = \llbracket 0, m \rrbracket$.

Even if it means to change the order of the S-boxes and the bundles of the spaces V and W , we can assume that $E = \llbracket 0, p \rrbracket$ with $0 < p < m$, and hence $E^c = \llbracket p, m \rrbracket$.

Thus, every element x of \mathbb{F}_2^{nm} can be written as $[T_E(x) \parallel T_{E^c}(x)]$. Define $q = m - p$. Firstly, note that $T_E(V) = T_E(W) = (\mathbb{F}_2^n)^p$ by Lemma 3.56. According to Lemma 3.58, there exist two isomorphisms $\varphi : T_E(V) \rightarrow T_{E^c}(V)$ and $\psi : T_E(W) \rightarrow T_{E^c}(W)$ such that

$$V = \{[y \parallel \varphi(y)] \mid y \in (\mathbb{F}_2^n)^p\} \quad \text{and} \quad W = \{[z \parallel \psi(z)] \mid z \in (\mathbb{F}_2^n)^p\}.$$

Let g be the permutation of $(\mathbb{F}_2^n)^p$ defined by the formula

$$g(y) = \psi^{-1}(\sigma_{E^c}(\varphi(y)) + \sigma_{E^c}(0_{nq})).$$

Before going any further, we should explain why g is well-defined. Let y be an element of $(\mathbb{F}_2^n)^p = T_E(V)$. First, $\varphi(y)$ belongs to $T_{E^c}(V)$. By Proposition 3.31, σ_{E^c} maps $\mathcal{L}(T_{E^c}(V))$ to $\mathcal{L}(T_{E^c}(W))$. Then, the part $T_{E^c}(V)$ is mapped to $\sigma_{E^c}(0_{nq}) + T_{E^c}(W)$ according to Lemma 3.18. Therefore, $\sigma_{E^c}(\varphi(y)) + \sigma_{E^c}(0_{nq})$ lies in $T_{E^c}(W)$, and thus ψ^{-1} brings it back to $T_E(W) = (\mathbb{F}_2^n)^p$.

Let x be an element of $(\mathbb{F}_2^n)^p$. From Lemma 3.18, we have the following:

$$\sigma([x \parallel 0_{nq}] + V) = \sigma([x \parallel 0_{nq}]) + W.$$

On one hand,

$$\begin{aligned} \sigma([x \parallel 0_{nq}] + V) &= \sigma(\{[x \parallel 0_{nq}] + [y \parallel \varphi(y)] \mid y \in (\mathbb{F}_2^n)^p\}) \\ &= \sigma(\{[x + y \parallel \varphi(y)] \mid y \in (\mathbb{F}_2^n)^p\}) \\ &= \{[\sigma_E(x + y) \parallel \sigma_{E^c}(\varphi(y))] \mid y \in (\mathbb{F}_2^n)^p\}. \end{aligned}$$

On the other hand,

$$\begin{aligned} \sigma([x \parallel 0_{nq}]) + W &= \{\sigma([x \parallel 0_{nq}]) + [z \parallel \psi(z)] \mid z \in (\mathbb{F}_2^n)^p\} \\ &= \{[\sigma_E(x) \parallel \sigma_{E^c}(0_{nq})] + [z \parallel \psi(z)] \mid z \in (\mathbb{F}_2^n)^p\} \\ &= \{[\sigma_E(x) + z \parallel \sigma_{E^c}(0_{nq}) + \psi(z)] \mid z \in (\mathbb{F}_2^n)^p\}. \end{aligned}$$

Let y be an element of $(\mathbb{F}_2^n)^p$. Since $[\sigma_E(x + y) \parallel \sigma_{E^c}(\varphi(y))]$ belongs to the part $\sigma([x \parallel 0_{nq}]) + W$, there exists z in $(\mathbb{F}_2^n)^p$ satisfying the following two equations:

$$\begin{cases} \sigma_E(x + y) = \sigma_E(x) + z, \\ \sigma_{E^c}(\varphi(y)) = \sigma_{E^c}(0_{nq}) + \psi(z). \end{cases} \quad (3.3)$$

The bottom equation can be restated as $z = g(y)$. Combining with the top equation, we see that

$$\sigma_E(x + y) = \sigma_E(x) + g(y).$$

Since this equality holds for all x and y in $(\mathbb{F}_2^n)^p$, Lemma 3.59 states that the truncated substitution layer σ_E is an affine mapping.

Now, it remains to prove that all the S-boxes involved in σ_E are affine mappings. Let i be an element of E . The mapping $I_i : \mathbb{F}_2^n \rightarrow (\mathbb{F}_2^n)^m$, $x \mapsto (\delta_{i,0}x, \dots, \delta_{i,m-1}x)$ is clearly linear (where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise). Observe that $S_i = \sigma_{\{i\}} = T_{\{i\}}\sigma_E I_i$. Therefore, the S-box S_i is the composition of several affine (or linear) mappings, and hence, is itself an affine mapping. ■

3.3.5. Reduction to one S-Box

To prove our main result about the substitution layer, we need the following two preliminary lemmas.

Lemma 3.62. Let I be an element of \mathcal{I}_{ld} . Let E be a non-empty proper subset of I . Then $V_E \not\subseteq P_E(V)$ and $P_E(V) \neq \{0_{nm}\}$.

Proof. By construction, V_E is a subset of $P_E(V)$. Let us prove that $V_E \neq P_E(V)$. By contradiction, suppose that $V_E = P_E(V)$. Let v be an element of V . By hypothesis, $P_E(v)$ belongs to V_E . Especially, $P_E(v)$ lies in V , so $v + P_E(v)$ also lies in V . Since $v + P_E(v) = P_{E^c}(v)$, we deduce that $P_{E^c}(v)$ belongs to V_{E^c} . Let \mathcal{J} denotes the partition $\{E, E^c\}$. Lemma 3.40 states that $V = \bigoplus_{J \in \mathcal{J}} V_J$. Then, $V = \bigoplus_{K \in \mathcal{I}_{\text{ld}} \cap \mathcal{J}} V_K$ follows from Lemma 3.49. Observe that the partition $\mathcal{I}_{\text{ld}} \cap \mathcal{J}$ is strictly finer than \mathcal{I}_{ld} because E is a proper subset of I . This is a contradiction, and therefore $V_E \not\subseteq P_E(V)$.

By contradiction, suppose that $P_E(V) = \{0_{nm}\}$. From the previous result, we obtain $\{0_{nm}\} \subseteq V_E \not\subseteq P_E(V) = \{0_{nm}\}$, which is a contradiction. Thus, $P_E(V) \neq \{0_{nm}\}$. ■

Lemma 3.63. Let I be a part of \mathcal{I}_{ld} and E be a non-empty proper subset of I .

- If V_E is a wall, then $V_E = \text{Wall}_\emptyset = \{0_{nm}\}$.
- If $P_E(V)$ is a wall, then $P_E(V) = \text{Wall}_E$.

Proof. By contradiction, suppose that V_E is any wall different from $\{0_{nm}\}$. Hence, there exists a non-empty subset F of E such that $V_E = \text{Wall}_F$. Therefore $\text{Wall}_F \subseteq V$ and so $\text{Wall}_F = \text{Wall}_F \cap V = V_F$. Next, $\text{Wall}_F = V_F \subseteq P_F(V) \subseteq \text{Wall}_F$, and thus $V_F = P_F(V)$. Since F is a non-empty proper subset of I , we have a contradiction with Lemma 3.62. Consequently, $V_E = \{0_{nm}\}$.

By contradiction, suppose that $P_E(V)$ is any wall different from Wall_E . There exists a proper subset F of E such that $P_E(V) = \text{Wall}_F$. Thus, for every v in V and every i in $E \setminus F$, $P_E(v)_i = 0_n$. As a consequence, $P_{E \setminus F}(V) = \{0_{nm}\}$. This is a contradiction with Lemma 3.62 because $E \setminus F$ is a non-empty proper subset of I . The result follows. ■

Now we have all the results needed, let us state and prove the main result of Section 3.3 which is depicted in Figure 3.13.

Theorem 3.64. Let $n \geq 2$ and m be two positive integers. Let S_0, \dots, S_{m-1} be n -bit S-boxes. Define the permutation σ of $(\mathbb{F}_2^n)^m$ which maps the element $(x_i)_{0 \leq i < m}$ to $(S_i(x_i))_{0 \leq i < m}$. Let V and W be two subspaces of $(\mathbb{F}_2^n)^m$ such that σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Suppose that V is not a wall. Then, at least one of the S-boxes maps a non-trivial linear partition to another one.

Proof. Let us prove this result by complete induction on the number m of S-boxes. Suppose that $m = 1$. In this case, $\sigma = S_0$. By hypothesis, V is different from $\{0_n\}$ and \mathbb{F}_2^n . Hence, $\mathcal{L}(V)$ is a non-trivial partition and S_0 maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$.

Let $m \geq 2$ be an integer. Suppose that the result holds for any positive integer strictly less than m . Firstly, suppose that all the S-boxes are independent. In

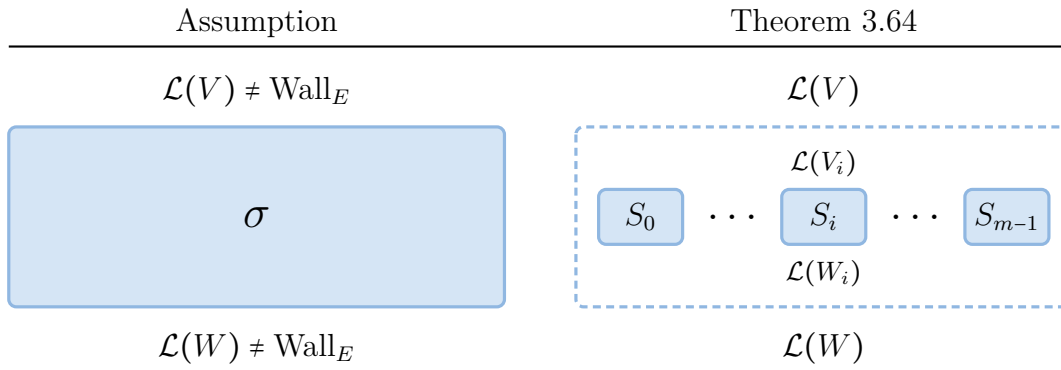


Figure 3.13: Diagrammatic representation of Theorem 3.64.

other words, $\mathcal{I}_{\text{ld}} = \{\{i\} \mid i \in \llbracket 0, m \rrbracket\}$. If each S-box is inactive, then V is a wall, a contradiction with our hypothesis. Thus, there exists at least one active S-box S_i . In this case, $\{0_{nm}\} \subsetneq V_{\{i\}} \subsetneq \text{Wall}_{\{i\}}$. According to Lemma 3.40, the equality $P_{\{i\}}(V) = V_{\{i\}}$ holds. Then, $T_{\{i\}}(V_{\{i\}}) = T_{\{i\}}(P_{\{i\}}(V)) = T_{\{i\}}(V)$ is a non-trivial subspace of \mathbb{F}_2^n , so $\mathcal{L}(T_{\{i\}}(V))$ is also non-trivial. Finally, Proposition 3.31 states that S_i maps $\mathcal{L}(T_{\{i\}}(V))$ to $\mathcal{L}(T_{\{i\}}(W))$, and thus the result holds in this case.

Now, suppose that some S-boxes are linked together. Then, there exists an element I of \mathcal{I}_{ld} such that $\#I \geq 2$. Next, at least one of the following three cases holds.

1. Suppose that there exists a non-empty proper subset E of I such that $P_E(V)$ is not a wall. Let p denote the cardinality of E . Recall that $T_E(P_E(V)) = T_E(V)$. It follows that $T_E(V)$ is not a wall of $(\mathbb{F}_2^n)^p$. According to Proposition 3.31, σ_E maps $\mathcal{L}(T_E(V))$ to $\mathcal{L}(T_E(W))$. Note that E is a non-empty proper subset of I , so of $\llbracket 0, m \rrbracket$. Hence $p < m$, so the induction hypothesis ensures that at least one of the S-boxes of σ_E maps a non-trivial partition to another one.
2. Suppose that there exists a non-empty proper subset E of I such that V_E is not a wall. Recall that σ maps $\mathcal{L}(V_E)$ to $\mathcal{L}(W_E)$. Proposition 3.31 ensures that σ_E maps $\mathcal{L}(T_E(V_E))$ to $\mathcal{L}(T_E(W_E))$. It is easily seen that $T_E(V_E)$ is not a wall. As before, the result is a consequence of the induction hypothesis.
3. Suppose that there exists a non-empty proper subset E of I such that $P_E(V)$, V_E and $V_{I \setminus E}$ are all walls. Then, Lemma 3.63 implies that $P_E(V) = \text{Wall}_E$ and $V_E = V_{I \setminus E} = \{0_{nm}\}$. According to Proposition 3.54, the S-boxes whose indices belong to E are affine mappings. Combining Proposition 3.25 and 3.23, we see that these S-boxes map any non-trivial linear partition to another one.

In any case, the result holds for this integer m . The result follows by induction. ■

Example 3.65. It is worthwhile to note that the proof of Theorem 3.64 is constructive. Therefore, it gives a method to find necessary conditions on the S-boxes for the substitution layer to map $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Let us apply this method to our main example.

The first step is equivalent to what had been done in Examples 3.47 and 3.53. Consider the least decomposition partition $\mathcal{I}_{\text{ld}} = \{\{1\}, \{2\}, \{0, 3\}\}$ and deduce that:

- S_1 is inactive;
- S_2 is active and maps $\mathcal{L}(\text{span}(07, 1A))$ to $\mathcal{L}(\text{span}(0E, 12))$ (see Figure 3.3);
- S_0 and S_3 are linked together.

Now, consider the part $I = \{0, 3\}$ of \mathcal{I}_{Id} . Thus, the non-empty proper subsets of I are $\{0\}$ and $\{3\}$. The first case requires to compute the following projections:

$$P_{\{0\}}(V) = \text{Wall}_{\{0\}} \quad \text{and} \quad P_{\{3\}}(V) = \text{span}((00, 00, 00, 0B), (00, 00, 00, 1C)).$$

Thus, $P_{\{3\}}(V)$ is not a wall. As in Example 3.33 and Figure 3.9, we see that S_3 maps $\mathcal{L}(0B, 1C)$ to $\mathcal{L}(08, 15)$ by truncating σ and the subspaces $P_{\{3\}}(V)$, $P_{\{3\}}(W)$ to $\{3\}$. Now, we need to compute the following subspaces:

$$V_{\{0\}} = \text{span}((03, 00, 00, 00), (0D, 00, 00, 00), (15, 00, 00, 00)) \quad \text{and} \quad V_{\{3\}} = \text{Wall}_{\emptyset}.$$

Since $V_{\{0\}}$ is not a wall, the second case applies. Then, truncate the substitution layer σ and the subspaces $V_{\{0\}}$ and $W_{\{0\}}$ to prove that S_0 maps $\mathcal{L}(03, 0D, 15)$ to $\mathcal{L}(01, 0E, 14)$. This property was stressed in Example 3.37 and Figure 3.10. Finally, recall that the third case does not apply to these subspaces, as observed in Example 3.55. ▀

The preceding example covers only the first and the second cases in the treatment of linked S-boxes given by the proof of Theorem 3.64. To illustrate the third case, we introduced the following example.

Example 3.66. Let $n = m = 3$. Thus, the substitution layer σ is made up of three 3-bit S-boxes denoted by S_0 , S_1 and S_2 . Define the subspaces V and W of $(\mathbb{F}_2^3)^3$ to be

$$V = W = \{(x, y, x + y) \mid x, y \in \mathbb{F}_2^3\}$$

and assume that the substitution layer σ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. By definition, it holds that $P_{\emptyset}(V) = \{(0, 0, 0)\}$ and $P_{\{0,1,2\}}(V) = V$. Then, for each non-empty proper subset E of $\{0, 1, 2\}$, it is easily seen that $P_E(V) = \text{Wall}_E$. For instance,

$$P_{\{0,1\}}(V) = \{(x, y, 0) \mid x, y \in \mathbb{F}_2^3\} = \text{Wall}_{\{0,1\}}.$$

We know that $V_{\emptyset} = \{(0, 0, 0)\}$ and $V_{\{0,1,2\}}(V) = V$. The other subspaces V_E are the following:

$$\begin{aligned} V_{\{0\}} &= \{(0, 0, 0)\}, & V_{\{1\}} &= \{(0, 0, 0)\}, & V_{\{2\}} &= \{(0, 0, 0)\}, \\ V_{\{0,1\}} &= \{(x, x, 0) \mid x \in \mathbb{F}_2^3\}, & V_{\{0,2\}} &= \{(x, 0, x) \mid x \in \mathbb{F}_2^3\}, & V_{\{1,2\}} &= \{(0, x, x) \mid x \in \mathbb{F}_2^3\}. \end{aligned}$$

Thus, the equality $P_E(V) = V_E$ holds only for $E = \emptyset$ and $E = \{0, 1, 2\}$. Consequently, the least decomposition partition is $\mathcal{I}_{\text{Id}} = \{\{0, 1, 2\}\}$, and hence all the S-boxes are linked together.

From now on, we follow the method given in the proof of Theorem 3.64. As previously noted, for each non-empty proper subset E of $\{0, 1, 2\}$, the projection $P_E(V)$ is a wall. Therefore, the first case does not apply to this example. We move on to the second case. By induction, the substitution layer and the subspaces

$V_{\{0,1\}}$ and $W_{\{0,1\}}$ are truncated to $\{0,1\}$. Hence, we now consider the permutation $\sigma' = \sigma_{\{0,1\}}$ which maps $\mathcal{L}(V')$ to $\mathcal{L}(W')$ where

$$V' = W' = T_{\{0,1\}}(V_{\{0,1\}}) = \{(x, x) \mid x \in \mathbb{F}_2^3\}.$$

Such a substitution layer has already been studied in Example 3.55. Recall that

$$\begin{aligned} V'_\emptyset &= \{(0, 0)\}, & V'_{\{0\}} &= \{(0, 0)\}, & V'_{\{1\}} &= \{(0, 0)\}, & V'_{\{0,1\}} &= V, \\ P_\emptyset(V') &= \text{Wall}_\emptyset, & P_{\{0\}}(V') &= \text{Wall}_{\{0\}}, & P_{\{1\}}(V') &= \text{Wall}_{\{1\}}, & P_{\{0,1\}}(V') &= V. \end{aligned}$$

Thus, the least decomposition partition with regards to V' and W' is $\{\{0,1\}\}$. Since $V'_{\{0\}}, V'_{\{1\}}, P_{\{0\}}(V')$ and $P_{\{1\}}(V')$ are all walls, the first and second cases do not apply. Choosing $E = \{0\}$ and $E = \{1\}$ in the third case proves that S_0 and S_1 are affine mappings. Come back to the full substitution layer. Similarly, it is straightforward to verify that S_2 must be affine by truncating σ and the subspaces $V_{\{0,2\}}, W_{\{0,2\}}$ to $\{0,2\}$. To summarize, we have proven that any substitution layer mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$ is necessarily affine. ▲

3.4. Conclusion

In this chapter, we have studied a generic SPN mapping a partition of the plaintexts to a partition of the ciphertexts, independently of the round keys used. Combining Theorem 3.26 and Corollary 3.27, we proved that there exist two families $(V^{[i]})_{0 \leq i \leq r}$ and $(W^{[i]})_{0 \leq i < r}$ of subspaces such that the substitution layer σ maps $\mathcal{L}(V^{[i]})$ to $\mathcal{L}(W^{[i]})$ for each $0 \leq i < r$. This result has been illustrated in Figure 3.6.

First, suppose that all the $V^{[i]}$ are walls. In such a case, the diffusion layer of the cipher is probably not playing its role (or the round number is very small). As it is generally the case, suppose that there is no diffusion layer in the last round of the SPN. Then, the input and the output partitions are both linear partitions associated with walls. This implies that some ciphertext bundles are independent of some plaintext bundles. Such a property must be avoided in any good cipher. To characterize a diffusion layer which does not have this weakness, Calderini introduced the following definition in [23].

Definition 3.67 (Strongly Proper r -Round Diffusion Layer). The diffusion layer π is said to be *strongly proper over r rounds* if for each proper wall W , there exists an integer $1 \leq i < r$ such that $\pi^i(W)$ is not a wall.

Assuming that the diffusion layer of the SPN is strongly proper over r -rounds, at least one of the $V^{[i]}$ is not a wall. This second case is far more interesting than the previous one. By virtue of Theorem 3.64, at least one of the S-boxes must map a non-trivial linear partition to another one, as illustrated in Figure 3.13.

Thus, we have proven in this chapter that any partition-based backdoor SPN with a strongly proper diffusion layer has at least one S-box mapping a non-trivial linear partition to another one. The following chapter aims at designing such an S-box with the best security against differential and linear cryptanalysis.

Analysis of a Backdoor S-Box

In the preceding chapter, we have considered a generic substitution-permutation network together with a partition-based backdoor holding independently of the key schedule. Assuming that its diffusion layer is strongly proper, we have proven that at least one S-box must map a linear partition to another one, thereby reducing the study of the whole cipher to the study of one single S-box.

As said in introduction, differential and linear cryptanalysis are considered as the most important attacks against block ciphers, and therefore any new cipher should be proven secure against these two attacks. Since the S-boxes are the only primitives of an SPN which are not affine, they must provide sufficient resistance to make the whole cipher secure. On the other hand, the diffusion layer aims at enhancing the confusion provided by the substitution layer. But even with a carefully designed diffusion layer, an SPN which has poor S-boxes is unlikely to achieve performance and security. To summarize, if we want our backdoor cipher to be secure against these attacks, then we must design an S-box mapping a linear partition to another one while providing good differential and linear properties. This is the purpose of this chapter.

Firstly, Section 4.1 explains how an S-box mapping a linear partition to another one can be associated with an imprimitive S-box which has the same properties with respect to differential and linear cryptanalysis. Then, we recall a fundamental decomposition result of imprimitive S-boxes. Secondly, Section 4.2 relates the linear and differential properties of an imprimitive S-box to the ones of its decomposition. Following these results, we derive an algorithm to design strong S-boxes mapping a linear partition to another one. Next, a toy partition-based backdoor cipher is given in Section 4.3. This example illustrates the results of the previous and this chapters. Lastly, we discuss ways to prevent partition-based backdoor ciphers in Section 4.4. The content of this chapter was published in the same papers as the previous chapter, that is in [9] and [12].

4.1. Structure of a Backdoor S-Box

Optimal differential and linear resistances of vectorial Boolean functions are generally studied by means of equivalence relations preserving their properties. Following the terminology introduced in [20], the three widely used equivalence relations are

affine-equivalence, *EA-equivalence* (Extended Affine) and *CCZ-equivalence* (Carlet-Charpin-Zinoviev [34, Proposition 3]), sorted here from the least to the most general. In our treatment of backdoor S-boxes we will use the simplest, namely the affine-equivalence.

Recall that two permutations S_1 and S_2 of \mathbb{F}_2^n are said to be *affine-equivalent* if there exist two linear mappings L_1, L_2 of \mathbb{F}_2^n and two elements v_1, v_2 of \mathbb{F}_2^n such that

$$\forall x \in \mathbb{F}_2^n, \quad S_2(x) = L_2(S_1(L_1(x) + v_1)) + v_2. \quad (4.1)$$

It is well known that affine-equivalent S-boxes have the same security against differential and linear cryptanalysis [80, Proposition 1]. Indeed, assuming that S_1 and S_2 are affine-equivalent, it is straightforward to prove that for all a, b in \mathbb{F}_2^n , we have

$$\text{DP}_{S_2}(a, b) = \text{DP}_{S_1}(L_1(a), L_2^{-1}(b)), \quad (4.2)$$

$$\text{LP}_{S_2}(a, b) = \text{LP}_{S_1}((L_1^{-1})^\top(a), L_2^\top(b)), \quad (4.3)$$

see for instance [27, Proposition 2.16]. The first relation means that their differential probability matrices are equal up to row and column permutations. The second is its analogous for linear potentials. More precisely, their correlation matrices are linked by

$$C_{S_2}(a, b) = (-1)^{\langle a, L_1^{-1}(v_1) \rangle + \langle b, v_2 \rangle} C_{S_1}((L_1^{-1})^\top(a), L_2^\top(b)). \quad (4.4)$$

Thus, they are equal up to row and column permutations and up to the signs of their coefficients.

Coming back to partition-based backdoor S-boxes, let V and W be two subspaces of \mathbb{F}_2^n and suppose that S' is an n -bit S-Box mapping the partition $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Then there exists an automorphism L of \mathbb{F}_2^n mapping the subspace V to W , as ensured by Proposition 3.21. Naturally its inverse L^{-1} maps W to V and Proposition 3.25 states that L^{-1} maps the partition $\mathcal{L}(W)$ to $\mathcal{L}(V)$. Finally, the S-box S defined to be $L^{-1} \circ S'$ is by construction affine-equivalent to S' and preserves the partition $\mathcal{L}(V)$. This discussion establishes the following Proposition.

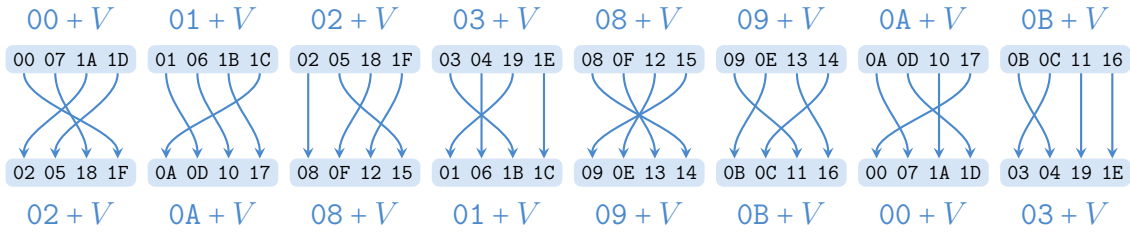
Proposition 4.1. Let V and W be two subspaces of \mathbb{F}_2^n and let S' be an n -bit S-box mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. There exists an affine-equivalent S-box S to S' preserving $\mathcal{L}(V)$.

Remark 4.2. Conversely, suppose that S preserves the partition $\mathcal{L}(V)$. Let W be any subspace isomorphic to V and denote by L an isomorphism from V to W . By Proposition 3.25, the composite $L \circ S$ maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ and is obviously affine-equivalent to S .

Example 4.3. Let us consider the 5-bit S-box S' given in Figure 4.1. This S-box has already been met in Examples 3.17 and 3.28 (denoted by f and S_2 respectively). Thus, we know that S' maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ where

$$V = \text{span}(07, 1A) \quad \text{and} \quad W = \text{span}(0E, 12).$$

		.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
$S'(x)$	0.	1E	08	04	13	0F	18	14	10	19	15	0E	0D	03	1C	07	17
	1.	12	11	0B	1B	09	05	1F	00	0A	01	02	1A	06	0C	1D	16
$L^{-1}(x)$	0.	00	01	02	03	08	09	0A	0B	0D	0C	0F	0E	05	04	07	06
	1.	18	19	1A	1B	10	11	12	13	15	14	17	16	1D	1C	1F	1E
$S(x)$	0.	1F	0D	08	1B	06	15	10	18	14	11	07	04	03	1D	0B	13
	1.	1A	19	0E	16	0C	09	1E	00	0F	01	02	17	0A	05	1C	12

 Figure 4.1: Construction of the S-Box S used throughout Chapter 4.

 Figure 4.2: The permutation S preserving $\mathcal{L}(V)$ where $V = \text{span}(07, 1A)$.

Referring back to Example 3.22, we end up with an automorphism L of \mathbb{F}_2^5 satisfying $L(V) = W$. Its inverse L^{-1} and the composite $S = L^{-1}S'$ are given in Figure 4.1. For instance, $S(07) = L^{-1}(S'(07)) = L^{-1}(10) = 18$.

By construction, this new permutation S is equivalent to S' and preserves the linear partition $\mathcal{L}(V)$, as can be seen in Figure 4.2. The similarity between Figures 3.3 and 4.2 is striking, thereby clarifying the choices we have made when constructing the automorphism L in Example 3.22. This S-box S will be studied throughout this chapter. ▀

As a consequence of Proposition 4.1, it can be assumed without loss of generality that the subspaces V and W are equal when studying the linear and differential properties of an S-box mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Therefore, we consider the following in the remainder of this section:

- let V be a d -dimensional non-trivial subspace of \mathbb{F}_2^n ,
- let U be a complement space of V ,
- let S be an n -bit S-box preserving $\mathcal{L}(V)$.

Since U is a complement subspace of V , the space \mathbb{F}_2^n is equal to the direct sum $U \oplus V$. In other words, every element x of \mathbb{F}_2^n can be uniquely written as the sum $x = u + v$ where u and v belong to U and V respectively. Let $[u]$ denote the coset of V with respect to u . Thus, $[u] = u + V$ is the unique part of $\mathcal{L}(V)$ containing u and we have

$$\mathcal{L}(V) = \{[u] \mid u \in U\}.$$

Since V is d -dimensional, the complement space U is $(n - d)$ -dimensional. In addition, we have the inequalities

$$1 \leq d \leq n - 1 \quad \text{and} \quad 1 \leq n - d \leq n - 1$$

because V is assumed to be a non-trivial subspace of \mathbb{F}_2^n .

The following theorem describes the structure of permutations preserving a linear partition. A similar result has been introduced by Harpes in his thesis [50, Theorem 5.6]. However, our statement will be more appropriate for the next results of this chapter.

Theorem 4.4 (Decomposition of an Imprimitive S-Box). There exist a unique permutation ρ of U and a unique family of permutations $(\tau_u)_{u \in U}$ of V such that, for all $x = u + v$ in \mathbb{F}_2^n ,

$$S(u + v) = \rho(u) + \tau_u(v) .$$

Conversely, if ρ is a permutation of U and if $(\tau_u)_{u \in U}$ is a family of permutations of V , then the mapping S' defined by the rule $S'(u + v) = \rho(u) + \tau_u(v)$ preserves $\mathcal{L}(V)$.

As will be seen in Section 4.1.3, this theorem is a corollary of Krasner-Kaloujnine embedding theorem [65] (see Theorem 4.13). But for convenience, we give below a direct proof.

Proof. By hypothesis, S preserves $\mathcal{L}(V)$. Thus, S induces a permutation ρ of U defined as follows. Let u be an element of U . Hence, there exists a unique u' in U such as $S([u]) = [u']$. Define then $\rho(u) = u'$. For each element u of U , define the permutation τ_u of V which maps v to $S(u + v) + \rho(u)$. By construction, for any u in U and any v in V we have

$$\tau_u(v) = S(u + v) + \rho(u) \quad \text{and hence} \quad S(u + v) = \rho(u) + \tau_u(v) .$$

The existence of the permutations ρ and τ_u is proven. Now, let us show their uniqueness. Suppose that there exist a permutation $\tilde{\rho}$ of U and a family of permutations $(\tilde{\tau}_u)_{u \in U}$ of V satisfying the result. Let (u, v) be an element of $U \times V$. By hypothesis, we have the relation

$$\rho(u) + \tau_u(v) = \tilde{\rho}(u) + \tilde{\tau}_u(v) .$$

Because the sum of U and V is direct, it follows that $\rho(u) = \tilde{\rho}(u)$ and $\tau_u(v) = \tilde{\tau}_u(v)$. The uniqueness of ρ and the τ_u follows.

Conversely, let ρ be a permutation of U and $(\tau_u)_{u \in U}$ be a family of permutations of V . Denote by S' the mapping from \mathbb{F}_2^n to \mathbb{F}_2^n defined by the rule $S'(u + v) = \rho(u) + \tau_u(v)$. Because ρ and the τ_u are permutations of U and V respectively the mapping S' is a permutation of $U \oplus V = \mathbb{F}_2^n$. For every element u of U , it holds that

$$\begin{aligned} S'([u]) &= \{S'(u + v) \mid v \in V\} = \{\rho(u) + \tau_u(v) \mid v \in V\} \\ &= \rho(u) + \{\tau_u(v) \mid v \in V\} = \rho(u) + V = [\rho(u)] . \end{aligned}$$

Hence, S' preserves the linear partition $\mathcal{L}(V)$, as desired. ■

This theorem is of great significance as it yields a general construction for imprimitive S-boxes using permutations with smaller domains. Intuitively, this result can be explained as follows. We already know that the linear partition $\mathcal{L}(V)$ consists

of cosets of V . By permuting the elements of each coset $[u]$, the whole partition is left unchanged. The way we permute the elements of $[u]$ is represented by the permutation τ_u of V , namely $u + v$ is mapped to $u + \tau_u(v)$. Therefore, we need a family $(\tau_u)_{u \in U}$ to represent all these local permutations. Up to this point, each coset is mapped to itself, so it remains to explain how the cosets are permuted. This is the role of the permutation ρ of U . Thus the coset $[u]$ is mapped as a whole to $[\rho(u)]$, that is to say, each element $u + \tau_u(v)$ is mapped to $\rho(u) + \tau_u(v)$. To summarize, the family $(\tau_u)_{u \in U}$ describes how the elements are moved inside each part and the permutation ρ tells us how S permutes the parts of the partition $\mathcal{L}(V)$.

Example 4.5. Again, we consider the S-box S introduced in Example 4.3. Define the following complement subspace of V :

$$U = \text{span}(01, 02, 08) = \{00, 01, 02, 03, 08, 09, 0A, 0B\}.$$

Figure 4.2 shows that S induces a permutation ρ of U . For instance, $\rho(01) = 0A$ because S maps the coset $[01]$ to $[0A]$. Next, for each u in U , define the permutation τ_u of V by the rule $\tau_u(v) = S(u + v) + \rho(u)$. For instance,

$$\tau_{01}(07) = S(01 + 07) + \rho(01) = S(06) + \rho(01) = 10 + 0A = 1A.$$

This decomposition of S is illustrated in Figure 4.3. For example, the image of 06 under S is computed as follows. First, 06 is uniquely written as the sum $u + v = 01 + 07$ of an element u in U with an element v of V . Hence, 06 lies in the coset

$$[01] = 01 + V = 01 + \{00, 07, 1A, 1D\} = \{01, 06, 1B, 1C\}.$$

The first step consists in moving 06 inside the coset $[01]$, or equivalently modifying its component in V . Thus, 06 is mapped to $01 + \tau_{01}(07) = 01 + 1A = 1B$. The second step permutes the coset representatives in U , and hence leaves the elements of same coset in the same relative positions. The element $1B = 01 + 1A$ is then mapped to $\rho(01) + 1A = 0A + 1A = 10$. Therefore, the S-box S maps 06 to 10. ▲

The next three subsections aim at explaining how Theorem 4.4 can be seen as a corollary of Krasner-Kaloujnine embedding theorem. Since this result has already been proven, the reader can jump directly to Section 4.2 for a first reading. Nevertheless, this connection highlights the group structure behind the decomposition of an S-box preserving a linear partition, bringing us back to imprimitive groups.

4.1.1. Wreath Product

As direct products for finite-dimensional vector spaces, wreath products naturally arise when studying imprimitive permutation groups. The result justifying this analogy is known as Krasner-Kaloujnine embedding theorem. Informally, it establishes that every imprimitive permutation group can be embedded into a wreath product, namely seen as a subgroup of a wreath product up to isomorphism. Roughly speaking, this result implies that wreath products are the biggest imprimitive groups.

In this section, we recall the definition of wreath products. However, it turns out that there are several ways to define wreath products. Several groups are involved



in this structure and each of them can act on the right or on the left. The following choices are made in order to derive Theorem 4.4 from Krasner-Kaloujnine embedding theorem. Because some of them are very unusual, all the results will be proven. To see other classic representations of wreath products, readers may refer to [42, 26, 6, 55]. Wreath products are based on the so-called semidirect products recalled below.

Definition 4.6 (Outer Semidirect Product). Let N and H be two groups and let ϕ be a homomorphism from H to $\text{Aut}(N)$. To simplify the reading, let \star denote the group law of N and let ϕ_h denote the automorphism of N associated with an element h of H . The *semidirect product* of H and N , denoted by $H \ltimes_{\phi} N$, is the group formed by the Cartesian product $H \times N$ together with the following binary operation

$$(h_1, n_1) \otimes (h_2, n_2) = (h_1 h_2, \phi_{h_2^{-1}}(n_1) \star n_2).$$

Remark 4.7. As explained in Definition 3.1, an action of a group G on X can be defined to be a homomorphism from G to $\text{Sym}(X)$. The homomorphism ϕ is then an action of H on N whose image is included in the subgroup $\text{Aut}(N)$ of $\text{Sym}(N)$. In this case, the group H is said to act *by automorphisms* on N . Finally, it should be noted that when this action is trivial (namely $\phi_h = \text{Id}_N$ for every h in H), the semidirect product of H and N is just their direct product.

Proof. First, let us prove that (e_H, e_N) is the identity element of $H \ltimes_{\phi} N$. Let (h, n) be an element of $H \times N$. It holds that

$$\begin{aligned} (h, n) \otimes (e_H, e_N) &= (h e_H, \phi_{e_H^{-1}}(n) \star e_N) = (h, \phi_{e_H}(n)) = (h, \text{Id}(n)) = (h, n), \\ (e_H, e_N) \otimes (h, n) &= (e_H h, \phi_{h^{-1}}(e_N) \star n) = (h, e_N \star n) = (h, n). \end{aligned}$$

Thus, (e_H, e_N) is the identity element. Now, let us prove that $(h^{-1}, \phi_h(N^{-1}))$ is the inverse of (h, n) .

$$\begin{aligned} (h, n) \otimes (h^{-1}, \phi_h(n^{-1})) &= (h h^{-1}, \phi_h(n) \star \phi_h(n^{-1})) = (e_H, \phi_h(n \star n^{-1})) \\ &= (e_H, \phi_h(e_N)) = (e_H, e_N), \\ (h^{-1}, \phi_h(n^{-1})) \otimes (h, n) &= (h^{-1} h, \phi_{h^{-1}}(\phi_h(n^{-1})) \star n) = (e_H, \phi_{h^{-1}h}(n^{-1}) \star n) \\ &= (e_H, \phi_{e_H}(n^{-1}) \star n) = (e_H, n^{-1} \star n) = (e_H, e_N). \end{aligned}$$

Finally, it remains to prove that \otimes is associative. Let (h_1, n_1) , (h_2, n_2) and (h_3, n_3) be three elements of $H \times N$. We have

$$\begin{aligned} (h_1, n_1) \otimes [(h_2, n_2) \otimes (h_3, n_3)] &= (h_1, n_1) \otimes (h_2 h_3, \phi_{h_3^{-1}}(n_2) \star n_3) \\ &= (h_1 h_2 h_3, \phi_{(h_2 h_3)^{-1}}(n_1) \star \phi_{h_3^{-1}}(n_2) \star n_3), \\ [(h_1, n_1) \otimes (h_2, n_2)] \otimes (h_3, n_3) &= (h_1 h_2, \phi_{h_2^{-1}}(n_1) \star n_2) \otimes (h_3, n_3) \\ &= (h_1 h_2 h_3, \phi_{h_3^{-1}}(\phi_{h_2^{-1}}(n_1) \star n_2) \star n_3). \end{aligned}$$

Observe that

$$\begin{aligned} \phi_{h_3^{-1}}(\phi_{h_2^{-1}}(n_1) \star n_2) &= \phi_{h_3^{-1}}(\phi_{h_2^{-1}}(n_1)) \star \phi_{h_3^{-1}}(n_2) = \phi_{h_3^{-1} h_2^{-1}}(n_1) \star \phi_{h_3^{-1}}(n_2) \\ &= \phi_{(h_2 h_3)^{-1}}(n_1) \star \phi_{h_3^{-1}}(n_2). \end{aligned}$$

Consequently, the associativity of the binary operation \otimes follows and the desired result is proven. \blacksquare

Proposition 4.8. Let A be a group and let B be a group acting on a set I . Define $\phi : B \rightarrow \mathfrak{S}(A^I)$ which maps an element b of B to the permutation

$$\begin{aligned}\phi_b : A^I &\longrightarrow A^I \\ (a_i)_{i \in I} &\longmapsto (a_{b^{-1} \cdot i})_{i \in I}.\end{aligned}$$

Then, ϕ is a homomorphism whose image is included in $\text{Aut}(A^I)$.

Remark 4.9. Using the vocabulary introduced in Remark 4.7, the group B acts on the direct product A^I by automorphisms. This action is quite simple as B only permutes the components of the elements of A^I in the natural way. Let $(a_i)_{i \in I}$ be an element of A^I and let $(a'_i)_{i \in I} = \phi_b((a_i)_{i \in I})$ denote its image under the action of a given b in B . The component a_i is moved to the index $b \cdot i$, and hence $a_i = a'_{b \cdot i}$ leading to the equality $(a_i)_{i \in I} = (a'_{b \cdot i})_{i \in I}$. Equivalently, $(a_{b^{-1} \cdot i})_{i \in I} = (a'_i)_{i \in I}$, whence $\phi_b((a_i)_{i \in I}) = (a_{b^{-1} \cdot i})_{i \in I}$. This discussion explains the definition of the action ϕ .

Proof. First, let us prove that ϕ is well-defined. Let b be an element of B . We need to prove that ϕ_b is a permutation of A^I . Let $(a_i)_{i \in I}$ be an element of A^I . We have

$$\phi_{b^{-1}} \circ \phi_b((a_i)_{i \in I}) = \phi_{b^{-1}}((a_{b^{-1} \cdot i})_{i \in I}) = (a_{b^{-1} \cdot (b \cdot i)})_{i \in I} = (a_{b^{-1} b \cdot i})_{i \in I} = (a_{e_B \cdot i})_{i \in I} = (a_i)_{i \in I}.$$

Similarly, $\phi_b \circ \phi_{b^{-1}} = \text{Id}_{A^I}$. As a consequence, ϕ_b is a permutation of A^I and ϕ is well-defined. Now, let us prove that ϕ is a group homomorphism. Let b_1, b_2 be two elements of B and let $(a_i)_{i \in I}$ be an element of A^I . Next,

$$\phi_{b_1} \circ \phi_{b_2}((a_i)_{i \in I}) = \phi_{b_1}((a_{b_2^{-1} \cdot i})_{i \in I}) = (a_{b_2^{-1} b_1^{-1} \cdot i})_{i \in I} = (a_{(b_1 b_2)^{-1} \cdot i})_{i \in I} = \phi_{b_1 b_2}((a_i)_{i \in I}).$$

It follows that $\phi_{b_1} \circ \phi_{b_2} = \phi_{b_1 b_2}$ proving that ϕ is a homomorphism. Let b be an element of B . To prove that the image of ϕ is included in $\text{Aut}(A^I)$, it suffices to show that ϕ_b is a homomorphism. For all $(a_i)_{i \in I}$ and $(a'_i)_{i \in I}$ in A^I , it holds that

$$\begin{aligned}\phi_b((a_i)_{i \in I} \times (a'_i)_{i \in I}) &= \phi_b((a_i a'_i)_{i \in I}) = (a_{b^{-1} \cdot i} a'_{b^{-1} \cdot i})_{i \in I} = (a_{b^{-1} \cdot i})_{i \in I} \times (a'_{b^{-1} \cdot i})_{i \in I} \\ &= \phi_b((a_i)_{i \in I}) \times \phi_b((a'_i)_{i \in I}).\end{aligned}$$

This concludes the proof of our proposition. \blacksquare

Given two groups A and B acting respectively on X and I , their wreath product $A \wr B$ naturally acts on the product $I \times X$. Intuitively, the set $I \times X$ may be thought as a collection of $\#I$ copies of the set X . Similarly, the wreath product $A \wr B$ contains $\#I$ copies of the group A , each acting on its associated copy of X . Then, B acts on $I \times X$ by permuting the copies of X . The resulting action of $A \wr B$ on $I \times X$ is thus imprimitive since it preserves these copies.

Definition 4.10 (Wreath Product). Let A be a group and let B be a group acting on a set I . The *wreath product* of A and B denoted by $A \wr B$, is defined to be the semidirect product $B \ltimes_{\phi} A^I$ where ϕ is given in Proposition 4.8. Explicitly, the group law of $A \wr B$ is given for all $(b, (a_i)_{i \in I})$ and $(b', (a'_i)_{i \in I})$ in $B \times A^I$ by

$$(b, (a_i)_{i \in I}) \otimes (b', (a'_i)_{i \in I}) = (bb', \phi_{b'^{-1}}((a_i)_{i \in I}) \times (a'_i)_{i \in I}) = (bb', (a_{b' \cdot i} a'_i)_{i \in I}).$$

Proposition 4.11 (Imprimitive Action of $A \wr B$). Let A and B be two groups and suppose that A acts on X and that B acts on I . Then, the wreath product $A \wr B$ acts on $I \times X$ by

$$(b, (a_i)_{i \in I}) \cdot (j, x) = (b \cdot j, a_j \cdot x)$$

for all $(b, (a_i)_{i \in I})$ in $B \times A^I$ and (j, x) in $I \times X$. Furthermore, let X_i denote the set $\{i\} \times X$ for each i in I . Then, $A \wr B$ preserves the partition $\{X_i \mid i \in I\}$.

Remark 4.12. In other words, $\mathcal{B} = \{X_i \mid i \in I\}$ is an $(A \wr B)$ -invariant partition of $I \times X$. Since this partition is non-trivial (whenever $\#I, \#X \geq 2$), the wreath product $A \wr B$ acts imprimitively on $I \times X$.

Proof. Let (j, x) be an element of $I \times X$. Observe that

$$(e_B, (e_A)_{i \in I}) \cdot (j, x) = (e_B \cdot j, e_A \cdot x) = (j, x).$$

Thus, the element (j, x) is fixed under the action of the identity element of $A \wr B$. Now, let $(b, (a_i)_{i \in I})$ and $(b', (a'_i)_{i \in I})$ be two elements of $A \wr B$. On the one hand,

$$(b, (a_i)_{i \in I}) \cdot [(b', (a'_i)_{i \in I}) \cdot (j, x)] = (b, (a_i)_{i \in I}) \cdot (b' \cdot j, a'_j \cdot x) = (bb' \cdot j, a'_{b' \cdot j} a'_j \cdot x).$$

On the other hand,

$$[(b, (a_i)_{i \in I}) \cdot (b', (a'_i)_{i \in I})] \cdot (j, x) = (bb', (a_{b' \cdot i} a'_i)_{i \in I}) \cdot (j, x) = (bb' \cdot j, a_{b' \cdot j} a'_j \cdot x).$$

Therefore, $A \wr B$ acts on $I \times X$.

It remains to prove that this action preserves the partition $\{X_i \mid i \in I\}$. Let $g = (b, (a_i)_{i \in I})$ be an element of $A \wr B$ and let j be an element of I . We will prove that $g \cdot X_j = X_{b \cdot j}$. Let $g \cdot (j, x)$ be an element of $g \cdot X_j$. Then $g \cdot (j, x) = (b \cdot j, a_j \cdot x)$ belongs to $X_{b \cdot j}$. Thus, $g \cdot X_j$ is included in $X_{b \cdot j}$. Now, let $(b \cdot j, x)$ be an element of $X_{b \cdot j}$. Then $(j, a_j^{-1} \cdot x)$ belongs to X_j and

$$g \cdot (j, a_j^{-1} \cdot x) = (b \cdot j, a_j a_j^{-1} \cdot x) = (b \cdot j, x).$$

Thus, $(b \cdot j, x)$ lies in $g \cdot X_j$. Therefore, $g \cdot X_j = X_{b \cdot j}$. The desired result follows from Lemma 3.14. ■

4.1.2. Krasner-Kaloujnine Embedding Theorem

It is now time to state Krasner-Kaloujnine embedding theorem which relates every imprimitive group action to the imprimitive action of a wreath product.

Theorem 4.13 (Krasner, Kaloujnine [65]). Let G be an imprimitive permutation group on a set E and let \mathcal{P} be a G -invariant partition of E . Let R be a system of distinct representatives of \mathcal{P} . For each r in R , let $[r]$ denote the unique part of \mathcal{P} containing r . Let r_0 be an element of R .

- Let A be the permutation group on $[r_0]$ induced by the action of the set-wise stabilizer of $[r_0]$ in G .
- Let B be the permutation group on R induced by G .

Then E may be identified with $R \times [r_0]$ in a such way that G on E is permutation isomorphic to a subgroup of $A \wr B$ acting imprimitively on $R \times [r_0]$.

Proof. For each r in R , choose an element t_r of G satisfying $t_r([r_0]) = [r]$. Such elements exist by virtue of Lemma 3.5. Let λ be the mapping from E to $R \times [r_0]$ which maps x to $(r, t_r^{-1}(x))$ where r is the unique element of R such that $x \in [r]$. Finally, let φ denote the mapping from G to $A \wr B$ defined by the formula

$$\varphi(g) = (\bar{g}, (t_{\bar{g}(r)}^{-1} \circ g \circ t_r)_{r \in R})$$

where $\bar{g} : R \rightarrow R$ maps r to the representative of $[g(r)]$ in R . We will prove that (φ, λ) is a permutation isomorphism from G on E to $\varphi(G)$ on $R \times [r_0]$.

Let us begin by showing that λ is a bijection. Since $\#E = \#\mathcal{P} \times \#P_0$, it is sufficient to prove that λ is one-to-one. Let x and x' be two elements of E such that $\lambda(x) = \lambda(x')$. Denote r and r' the elements of R satisfying $x \in [r]$ and $x' \in [r']$. Hence, the equality $\lambda(x) = \lambda(x')$ becomes $(r, t_r^{-1}(x)) = (r', t_{r'}^{-1}(x'))$. This implies that $t_r^{-1}(x) = t_{r'}^{-1}(x')$ and finally that $x = x'$ as t_r^{-1} is a permutation of E .

Now, we will prove that φ is a one-to-one homomorphism. Let g_1 and g_2 be two elements of G . It holds that

$$\begin{aligned} \varphi(g_1) \otimes \varphi(g_2) &= (\bar{g}_1, (t_{\bar{g}_1(r)}^{-1} g_1 t_r)_{r \in R}) \otimes (\bar{g}_2, (t_{\bar{g}_2(r)}^{-1} g_2 t_r)_{r \in R}) \\ &= (\bar{g}_1 \bar{g}_2, (t_{\bar{g}_1(\bar{g}_2(r))}^{-1} g_1 t_{\bar{g}_2(r)} t_{\bar{g}_2(r)}^{-1} g_2 t_r)_{r \in R}) \\ &= (\overline{g_1 g_2}, (t_{\overline{g_1 g_2}(r)}^{-1} g_1 g_2 t_r)_{r \in R}) = \varphi(g_1 g_2). \end{aligned}$$

Therefore, φ is a homomorphism. We still have to prove that φ is one-to-one. Assume that $\varphi(g_1) = \varphi(g_2)$. This hypothesis can be restated as follows

$$(\bar{g}_1, (t_{\bar{g}_1(r)}^{-1} g_1 t_r)_{r \in R}) = (\bar{g}_2, (t_{\bar{g}_2(r)}^{-1} g_2 t_r)_{r \in R}). \quad (4.5)$$

Let x be an element of E and let r be its representative in R . Then, $x_0 = t_r^{-1}(x)$ belongs to $[r_0]$. By assumption,

$$(t_{\bar{g}_1(r)}^{-1})^{-1} g_1 t_r(x_0) = (t_{\bar{g}_2(r)}^{-1})^{-1} g_2 t_r(x_0).$$

Simplifying, we obtain $(t_{\bar{g}_1(r)}^{-1})^{-1} g_1(x) = (t_{\bar{g}_2(r)}^{-1})^{-1} g_2(x)$. Equation 4.5 implies that $\bar{g}_1 = \bar{g}_2$. Thus, $\bar{g}_1(r) = \bar{g}_2(r)$ and $(t_{\bar{g}_1(r)}^{-1})^{-1} = (t_{\bar{g}_2(r)}^{-1})^{-1}$. Consequently, $g_1(x) = g_2(x)$. As this equality holds for all x in E , it follows that g_1 and g_2 are equal, proving that φ is one-to-one.

Let x be an element of E and let g be an element of G . It remains to prove that $\lambda(g(x)) = \varphi(g) \cdot \lambda(x)$. Let r be the representative of x in R . Then,

$$\begin{aligned} \varphi(g) \cdot \lambda(x) &= (\bar{g}, (t_{\bar{g}(r)}^{-1} g t_r)_{r \in R}) \cdot (r, t_r^{-1}(x)) = (\bar{g}(r), t_{\bar{g}(r)}^{-1} g t_r(t_r^{-1}(x))) \\ &= (\bar{g}(r), t_{\bar{g}(r)}^{-1} g(x)) = \lambda(g(x)). \end{aligned}$$

The result is proven. ■

Theorem 4.4 establishes that any imprimitive permutation group can be embedded into a wreath product. In the following, we use the same notation as in Theorem 4.4. Observe that A is a subgroup of $\text{Sym}([r_0])$ and that B is a subgroup of $\text{Sym}(R)$. Therefore, the group G can be identified with a subgroup of the wreath product $W = \text{Sym}([r_0]) \wr \text{Sym}(R)$. More formally, G is a subgroup of the (isomorphic) image of W under φ^{-1} .

Conversely, Proposition 4.11 ensures that the action of W on $R \times [r_0]$ is imprimitive and preserves the partition

$$\mathcal{Q} = \{\{r\} \times [r_0] \mid r \in R\}.$$

Hence $\varphi^{-1}(W)$ preserves the partition $\lambda^{-1}(\mathcal{Q}) = \mathcal{P}$. Combining these two results, we see that $\varphi^{-1}(W)$ is the biggest permutation group on E preserving the partition \mathcal{P} .

By assumption, \mathcal{P} is a G -invariant partition of E . Let p denote the cardinality of R and let q denote the cardinality of $[r_0]$. In other words, \mathcal{P} consists of p parts, each of cardinality q by virtue of Corollary 3.7. The discussion above proves that the number of permutations of E preserving \mathcal{P} is equal to the order of $\text{Sym}([r_0]) \wr \text{Sym}(R)$, given by

$$\#(\text{Sym}(R) \times \text{Sym}([r_0])^R) = (p!) \times (q!)^p. \quad (4.6)$$

4.1.3. Application of the Embedding Theorem

Recall that V is a d -dimensional subspace of \mathbb{F}_2^n and S is an n -bit S-box preserving $\mathcal{L}(V)$. Since U is a complement space of V , the partition $\mathcal{L}(V)$ is equal to $\{[u] \mid u \in U\}$. Thus, U is a system of distinct representatives of $\mathcal{L}(V)$.

Let G denote the subgroup of $\text{Sym}(\mathbb{F}_2^n)$ generated by S and the key additions $\{\alpha_x \mid x \in \mathbb{F}_2^n\}$. By assumption, S preserves $\mathcal{L}(V)$ and by Proposition 3.23, each key addition preserves $\mathcal{L}(V)$. Hence, the linear partition $\mathcal{L}(V)$ is G -invariant. Note that for each x and y in \mathbb{F}_2^n , the equality $\alpha_{x+y}(x) = y$ holds. Consequently, the group G is transitive and imprimitive because $\mathcal{L}(V)$ is assumed to be non-trivial.

Following the proof of Theorem 4.13, let t_u denote the key addition α_u for each u in U . It is easily seen that $t_u([0]) = t_u(V) = u + V = [u]$ so this definition meets the requirements of the proof. Moreover, we know that α_u is an involution, hence $(\alpha_u)^{-1} = \alpha_u$. Define the following mappings:

$$\begin{aligned} \lambda : \mathbb{F}_2^n &\longrightarrow U \times V & \varphi : G &\longrightarrow \text{Sym}(U) \times \text{Sym}(V)^U \\ x &\longmapsto (u_x, t_{u_x}(x)), & g &\longmapsto (\bar{g}, (t_{\bar{g}(u)}^{-1} g t_u)_{u \in U}), \end{aligned}$$

where u_x denotes the representative in U of $[x]$. Let $x = u + v$ be an element of \mathbb{F}_2^n and let g be an element of G . Then,

$$\begin{aligned}\lambda(x) &= (u, \alpha_u(x)) = (u, u + u + v) = (u, v), \\ \varphi(g) &= (\bar{g}, (t_{\bar{g}(u)}^{-1} g t_u)_{u \in U}) = (\bar{g}, (\alpha_{\bar{g}(u)} g \alpha_u)_{u \in U}).\end{aligned}$$

Since (λ, φ) is a permutation isomorphism from G on \mathbb{F}_2^n to $\text{Sym}(V) \wr \text{Sym}(U)$ on $U \times V$, the equality $\lambda(g(x)) = \varphi(g) \cdot \lambda(x)$ holds. Note that

$$\begin{aligned}\varphi(g) \cdot \lambda(x) &= (\bar{g}, (\alpha_{\bar{g}(u)} g \alpha_u)_{u \in U}) \cdot (u, v) = (\bar{g}(u), \alpha_{\bar{g}(u)} g \alpha_u(v)) \\ &= (\bar{g}(u), \bar{g}(u) + g(u + v)).\end{aligned}$$

Therefore,

$$\begin{aligned}g(u + v) &= \lambda^{-1}(\varphi(g) \cdot \lambda(x)) = \lambda^{-1}(\bar{g}(u), \bar{g}(u) + g(u + v)) \\ &= \bar{g}(u) + (\bar{g}(u) + g(u + v)).\end{aligned}$$

Let ρ denote the permutation \bar{S} of U and for each u in U , let τ_u denote the permutation of V defined by the rule $\tau_u(v) = \rho(u) + S(u + v)$. Then, apply the preceding equality with $g = S$ to obtain

$$S(u + v) = \rho(u) + \tau_u(v).$$

This proves that the decomposition of S given by Theorem 4.4 can be obtained using Krasner-Kaloujnine embedding theorem.

Even though it might seem harder to derive Theorem 4.4 from Theorem 4.13 rather than to prove it directly, this new perspective emphasizes the group structure associated with the decomposition of imprimitive S-boxes. For instance, suppose that S' is another n -bit S-box preserving \mathcal{P} . Denote by $\varphi(S') = (\rho', (\tau'_u)_{u \in U})$ the decomposition of S' . As φ is a homomorphism, $\varphi(SS') = \varphi(S) \otimes \varphi(S')$. Then, the decomposition of SS' is given for any $x = u + v$ in \mathbb{F}_2^n by

$$SS'(u + v) = \rho\rho'(u) + \tau_{\rho'(u)}\tau'_u(v).$$

In addition, the decomposition of S^{-1} is given by the formula

$$S^{-1}(u + v) = \rho^{-1}(u) + \tau_{\rho(u)}^{-1}.$$

Finally, Equation (4.6) asserts that the number of S-boxes preserving $\mathcal{L}(V)$ is given by

$$(2^{n-d}!) \times (2^d!)^{2^{n-d}}.$$

4.2. Differential and linear analyses

First, let us recall some basic facts about the differential and linear properties of S-boxes detailed in Chapter 1. Consider an n -bit S-box S and two elements a, b of \mathbb{F}_2^n .

The probability of the differential (a, b) and the correlation of the approximation (a, b) with respect to S are defined to be

$$\begin{aligned} \text{DP}_S(a, b) &= 2^{-n} \times \#\{x \in \mathbb{F}_2^n \mid S(x) + S(x + a) = b\}, \\ \text{C}_S(a, b) &= 2^{-(n-1)} \times \#\{x \in \mathbb{F}_2^n \mid \langle a, x \rangle = \langle b, S(x) \rangle\} - 1. \end{aligned}$$

The linear potential of the approximation (a, b) of S is then the square of its correlation, that is $\text{LP}_S(a, b) = \text{C}_S(a, b)^2$. The maximum differential probability DP_S^{\max} , the maximum absolute correlation C_S^{\max} and the maximum linear potential LP_S^{\max} of S are defined to be

$$\begin{aligned} \text{DP}_S^{\max} &= \max\{\text{DP}(a, b) \mid a \in (\mathbb{F}_2^n)^*, b \in \mathbb{F}_2^n\}, \\ \text{C}_S^{\max} &= \max\{\text{C}(a, b) \mid a \in \mathbb{F}_2^n, b \in (\mathbb{F}_2^n)^*\}, \\ \text{LP}_S^{\max} &= \max\{\text{LP}(a, b) \mid a \in \mathbb{F}_2^n, b \in (\mathbb{F}_2^n)^*\} = (\text{C}_S^{\max})^2. \end{aligned}$$

Moreover, in Chapter 1 Section 1.5.1, we have observed that $\text{DP}_S(a, b)$ is a multiple of $2^{-(n-1)}$ and that $\text{DP}_S^{\max} \geq 2^{-(n-1)}$. According to the Sidelnikov-Chabaud-Vaudenay bound (see Equations (1.13) and (1.14)), we have

$$\text{C}_S^{\max} \geq 2^{-\frac{n-1}{2}}, \quad \text{LP}_S^{\max} \geq 2^{-(n-1)}. \quad (4.7)$$

Now, assume that S preserves the partition $\mathcal{L}(V)$. In the previous section, we have proven that S can be constructed using permutations with smaller domains. More precisely, Theorem 4.4 establishes the existence of a permutation ρ of U and permutations $(\tau_u)_{u \in U}$ of V such that the relation

$$S(u + v) = \rho(u) + \tau_u(v)$$

holds for every $x = u + v$ in \mathbb{F}_2^n . This decomposition is fixed in the remainder of this section. In view of this result, it is natural to wonder if the differential probabilities and linear potentials of S are related to the ones of the permutations in its decomposition. The first problem is that these notions are defined for vectorial Boolean function whereas the domains of ρ and the τ_u are proper subspaces of \mathbb{F}_2^n . To solve this problem, we identify U with \mathbb{F}_2^{n-d} and V with \mathbb{F}_2^n using two isomorphisms, and then consider the permutations induced by ρ and the τ_u on these sets.

Notation 4.14. Let $\mathcal{B}_U = (u_i)_{i < n-d}$ and $\mathcal{B}_V = (v_i)_{i < n-d}$ be two bases of U and V respectively. Define the following mappings:

$$\begin{aligned} L_U : \mathbb{F}_2^{n-d} &\longrightarrow U & L_V : \mathbb{F}_2^d &\longrightarrow V \\ (x_{n-d-1}, \dots, x_0) &\longmapsto \sum_{i=0}^{n-d-1} x_i u_i, & (y_{d-1}, \dots, y_0) &\longmapsto \sum_{i=0}^{d-1} y_i v_i. \end{aligned}$$

It is easily seen that L_U and L_V are both isomorphisms of vector spaces. Define $\bar{\rho}$ to be the permutation $L_U^{-1} \circ \rho \circ L_U$ induced by ρ on \mathbb{F}_2^{n-d} . Similarly, for each u in U , denote by $\bar{\tau}_u$ the permutation $L_V^{-1} \circ \tau_u \circ L_V$ induced by τ_u on \mathbb{F}_2^d .

The following lemma explains how a permutation μ of an m -dimensional subspace W of \mathbb{F}_2^n is linked to the differential probabilities and correlations of its induced permutation $\bar{\mu}$ on \mathbb{F}_2^m .

Lemma 4.15. Let W be an m -dimensional subspace of \mathbb{F}_2^n and let L be an isomorphism from \mathbb{F}_2^m to W . Consider a permutation μ of W and denote by $\bar{\mu}$ its induced permutation $L^{-1} \circ \mu \circ L$ on \mathbb{F}_2^m . Let a and b be elements of W and denote $a' = L^{-1}(a)$, $b' = L^{-1}(b)$, $a^t = L^\top(a)$ and $b^t = L^\top(b)$. Then,

$$\begin{aligned} 2^m \times \text{DP}_{\bar{\mu}}(a', b') &= \#\{w \in W \mid \mu(w) + \mu(w + a) = b\}, \\ 2^{m-1} \times (C_{\bar{\mu}}(a^t, b^t) + 1) &= \#\{w \in W \mid \langle a, w \rangle = \langle b, \mu(w) \rangle\}. \end{aligned}$$

Proof. We begin with the correlation matrix of the permutation $\bar{\mu}$ induced by μ on \mathbb{F}_2^m . It is easily seen that

$$\begin{aligned} 2^{m-1} \times (C_{\bar{\mu}}(a^t, b^t) + 1) &= \#\{x \in \mathbb{F}_2^m \mid \langle a^t, x \rangle = \langle b^t, \bar{\mu}(x) \rangle\} \\ &= \#\{x \in \mathbb{F}_2^m \mid \langle L^\top(a), x \rangle = \langle L^\top(b), L^{-1}\mu L(x) \rangle\} \\ &= \#\{x \in \mathbb{F}_2^m \mid \langle a, L(x) \rangle = \langle b, \mu(L(x)) \rangle\}. \end{aligned}$$

Denote by E the set on the right side of the previous equation. As L is bijective, the set E and its image $L(E)$ have the same cardinality. Thus, replacing E by $L(E)$ yields

$$2^{m-1} \times (C_{\bar{\mu}}(a^t, b^t) + 1) = \#\{w \in W \mid \langle a, w \rangle = \langle b, \mu(w) \rangle\}.$$

Next, it remains to prove the statement about the differential probability of $\bar{\mu}$. By definition,

$$\begin{aligned} 2^m \times \text{DP}_{\bar{\mu}}(a', b') &= \#\{x \in \mathbb{F}_2^m \mid \bar{\mu}(x) + \bar{\mu}(x + a') = b'\} \\ &= \#\{x \in \mathbb{F}_2^m \mid L^{-1}\mu L(x) + L^{-1}\mu L(x + L^{-1}(a)) = L^{-1}(b)\}. \end{aligned}$$

Because L is bijective, $L(x) = L(y)$ if and only if $x = y$. Therefore,

$$\begin{aligned} 2^m \times \text{DP}_{\bar{\mu}}(a', b') &= \#\{x \in \mathbb{F}_2^m \mid L(L^{-1}\mu L(x) + L^{-1}\mu L(x + L^{-1}(a))) = LL^{-1}(b)\} \\ &= \#\{x \in \mathbb{F}_2^m \mid \mu(L(x)) + \mu(L(x) + a) = b\}. \end{aligned}$$

Again, considering the image of the last set under L , we obtain

$$2^m \times \text{DP}_{\bar{\mu}}(a', b') = \#\{w \in W \mid \mu(w) + \mu(w + a) = b\},$$

as was to be shown. ■

Example 4.16. Consider the bases $\mathcal{B}_U = (01, 02, 08)$ and $\mathcal{B}_V = (07, 1A)$ of the subspaces U and V . Next, we define the isomorphisms $L_U : \mathbb{F}_2^3 \rightarrow U$ and $L_V : \mathbb{F}_2^2 \rightarrow V$ following the construction given in Notation 4.14. For instance,

$$L_U(6) = L_U(110) = 1u_2 + 1u_1 + 0u_0 = 08 + 02 = 0A.$$

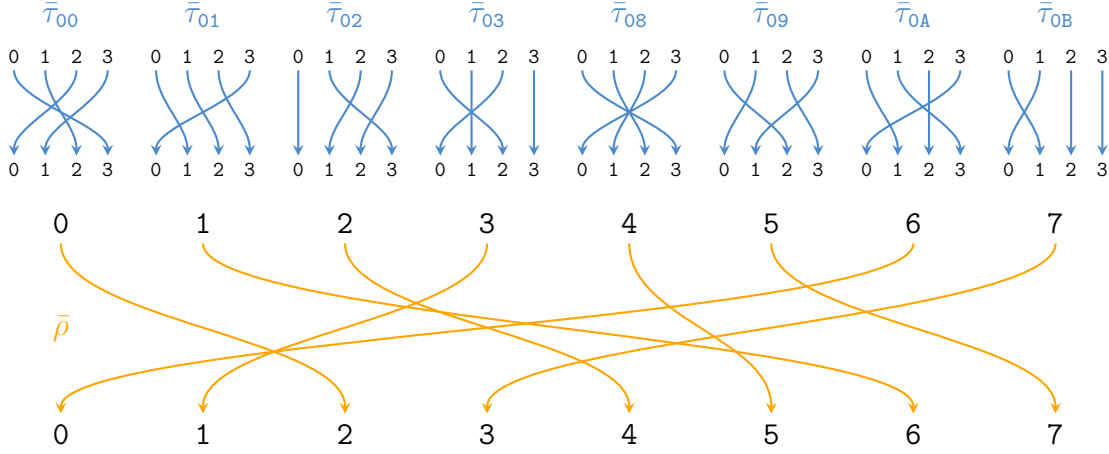
Explicitly, these isomorphisms are given in Figure 4.4. We can now compute the permutation $\bar{\rho}$ of \mathbb{F}_2^3 and the permutations $\bar{\tau}_u$ of \mathbb{F}_2^2 . For example,

$$\bar{\rho}(1) = (L_U^{-1} \circ \rho \circ L_U)(1) = L_U^{-1}(\rho(01)) = L_U^{-1}(0A) = 6.$$

The permutation $\bar{\rho}$ and the family $(\bar{\tau}_u)_{u \in U}$ are illustrated in Figure 4.5, which should be compared to the original decomposition of S represented in Figure 4.3. ▲

x	0	1	2	3	4	5	6	7
$L_U(x)$	00	01	02	03	08	09	0A	0B

x	0	1	2	3
$L_V(x)$	00	07	1A	1D

 Figure 4.4: The linear transformations L_U and L_V .

 Figure 4.5: The family of permutations $(\bar{\tau}_u)_{u \in U}$ and the permutation $\bar{\rho}$.

4.2.1. Correlation Matrices and Linear Potentials

Until now, we have divided our imprimitive S-box S into several smaller permutations and then transformed these permutations in order to reveal their differential and linear properties. We begin by investigating the correlation matrix of S . Our first result links some of its coefficients with the ones of the correlation matrix of $\bar{\rho}$. Even if the following theorem involves only few coefficients of C_S , it has a significant practical impact because these coefficients happen to be the greatest in general, and hence determine the resistance of S against linear cryptanalysis.

Theorem 4.17. Let a and b be two elements of V^\perp and denote by a^t and b^t their respective images under L_U^\top . Then,

$$C_S(a, b) = C_{\bar{\rho}}(a^t, b^t) \quad \text{and hence} \quad \text{LP}_S(a, b) = \text{LP}_{\bar{\rho}}(a^t, b^t).$$

Proof. Let $x = u + v$ be an element of \mathbb{F}_2^n . According to Theorem 4.4, the decomposition of $S(u + v)$ is $\rho(u) + \tau_u(v)$. Consequently,

$$\langle a, u + v \rangle = \langle b, S(u + v) \rangle \iff \langle a, u \rangle + \langle a, v \rangle = \langle b, \rho(u) \rangle + \langle b, \tau_u(v) \rangle$$

as the dot product is bilinear. Recall that a and b belong to V^\perp by assumption, so $\langle a, v \rangle$ and $\langle b, \tau_u(v) \rangle$ are both equal to 0. This discussion proves that

$$\{u + v \in \mathbb{F}_2^n \mid \langle a, u + v \rangle = \langle b, S(u + v) \rangle\} = \{u + v \in \mathbb{F}_2^n \mid \langle a, u \rangle = \langle b, \rho(u) \rangle\}.$$

Finally, combining Lemma 4.15 and the previous equation, we obtain

$$\begin{aligned} 2^{n-1}(C_S(a, b) + 1) &= \#\{u + v \in \mathbb{F}_2^n \mid \langle a, u \rangle = \langle b, \rho(u) \rangle\} \\ &= \#V \times \#\{u \in U \mid \langle a, u \rangle = \langle b, \rho(u) \rangle\} \\ &= 2^d \times 2^{n-d-1}(C_{\bar{\rho}}(a^t, b^t) + 1), \end{aligned}$$

which simplifies to give $C_S(a, b) = C_{\bar{\rho}}(a^t, b^t)$, as desired. \blacksquare

Remark 4.18. Consider the transpose L_U^\top of L_U seen as a mapping from \mathbb{F}_2^{n-d} to \mathbb{F}_2^n instead of U . Thus, L_U^\top is a mapping from \mathbb{F}_2^n to \mathbb{F}_2^{n-d} which cannot be injective since $d > 1$. According to Proposition 1.2, its kernel can be calculated as follows

$$\text{Ker}(L_U^\top) = (\text{Im } L_U)^\perp = U^\perp.$$

Then, observe that $U^\perp \cap V^\perp = (U + V)^\perp = (\mathbb{F}_2^n)^\perp = \{0\}$. Consequently, the restriction of L_U^\top to the orthogonal space of V in \mathbb{F}_2^n is injective and hence bijective because of the rank-nullity theorem. This discussion proves that the pairs (a^t, b^t) as defined in Theorem 4.17 are all distinct. Therefore, $C_{\bar{\rho}}$ is a submatrix of C_S .

Corollary 4.19. The maximum linear potential of S is lower bounded by $2^{-(n-d-1)}$.

Proof. As noted in Equation 4.7, there exist two elements a^t and b^t of \mathbb{F}_2^{n-d} both non-zero such that

$$|C_{\bar{\rho}}(a^t, b^t)| = C_{\bar{\rho}}^{\max} \geq 2^{-\frac{n-d-1}{2}}.$$

Let a and b denote the images of a^t and b^t under $(L_U^\top)^{-1}$. Then, Theorem 4.17 implies that

$$|C_S(a, b)| = |C_{\bar{\rho}}(a^t, b^t)| \geq 2^{-\frac{n-d-1}{2}}.$$

Multiplying both sides of this inequality by themselves yields $\text{LP}_S(a, b) \geq 2^{-(n-d-1)}$. Finally, observe that a and b are non-zero and the result is proven. \blacksquare

Remark 4.20. As explained in Section 1.5.1.c, the maximum absolute correlation of any 4-bit S-box is lower bounded by $2^{\frac{4+2}{2}} \times 2^{-4} = 2^{-1}$. Therefore, if $n - d = 4$ the previous reasoning yields the lower bound $\text{LP}_S^{\max} \geq 2^{-2}$, strengthening Corollary 4.19. Similarly, we know that every 2-bit S-box is affine, so has maximum absolute correlation equals to 1. Thus, the maximum linear potential of S is also equal to 1 when $n - d = 2$.

Example 4.21. First of all, we should explicit the transpose of L_U . Recall that we have defined L_U to be the linear mapping from \mathbb{F}_2^3 to \mathbb{F}_2^5 satisfying $L_U(1) = 01$, $L_U(2) = 02$ and $L_U(4) = 08$. Therefore, for any x in \mathbb{F}_2^3 , we have

$$L_U(x) = (x_2, x_1, x_0) \times \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = x \times A_U.$$

By definition, its transpose is the mapping L_U^\top from \mathbb{F}_2^5 to \mathbb{F}_2^3 defined by the rule

$$L_U^\top(x) = (x_4, \dots, x_0) \times A_U^\top = (x_3, x_1, x_0).$$

Recall that V was defined to be the subspace of \mathbb{F}_2^5 spanned by $\{07, 1A\}$. It is easily checked that the vectors in $\{05, 0B, 13\}$ are linearly independent and orthogonal to each element of the previous basis of V . Thus, the family $(05, 0B, 13)$ is a basis of V^\perp because this subspace is 3-dimensional. The restriction of L_U^\top to V^\perp is explicitly given by

$$\begin{array}{llll} L_U^\top(00) = 0, & L_U^\top(0B) = 7, & L_U^\top(13) = 3, & L_U^\top(18) = 4, \\ L_U^\top(05) = 1, & L_U^\top(0E) = 6, & L_U^\top(16) = 2, & L_U^\top(1D) = 5. \end{array}$$

As noted above, this restriction is a bijection. We will now reorder the rows and columns of the correlation matrix of S to highlight that $C_{\bar{\rho}}$ is one of its submatrices. Following Theorem 4.17, the firsts row and column indices should be

$$\underbrace{(L_U^\top)^{-1}(0)}_{00}, \quad \underbrace{(L_U^\top)^{-1}(1)}_{05}, \quad \dots, \quad \underbrace{(L_U^\top)^{-1}(6)}_{0E}, \quad \underbrace{(L_U^\top)^{-1}(7)}_{0B}.$$

The correlation matrix of S is illustrated in Figure 4.6 in its original then reordered forms. Next, Figure 4.7 shows that the top left 8×8 submatrix of the reordered form of C_S is exactly the correlation matrix of $\bar{\rho}$. It goes without saying that the coefficients affected by Theorem 4.17 stress the structure of such a correlation matrix. Moreover we see that they determine the absolute maximal correlation of S , as it is generally the case. Finally, it is worth noting that

$$\text{LP}_S^{\max} = \left(\frac{16}{32}\right)^2 = 2^{-2},$$

and thus S meets the bound of Corollary 4.19 with equality. ▀

4.2.2. Differential Probabilities

Along a similar line, we will investigate the differential probabilities of S and their links with the decomposition of S . But before stating our main results, we need two preliminary lemmas.

Lemma 4.22. Let $a = u_a + v_a$ and $b = u_b + v_b$ be two elements of \mathbb{F}_2^n . Denote by \mathcal{U} the set $\{u \in U \mid \rho(u) + \rho(u + u_a) = u_b\}$. Then,

$$2^n \times \text{DP}_S(a, b) = \sum_{u \in \mathcal{U}} \#\{v \in V \mid \tau_u(v) + \tau_{u+u_a}(v + v_a) = v_b\}.$$

Proof. Let x be any element of \mathbb{F}_2^n . Consider the following equation

$$S(x) + S(x + a) = b. \tag{4.8}$$

Write x as the sum $u + v$. According to Theorem 4.4, Equation (4.8) is equivalent to

$$\rho(u) + \tau_u(v) + \rho(u + u_a) + \tau_{u+u_a}(v + v_a) = u_b + v_b. \tag{4.9}$$

CHAPTER 4 – ANALYSIS OF A BACKDOOR S-Box

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
00	32
01	.	12	-12	8	-4	.	.	-4	-8	-4	4	.	4	.	4	8	-4	-12	.	4	.	.	4	.	-4	-12	-8	-4	.	.	-4	
02	.	-12	8	-12	-4	.	-4	-8	-4	.	4	.	.	-12	.	12	4	.	-4	.	.	-4	.	4	.	-4	-8	-4	4	.	4	8
03	.	8	-12	12	.	.	-4	-4	4	4	.	.	-4	-12	.	8	-4	4	8	.	-4	-4	4	4	8	.	4	12
04	.	-4	-4	.	-4	.	8	-12	.	4	4	.	-4	-8	.	-12	.	4	4	.	-4	8	.	4	.	-4	-4	.	12	.	-8	-12
05	16	-16	16	16	.	.
06	.	.	4	4	8	.	4	-4	-4	4	8	.	12	-4	.	.	-12	-12	.	-4	4	.	8	.	8	4	-4	.	.	12	-4	
07	.	-4	.	4	-12	.	-4	.	-4	8	12	.	-8	4	.	4	-12	-8	4	.	8	-4	.	-4	.	4	.	-4	-4	.	-12	.
08	.	8	4	-4	.	.	12	12	-12	4	.	.	-4	4	.	8	-4	4	8	.	-4	-4	.	.	.	-12	4	8	.	4	-4	
09	.	4	.	-4	-4	.	4	.	4	-8	4	.	8	-4	.	-4	-4	8	12	.	-8	-12	.	4	.	-4	.	-12	-12	.	-4	.
0A	.	-4	-4	.	-4	.	-8	-12	.	4	-12	.	-4	8	.	4	.	4	4	.	-4	-8	.	4	.	12	-4	.	-4	.	8	-12
0B	-16	.	.	16	16	16	.	.
0C	.	-4	-8	-4	4	.	-12	8	-12	.	-4	.	.	-4	.	4	-4	.	4	.	.	4	.	12	.	-12	8	4	-4	.	-4	-8
0D	.	.	4	4	8	.	4	-4	-4	-12	8	.	-4	-4	.	.	4	4	.	.	12	-12	.	8	.	8	4	12	.	.	-4	-4
0E	16	16	-16	.	16	
0F	.	-4	-12	-8	12	.	.	-4	8	-4	4	.	4	.	4	-8	-4	4	.	4	.	.	-12	.	-4	-12	8	-4	.	.	-4	
10	.	-8	4	12	.	.	-4	12	4	4	.	.	-4	-12	.	-8	-4	4	-8	.	-4	-4	-12	4	-8	.	4	-4
11	.	-4	.	4	4	.	12	.	12	8	-4	.	-8	4	.	4	4	-8	4	.	8	-4	.	12	.	-12	.	-4	-4	.	4	.
12	.	4	-4	-8	-12	.	.	4	8	4	-4	.	12	.	.	-4	-8	4	-4	.	12	.	.	12	.	4	-4	8	4	.	.	4
13	-16	-16	-16	16
14	.	4	8	4	-4	.	-4	-8	-4	.	4	.	.	4	.	-4	4	.	12	.	.	12	.	4	.	-4	-8	12	-12	.	4	8
15	.	.	4	4	-8	.	4	-4	-4	-12	-8	.	-4	-4	.	.	-12	4	.	.	12	4	.	-8	.	-8	4	-4	.	.	12	-4
16	-16	.	.	-16	.	-16	16	.	.
17	.	4	4	.	4	.	-8	-4	.	12	12	.	4	8	.	-4	.	12	-4	.	4	-8	.	-4	.	-12	4	.	4	.	8	-4
18	16	-16	-16	-16	.	.
19	.	4	4	.	4	.	8	-4	.	12	-4	.	4	-8	.	12	.	12	-4	.	4	8	.	-4	.	4	4	.	-12	.	-8	-4
1A	.	-4	-8	-4	-12	.	4	8	4	.	12	.	.	-4	.	4	12	.	4	.	.	4	.	-4	.	4	8	4	-4	.	12	-8
1B	.	-8	-12	-4	.	.	12	-4	-12	4	.	.	-4	4	.	-8	-4	4	-8	.	-4	-4	.	.	.	4	4	-8	.	4	12	
1C	.	-12	-4	8	4	.	.	4	-8	4	-4	.	12	.	.	-4	8	4	12	.	12	.	.	-4	.	4	-4	-8	4	.	.	4
1D	-16	.	.	-16	16	.	.	-16
1E	.	.	4	4	-8	.	4	-4	-4	4	-8	.	12	-4	.	.	4	-12	.	.	-4	-12	.	-8	.	-8	4	12	.	.	-4	-4
1F	.	-12	.	12	-4	.	4	.	4	-8	4	.	8	12	.	12	-4	8	-4	.	-8	4	.	4	.	-4	.	4	4	.	-4	.

V^\top

	00	05	16	13	18	1D	0E	0B	01	02	03	04	06	07	08	09	0A	0C	0D	0F	10	11	12	14	15	17	19	1A	1B	1C	1E	1F
00	32
05	.	16	16	.	.	16	-16
16	.	.	-16	-16	-16	16
13	.	-16	.	-16	16	.	-16
18	.	16	.	-16	.	-16	.	-16
1D	.	.	-16	16	.	.	-16	-16
0E	.	16	-16	.	16	.	.	16
0B	16	16	16	-16
01	12	-12	8	-4	.	-4	-8	-4	4	.	4	8	-4	-12	4	.	4	-4	-12	-8	-4	.	-4	
02	-12	8	-12	-4	-4	-8	-4	.	4	.	-12	12	4	.	-4	.	-4	4	-4	-8	-4	4	4	8
03	8	-12	12	.	-4	-4	4	4	.	-4	-12	8	-4	4	8	-4	-4	.	.	4	4	8	4	12
04	-4	-4	.	-4	8	-12	.	4	4	-4	-8	-12	.	4	4	-4	8	4	-4	-4	.	12	-8	-12
06	4	4	8	4	-4	-4	4	8	12	-4	.	-12	-12	.	-4	4	8	8	4	-4	.	12	-4
07	-4	.	4	-12	-4	.	-4	8	12	-8	4	4	-12	-8	4	8	-4	-4	4	.	-4	-4	-12	.
08	8	4	-4	.	12	12	-12	4	.	-4	4	8	-4	4	8	-4	-4	.	.	-12	4	8	4	-4
09	4	.	-4	-4	4	.	4	-8	4	8	-4	-4	-4	8	12	-8	-12	4	-4	.	-12	-12	-4	.
0A	-4	-4	.	-4	-8	-12	.	4	-12	-4	8	4	.	4	4	-4	-8	4	12	-4	.	-4	8	-12
0C	-4	-8	-4	4	-12	8	-12	.	-4	.	-4	4	-4	.	4	.	4	12	-12	8	4	-4	-4	-8
0D	4	4	8	4	-4	-4	-12	8	-4	-4	.	4	4	.	12	-12	8	8	4	12	.	-4	-4
0F	-4	-12	-8	12	.	-4	8	-4	4	4	.	4	-8	-4	4	4	.	-12	-4	-12	8	-4	.	-4
10	-8	4	12	.	-4	12	4	4	.	-4	-12	-8	-4	4	-8	-4	-4	.	.	-12	4	-8	4	-4
11	-4	.	4	4	12	.	12	8	-4	-8	4	4	4	-8	4	8	-4	12	-12	.	-4	-4	4	.
12	4	-4	-8	-12	.	4	8	4	-4	12	.	-4	-8	4	-4	12	.	12	4	-4	8	4	.	4
14	4	8	4	-4	-4	-8	-4	.	4	.	4	-4	4	.	12	.	12	4	-4	-8	12	-12	4	8
15	4	4	-8	4	-4	-4	-12	-8	-4	-4	.	-12	4	.	12	4	-8	-8	4	-4	.	12	-4
17	4	4	.	4	-8	-4	.	12	12	4	8	-4	.	12	-4	4	-8	-4	-12	4	.	4	8	-4
19	4	4	.	4	8	-4	.	12	-4	4	-8	12	.	12	-4	4	8	-4	4	4	.	-12	-8	-4
1A	-4	-8	-4	-12	4	8	4	.	12	.	-4	4	12	.	4	.	4	-4	4	8	4	-4	12	-8
1B	-8	-12	-4	.	12	-4	-12	4	.	-4	4	-8	-4	4</										

Figure 4.6: The reordered correlation matrix of S (multiplied by 2^5).

$2^5 \times C_S(a, b)$									$2^3 \times C_{\bar{\rho}}(a, b)$								
	00	05	16	13	18	1D	0E	0B		0	1	2	3	4	5	6	7
00	32	0	8
05	.	16	16	.	.	16	-16	.	1	.	4	4	.	.	4	-4	.
16	.	.	-16	-16	-16	16	.	.	2	.	.	-4	-4	-4	4	.	.
13	.	-16	.	-16	16	.	-16	.	3	.	-4	.	-4	4	.	-4	.
18	.	16	.	-16	.	-16	.	-16	4	.	4	.	-4	.	-4	.	-4
1D	.	.	-16	16	.	.	-16	-16	5	.	.	-4	4	.	.	-4	-4
0E	.	16	-16	.	16	.	.	16	6	.	4	-4	.	4	.	.	4
0B	16	16	16	-16	7	4	4	4	-4

Figure 4.7: The 8×8 top left submatrix of the reordered correlation matrix of S and the correlation matrix of $\bar{\rho}$.

Observe that $\rho(u) + \rho(u + u_a)$ lies in U and $\tau_u(v) + \tau_{u+u_a}(v + v_a)$ lies in V . Since \mathbb{F}_2^n is the direct sum of U and V , Equation (4.9) holds if and only if the following two equations hold:

$$\rho(u) + \rho(u + u_a) = u_b \quad \text{and} \quad (4.10)$$

$$\tau_u(v) + \tau_{u+u_a}(v + v_a) = v_b. \quad (4.11)$$

By definition, the statement “ $u \in \mathcal{U}$ ” is equivalent to Equation (4.10). Then, denoting by $P(u, v)$ the assertion “Equation (4.11) holds”, we have

$$\begin{aligned} 2^n \times \text{DP}_S(a, b) &= \#\{x \in \mathbb{F}_2^n \mid (4.8) \text{ holds}\} \\ &= \#\{(u, v) \in U \times V \mid u \in \mathcal{U} \text{ and } P(u, v)\} = \sum_{u \in \mathcal{U}} \#\{v \in V \mid P(u, v)\}. \end{aligned}$$

The result is proven. ■

Lemma 4.23. Let λ, μ be two permutations of V . For each v_a, v_b in V , denote by $D(v_a, v_b)$ the set $\{v \in V \mid \mu(v) + \lambda(v + v_a) = v_b\}$. Let v_a, v_b be elements of V . Then,

$$\sum_{\tilde{v}_b \in V} \#D(v_a, \tilde{v}_b) = \sum_{\tilde{v}_a \in V} \#D(\tilde{v}_a, v_b) = \#V.$$

Proof. Firstly, we contend that $\bigcup_{\tilde{v}_b \in V} D(v_a, \tilde{v}_b)$ is equal to V . Indeed, V is included in $\bigcup_{\tilde{v}_b \in V} D(v_a, \tilde{v}_b)$ since any element v belongs to $D(v_a, \mu(v) + \lambda(v + v_a))$ and the converse inclusion clearly holds. It goes without saying that the sets $D(v_a, \tilde{v}_b)$ are pairwise disjoint. Thus,

$$\#V = \# \bigcup_{\tilde{v}_b \in V} D(v_a, \tilde{v}_b) = \sum_{\tilde{v}_b \in V} \#D(v_a, \tilde{v}_b).$$

Next, we claim that $\bigcup_{\tilde{v}_a \in V} D(\tilde{v}_a, v_b)$ is equal to V . As previously, we only need to prove that V is included in $\bigcup_{\tilde{v}_a \in V} D(\tilde{v}_a, v_b)$. Let v in V . Since λ is onto, there exists an element x of V such that $\lambda(x) = \mu(v) + v_b$. Then, v lies in $D(x + v, v_b)$, proving our claim. Moreover, the sets $D(\tilde{v}_a, v_b)$ are pairwise disjoint because λ is one-to-one, implying that $\#V = \sum_{\tilde{v}_a \in V} \#D(\tilde{v}_a, v_b)$ as desired. ■

Now is the time to introduce our first theorem about the differential probabilities of S . Unlike Theorem 4.17, the next result involves all the coefficients of the matrix DP_S , thereby underlining its global structure.

Theorem 4.24. Let $a = u_a + v_a$ and $b = u_b + v_b$ be elements of \mathbb{F}_2^n and denote by u'_a and u'_b their images under L_U^{-1} . It holds that

$$\sum_{i \in [u_a]} \text{DP}_S(i, b) = \sum_{j \in [u_b]} \text{DP}_S(a, j) = \text{DP}_{\bar{\rho}}(u'_a, u'_b).$$

Especially, $\text{DP}_S(a, b) \leq \text{DP}_{\bar{\rho}}(u'_a, u'_b)$ and thus $\text{DP}_S^{\max} \leq \text{DP}_{\bar{\rho}}^{\max}$.

Proof. Denote by \mathcal{U} the set $\{u \in U \mid \rho(u) + \rho(u + u_a) = u_b\}$. According to Lemma 4.22, we have

$$\begin{aligned} 2^n \times \sum_{i \in [u_a]} \text{DP}_S(i, b) &= 2^n \times \sum_{\tilde{v}_a \in V} \text{DP}_S(u_a + \tilde{v}_a, b) \\ &= \sum_{\tilde{v}_a \in V} \left(\sum_{u \in \mathcal{U}} \#\{v \in V \mid \tau_u(v) + \tau_{u+u_a}(v + \tilde{v}_a) = v_b\} \right). \end{aligned}$$

Reversing the order of summation, we get

$$2^n \times \sum_{i \in [u_a]} \text{DP}_S(i, b) = \sum_{u \in \mathcal{U}} \left(\sum_{\tilde{v}_a \in V} \#\{v \in V \mid \tau_u(v) + \tau_{u+u_a}(v + \tilde{v}_a) = v_b\} \right).$$

In the same way, it can be proven that

$$2^n \times \sum_{j \in [u_b]} \text{DP}_S(a, j) = \sum_{u \in \mathcal{U}} \left(\sum_{\tilde{v}_b \in V} \#\{v \in V \mid \tau_u(v) + \tau_{u+u_a}(v + v_a) = \tilde{v}_b\} \right).$$

By virtue of Lemma 4.23, we have

$$2^n \times \sum_{i \in [u_a]} \text{DP}_S(i, b) = 2^n \times \sum_{j \in [u_b]} \text{DP}_S(a, j) = \sum_{u \in \mathcal{U}} \#V = \#\mathcal{U} \times 2^d.$$

Finally, Lemma 4.15 ensures that $\#\mathcal{U} = 2^{n-d} \times \text{DP}_{\bar{\rho}}(u'_a, u'_b)$. The result follows. \blacksquare

The next result is the analog of Theorem 4.17 for the differential probabilities. In a similar way, it considers few coefficients of DP_S but these coefficients are generally the greatest. Therefore, this result will be used to derive a lower bound on the resistance of S against differential cryptanalysis.

Theorem 4.25. Let v_a and v_b be two elements of V and denote by v'_a and v'_b their respective images under L_V^{-1} . Then

$$\text{DP}_S(v_a, v_b) = \frac{1}{2^{n-d}} \sum_{u \in U} \text{DP}_{\bar{\tau}_u}(v'_a, v'_b).$$

Particularly, the family $(\text{DP}_S(v_a, v_b))_{v_a, v_b \in V}$ is uniquely determined by $(\text{DP}_{\bar{\tau}_u})_{u \in U}$.

Proof. Applying Lemma 4.22 with $a = 0 + v_a$ and $b = 0 + v_b$ yields

$$2^n \times \text{DP}_S(v_a, v_b) = \sum_{u \in U} \#\{v \in V \mid \tau_u(v) + \tau_u(v + v_a) = v_b\},$$

since $\mathcal{U} = \{u \in U \mid \rho(u) + \rho(u + 0) = 0\} = U$. Then, Lemma 4.15 ensures that

$$\sum_{u \in U} \#\{v \in V \mid \tau_u(v) + \tau_u(v + v_a) = v_b\} = \sum_{u \in U} 2^d \times \text{DP}_{\bar{\tau}_u}(v'_a, v'_b).$$

Simplifying, we obtain the desired result. ■

Corollary 4.26. The maximum differential probability of S is lower bounded by the smallest multiple of $2^{-(n-1)}$ being directly greater than or equal to $\frac{1}{2^d-1}$.

Proof. Let v_a be a nonzero element of V . Applying Theorem 4.24 with $a = 0 + v_a$ and $b = 0 + 0$ yields

$$\sum_{j \in [0]} \text{DP}_S(v_a, j) = \text{DP}_{\bar{\rho}}(0, 0) = 1.$$

Since $[0] = V$, we have $\sum_{v \in V} \text{DP}_S(v_a, v) = 1$. Moreover, we know that $\text{DP}_S(v_a, 0) = 0$ because v_a is nonzero and S is a permutation. Thus, there are at most $2^d - 1$ elements v in V such that $\text{DP}_S(v_a, v)$ is nonzero. In order to minimize DP_S^{\max} , we would ideally require that $\text{DP}_S(v_a, v) = \frac{1}{2^d-1}$ for each nonzero element v of V . The result follows since any coefficient $\text{DP}_S(v_a, v)$ must be a multiple of $2^{-(n-1)}$. ■

Example 4.27. As was the case for Example 4.21, we will reorder the matrix DP_S to illustrate Theorems 4.24 and 4.25. Recall that the subspaces U and V can be expressed as

$$\begin{aligned} U &= \{L_U(x) \mid x \in \mathbb{F}_2^3\} = \{00, 01, 02, 03, 08, 09, 0A, 0B\}, \\ V &= \{L_V(x) \mid x \in \mathbb{F}_2^2\} = \{00, 07, 1A, 1D\}. \end{aligned}$$

Then, Theorem 4.24 suggests to consider the rows and columns of DP_S coset by coset. In other words, the row and column indices may be reordered as follows:

$$\begin{array}{cccc} L_U(0) + L_V(0), & L_U(0) + L_V(1), & L_U(0) + L_V(2), & L_U(0) + L_V(3), \\ \dots & \dots & \dots & \dots \\ L_U(7) + L_V(0), & L_U(7) + L_V(1), & L_U(7) + L_V(2), & L_U(7) + L_V(3). \end{array}$$

The natural and reordered forms of the matrix DP_S are represented in Figure 4.8. Thanks to this representation, it is now obvious that this matrix is highly structured. But to understand it, we give the differential probabilities matrices of $\bar{\rho}$ and $(\bar{\tau}_u)_{u \in U}$ in Figure 4.9. It is easily seen how $\bar{\rho}$ globally shapes the matrix DP_S . According to Theorem 4.24, if we fix a row and add all the coefficients whose column indices lie in the same coset, then we obtain a coefficient of $\text{DP}_{\bar{\rho}}$. For instance, consider the row $06 = 01 + 07$ and the coset $[02]$. Then

$$\begin{aligned} \sum_{j \in [02]} \text{DP}_S(06, j) &= \text{DP}_S(06, 02) + \text{DP}_S(06, 05) + \text{DP}_S(06, 18) + \text{DP}_S(06, 1F) \\ &= \frac{4}{32} + 0 + 0 + \frac{4}{32} \\ &= \frac{1}{4} = \text{DP}_{\bar{\rho}}(L_U^{-1}(01), L_U^{-1}(02)) = \text{DP}_{\bar{\rho}}(1, 2). \end{aligned}$$

CHAPTER 4 – ANALYSIS OF A BACKDOOR S-Box

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
00	32
01	.	.	.	4	.	4	.	.	2	4	2	.	.	2	4	.	2	.	.	4	4	.
02	2	2	.	.	.	4	2	2	.	.	2	2	2	2	8	4
03	.	.	.	2	2	4	4	.	.	.	4	4	4	4	2	2	.
04	.	.	.	2	2	4	4	.	.	.	4	4	4	4	2	2	.
05	2	2	4	8	.	.	2	2	4	.	2	2	2	2
06	.	.	4	.	4	.	.	.	2	4	2	.	.	2	4	.	2	.	.	.	4	4
07	12	8	.	.	12	.	.
08	.	2	2	.	.	2	2	.	2	.	.	4	4	.	.	2	.	.	2	.	.	2	.	.	2	.	.	2	2	.	.	2
09	.	.	.	4	2	.	.	4	.	2	.	.	.	2	2	.	2	.	.	4	.	4	4
0A	.	2	.	.	.	4	2	.	.	2	2	.	.	2	2	.	2	.	.	2	2	.	.	2	4	.	.	2	2	.	.	.
0B	4	.	.	4	.	4	4	.	.	.	4	.	8	.	.	4	.
0C	.	.	.	4	4	.	.	.	4	4	.	4	4	8	.	.	.
0D	.	2	4	.	.	.	2	.	.	2	2	.	.	2	2	.	2	.	.	2	2	.	2	2	2	.	.	4
0E	.	4	.	.	4	.	4	.	2	.	4	.	.	2	.	.	4	.	2	2	4
0F	.	2	2	.	.	2	2	.	2	.	4	4	.	.	2	.	.	2	.	.	2	.	2	.	2	.	.	2	2	.	.	2
10	.	2	.	.	.	4	2	.	2	2	.	.	2	2	.	2	.	.	2	2	.	2	.	2	4	.	.	2	2	.	.	.
11	8	4	4	4	.	4	.	4	4	.
12	.	2	2	.	.	2	2	.	2	2	.	4	2	.	.	2	.	2	4	.	2	.	.	2	2	.	2
13	.	4	.	4	.	.	4	.	.	2	.	4	.	.	2	.	.	4	.	2	2	4
14	4	.	.	.	2	.	.	4	.	2	2	2	.	2	.	4	4	4	.	4
15	.	2	2	.	.	2	2	.	2	2	.	4	2	.	.	2	4	.	2	.	.	.	2	2	.	.	2
16	.	8	.	4	4	.	.	.	4	.	4	.	.	4	.	.	.	4
17	.	2	4	.	.	.	2	.	.	2	2	.	.	2	2	.	2	.	.	2	2	.	2	.	2	.	.	.	2	2	.	4
18	2	2	4	.	8	.	2	2	4	.	2	2	2	2	2
19	.	.	4	2	2	4	4	.	4	.	.	4	2	2	4
1A	8	12	.	.	12	.	.	.
1B	.	.	4	4	2	4	2	.	.	2	.	4	2	4	4
1C	4	4	.	.	2	4	2	.	.	2	.	4	2	.	.	4	4
1D	12	12	.	.	.	8	.	.
1E	2	2	.	.	.	4	2	2	.	8	2	2	2	2	.	4	2	4

	[00]				[01]				[02]				[03]				[08]				[09]				[0A]				[0B]			
	00	07	1A	1D	01	06	1B	1C	02	05	18	1F	03	04	19	1E	08	0F	12	15	09	0E	13	14	0A	0D	10	17	0B	0C	11	16
[00]	00	32
	07	.	12	8	12
	1A	.	.	8	12	12
	1D	.	12	12	.	8
[01]	01
	06
	1B
	1C
[02]	02
	05
	18
	1F
[03]	03
	04
	19
	1E
[08]	08	2	2	2	2	2	2	2	2	2	2	2
	0F	2	2	2	2	2	2	2	2	2	2	2
	12	2	2	2	2	2	2	2	2	2	2	2
	15	2	2	2	2	2	2	2	2	2	2	2
[09]	09	4	4	4	.	4	2	2	2	2	
	0E	4	4	4	.	4	2	2	2	2	
	13	4	4	4	.	4	2	2	2	2	
	14	4	4	4	.	4	2	2	2	2	
[0A]	0A	2	2	2	2	.	4	4	2	2	2	2	2	2	2	2
	0D	2	2	2	2	4	.	.	4	2	2	2	2	2	2	2	2
	10	2	2	2	2	.	4	4	2	2	2	2	2	2	2	2
	17	2	2	2	2	4	.	.	4	2	2	2	2	2	2	2	2
[0B]	0B	8	4	4	.	4	.	4	4	4
	0C	8	4	4	.	.	4	4	4
	11	8	4	4	.	.	4	4	4
	16	8	4	4	.	.	4	4	4

Figure 4.8: The reordered differential probabilities matrix of S (multiplied by 2^5).

$2^3 \times \text{DP}_{\bar{\rho}}$								$2^2 \times \text{DP}_{\bar{\tau}_{00}}$				$2^2 \times \text{DP}_{\bar{\tau}_{01}}$				$2^2 \times \text{DP}_{\bar{\tau}_{02}}$				$2^2 \times \text{DP}_{\bar{\tau}_{03}}$			
0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	8	0	4	.	.	0	4	.	.	0	4	.	.	0	4	.	.
1	.	.	2	2	2	2	.	1	.	4	.	1	.	.	4	1	.	.	4	1	.	.	4
2	2	2	2	2	2	.	4	2	.	.	4	2	.	4	.	2	.	.	4
3	.	.	2	2	.	.	2	2	2	.	4	3	.	4	.	3	.	4	.	3	.	4	.
4	.	2	2	.	2	.	2	2	2	.	4	2	.	.	4	2	.	.	4	2	.	.	4
5	.	2	.	2	.	2	.	2	2	.	4	2	.	.	4	2	.	.	4	2	.	.	4
6	.	2	2	.	2	2	.	2	2	.	4	2	.	.	4	2	.	.	4	2	.	.	4
7	.	2	.	2	2	.	2	.	2	.	4	3	.	.	4	3	.	4	.	3	.	.	4

 Figure 4.9: The differential probabilities matrices of $\bar{\rho}$ and $(\bar{\tau}_u)_{u \in U}$.

Moreover, a similar result holds when we fix a column. Next, Theorem 4.25 ensures that the submatrix $(\text{DP}_S(v_a, v_b))_{v_a, v_b \in V}$ is the average of the matrices $\text{DP}_{\bar{\tau}_u}$. For instance,

$$\begin{aligned} \text{DP}_S(07, 1D) &= \frac{1}{8} \times \left(\sum_{u \in U} \text{DP}_{\bar{\tau}_u}(L_V^{-1}(07), L_V^{-1}(1D)) \right) = \frac{1}{8} \times \left(\sum_{u \in U} \text{DP}_{\bar{\tau}_u}(1, 3) \right) \\ &= \frac{1}{8} \times (0 + 1 + 1 + 1 + 0 + 0 + 0 + 0) = \frac{3}{8} = \frac{12}{32}. \end{aligned}$$

To conclude, let us take a look at the bound given in Corollary 4.26. The maximal differential probability of S is lower bounded by the smallest multiple of $2^{-(5-1)} = \frac{2}{32}$ greater than

$$\frac{1}{2^2 - 1} = \frac{10 + \frac{2}{3}}{32} \quad \text{hence} \quad \frac{12}{32}.$$

Since $\text{DP}_S^{\max} = \frac{12}{32}$ this bound is tight for $(n, d) = (5, 2)$. ▲

4.2.3. Designing a Backdoor S-Box

Relying on the results obtained so far, we will derive a construction for almost optimal partition-based backdoor S-boxes. For this purpose, let us summarize what we have learned from the three theorems of this section.

- According to Theorems 4.17 and 4.24 the maximum linear potential and differential probability of the permutation $\bar{\rho}$ should be as low as possible.
- In addition, Theorem 4.25 ensures that the sum of the matrices $\text{DP}_{\bar{\tau}_u}$ should have the smallest possible coefficients.

All the bounds given by Corollaries 4.19, 4.26 and Remark 4.20 for each $2 \leq n \leq 10$ and $1 \leq d < n$ are gathered in Figure 4.10.

Remark 4.28. By an (n, d) -PBB S-box, we mean an n -bit partition-based backdoor S-box mapping the linear partition associated with a d -dimensional subspace to another linear partition. As can be seen in Figure 4.10, if d is close to 0, any

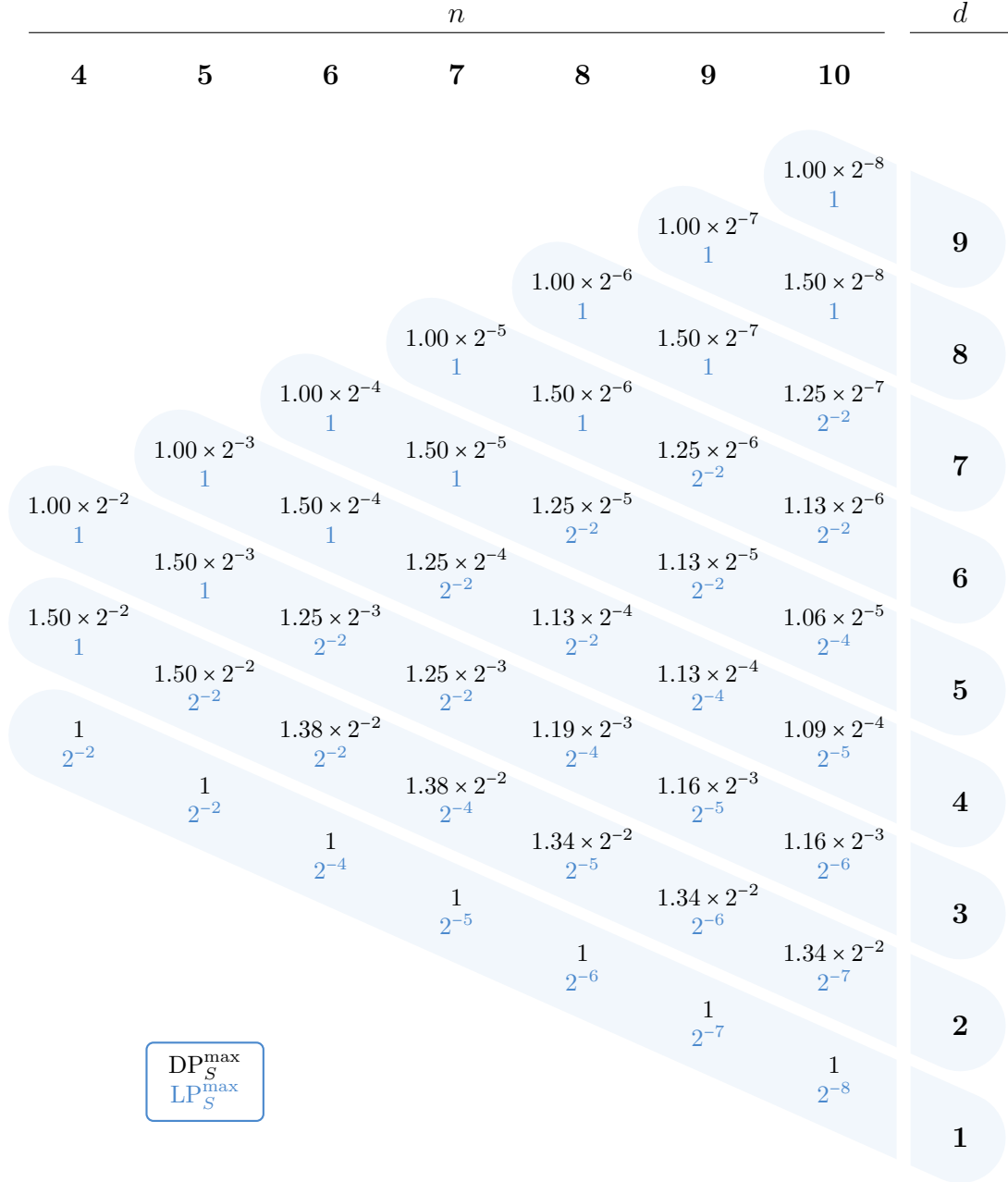


Figure 4.10: Lower bounds on maximum differential probability and linear potential of an S-box S mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$ where V and W are both d -dimensional subspaces of \mathbb{F}_2^n .

(n, d) -PBB S-box is weak against differential cryptanalysis. Inversely, if d is close to n , then any (n, d) -PBB S-box is weak against linear cryptanalysis. Therefore, an (n, d) -PBB S-box which resists differential and linear cryptanalysis must be such that $d \approx \frac{n}{2}$.

Let n be a positive integer. Choose two proper d -dimensional subspaces V and W of \mathbb{F}_2^n and an isomorphism L from V to W . We now detail how to design a “good” S-box mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. First, choose a complement subspace U of V in \mathbb{F}_2^n and define the isomorphisms L_U and L_V . Then, proceed as follows.

1. Construct a permutation $\bar{\rho}$ of \mathbb{F}_2^{n-d} which is (almost) optimal with respect to differential and linear cryptanalysis.
2. Construct a family of permutations $(\bar{\tau}_u)_{u \in U}$ of \mathbb{F}_2^d such that the sum SDP of their differential probability matrices satisfies

$$\frac{1}{2^{n-d}} \times \max_{a, b \in (\mathbb{F}_2^n)^*} \text{SDP}(a, b) \text{ is close to the bound of Corollary 4.26.}$$

3. Define the permutation S of \mathbb{F}_2^n by the formula

$$S(u + v) = (L_U \circ \bar{\rho} \circ L_U^{-1})(u) + (L_V \circ \bar{\tau}_u \circ L_V^{-1})(v).$$

4. If DP_S^{\max} and LP_S^{\max} are close to the bounds of Figure 4.10, then $L \circ S$ is a good S-box mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Otherwise, return to Step 1.

The reader may refer to Section 1.5.1.c which enumerates several families of permutations with optimal (or almost optimal) resistance against differential and linear cryptanalysis. Once we have such a permutation, other permutations which have the same differential and linear properties can be obtained using the affine-equivalence (see Equation 4.1), the EA-equivalence or the CCZ-equivalence [34] (see [20]). When d is greater than 4, we suggest to search the family of permutations $(\bar{\tau}_u)_{u \in U}$ among the permutations with good differential properties. In practice, we obtain partition-based backdoor S-boxes close to the bounds of Figure 4.10 after a small number of iterations.

4.3. A Toy Partition-Based Backdoor Cipher

Before concluding in the next section, we introduce a toy partition-based backdoor cipher called TBC (standing for *Toy Backdoor Cipher*) to illustrate this and the previous chapters.

4.3.1. Specification of TBC

TBC is a substitution-permutation network processing a 36-bit block of plaintext using a 36-bit cipher key. The round function of TBC is quite simple and is largely inspired from the lightweight cipher PRESENT [17] and its small-scale variants SMALL-PRESENT [69]. Just as in all substitution permutation networks, the round function

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	37	34	17	3E	1A	0E	10	04	15	1C	3D	3C	12	26	30	24
1.	13	2A	31	02	0F	2B	2D	0B	11	00	1B	20	05	21	07	03
2.	3F	1E	35	16	32	06	18	2E	1F	14	1D	36	38	2C	3A	0C
3.	3B	08	33	0A	0D	09	25	01	19	28	39	22	2F	23	27	29

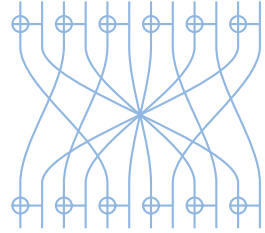
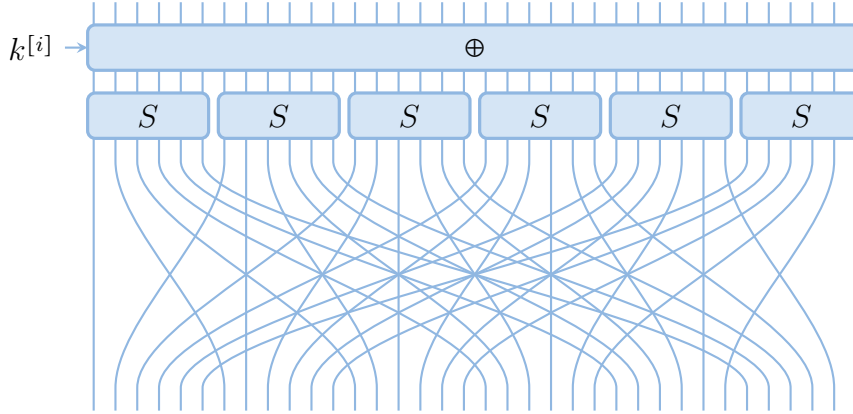


 Figure 4.11: The 6-bit S-box S (left) and the 12-bit diffusion D (right) of TBC.

 Figure 4.12: The round function $F_{k^{[i]}}$ of TBC.

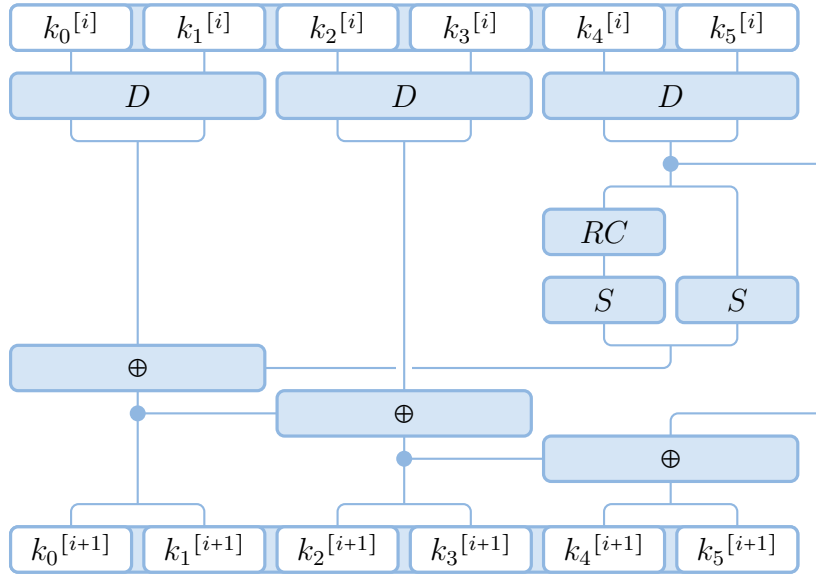
consists of a round-key addition, a substitution layer and a diffusion layer. The substitution layer uses one 6-bit S-box which is applied six times in parallel across the 36 bits of the block. This S-box is denoted by S and is defined in Figure 4.11. Then the diffusion layer is a bit permutation (see Definition 1.12) given by the formula

$$\phi(i) = 6 \times (i \bmod 6) + \left\lfloor \frac{i}{6} \right\rfloor.$$

A diagrammatic representation of the round function is provided in Figure 4.12. The encryption process consists of 21 iterative applications of this round function, then ends with a key addition. Therefore, the last round is equal to the other rounds and the encryption requires 22 round keys.

The TBC key schedule is inspired by the AES-128 key schedule. The cipher key K is also the first round key $k^{[0]}$ and each round key $k^{[i]}$ is computed by applying a function G_i to the preceding round key $k^{[i-1]}$. This function uses a 12-bit linear permutation D illustrated in Figure 4.11. Explicitly, the linear diffusion D is defined for each x in \mathbb{F}_2^{12} by the rule

$$D(x_0, \dots, x_{11}) = (x_2 + x_3 + x_{10} + x_{11}, x_{10} + x_{11}, x_4 + x_5 + x_8 + x_9, x_8 + x_9, x_0 + x_1 + x_6 + x_7, x_6 + x_7, x_5 + x_{11}, x_{11}, x_3 + x_7, x_7, x_1 + x_9, x_9).$$


 Figure 4.13: The round function G_i of TBC key schedule.

For instance, D maps $(30 \parallel 07)$ to $(0C \parallel 31)$. Now, let us explain how the round function G_i of the key schedule derives the round key $k^{[i+1]}$ from $k^{[i]}$. First, the diffusion D is applied three times in parallel to $k^{[i]}$ to obtain a 36-bit block denoted by x , that is to say

$$D(k_0^{[i]} \parallel k_1^{[i]}) = (x_0 \parallel x_1), \quad D(k_2^{[i]} \parallel k_3^{[i]}) = (x_2 \parallel x_3), \quad D(k_4^{[i]} \parallel k_5^{[i]}) = (x_4 \parallel x_5).$$

Then, compute $(y_4 \parallel y_5) = (S(x_4 + r_i) \parallel S(x_5))$. Here $x_4 + r_i$ denotes the addition of the round constant r_i which is performed in \mathbb{F}_2^6 , so is just a bitwise exclusive or (Xor) between x_4 and r_i . The round constant r_i is equal to the integer $i+1$ expressed in binary. For instance $r_0 = 01$ and $r_9 = 0A$. Finally, the next round key $k^{[i+1]}$ is computed as follows:

$$\begin{aligned} (k_0^{[i+1]} \parallel k_1^{[i+1]}) &= (x_0 \parallel x_1) + (y_4 \parallel y_5), \\ (k_2^{[i+1]} \parallel k_3^{[i+1]}) &= (x_2 \parallel x_3) + (k_0^{[i+1]} \parallel k_1^{[i+1]}), \\ (k_4^{[i+1]} \parallel k_5^{[i+1]}) &= (x_4 \parallel x_5) + (k_2^{[i+1]} \parallel k_3^{[i+1]}). \end{aligned}$$

An illustration of the round function of TBC key schedule is given in Figure 4.13.

4.3.2. Differential and Linear Cryptanalysis

It is easily checked with a computer that the S-box S has maximum linear potential equal to 2^{-2} and maximum differential probability equal to $\frac{14}{64} = 1.75 \times 2^{-3}$. Since any linear or differential trail has at least one active S-box per round, we can upper bound the potential of an optimal 20-round linear trail by $(2^{-2})^{20} = 2^{-40}$ and the probability of an optimal 20-round differential trail by $(1.75 \times 2^{-3})^{20} \approx 1.11 \times 2^{-46}$.

Using the algorithm `OptTrail` presented in Chapter 2, we have proven that the potential of an optimal 19-round linear trail is really equal to 2^{-40} but the probability

of an optimal 19-round differential trial is equal to 1.76×2^{-49} . Therefore, a linear cryptanalysis would require $c \times 2^{40}$ known plaintext/ciphertext pairs and a differential cryptanalysis $c \times 2^{49}$ chosen plaintext/ciphertext pairs with $c \geq 5$. Since there are only 2^{36} different plaintext blocks, this cipher is practically secure against differential and linear cryptanalysis.

Observe that we have used here the *heuristic measure* [57], so we have never considered the probability of an optimal 19-round differential. As will be seen in Section 4.3.4, TBC is actually weak to differential cryptanalysis.

4.3.3. The Backdoor

As claimed in introduction, TBC is a partition-based backdoor cipher. Thus, the encryption function maps a partition of the plaintext space to a partition of the ciphertext space, no matter the cipher key used. More precisely, this property still holds with independent round keys, and hence the theoretical framework of Chapter 3 applies.

Since the diffusion layer of TBC is strongly proper over 1 round, we know that the S-box S maps a non-trivial linear partition $\mathcal{L}(V)$ to another partition $\mathcal{L}(W)$. Actually, S preserves the partition $\mathcal{L}(V)$ where V is the subspace of \mathbb{F}_2^6 defined to be

$$V = \text{span}(20, 08, 02) = \{(x_5, 0, x_3, 0, x_1, 0) \mid x_5, x_3, x_1 \in \mathbb{F}_2\}.$$

Probably the simplest complement subspace U of V in \mathbb{F}_2^6 is

$$U = \text{span}(10, 04, 01) = \{(0, x_4, 0, x_2, 0, x_0) \mid x_4, x_2, x_0 \in \mathbb{F}_2\}.$$

Therefore, $\mathcal{L}(V)$ is equal to $\{u + V \mid u \in U\}$. Knowing that S preserves $\mathcal{L}(V)$, it is easily seen that the substitution layer preserves any linear partition of the form

$$\mathcal{L}\left(\prod_{i=0}^5 E_i\right) \quad \text{where for each } i, E_i = \{0_6\} \text{ or } E_i = V \text{ or } E_i = \mathbb{F}_2^6.$$

Denote by π the diffusion layer of TBC. If π maps the subspace $E = \prod_{i=0}^5 E_i$ to $E' = \prod_{i=0}^5 E'_i$, then one (and only one) of the following cases holds

- $E = E' = (V \times \{0_6\})^3$,
- $E = V^6$ and $E' = (\mathbb{F}_2^6 \times \{0_6\})^3$,
- $E = (\mathbb{F}_2^6 \times \{0_6\})^3$ and $E' = V^6$,
- $E = E' = (\mathbb{F}_2^6 \times V)^3$.

All these cases are illustrated in Figure 4.14. Next, we can easily derive the next theorem from Proposition 3.23 and 3.25.

Theorem 4.29 (TBC Round Function). The round function F of TBC preserves at the same time the linear partitions $\mathcal{L}((V \times \{0_6\})^3)$ and $\mathcal{L}((\mathbb{F}_2^6 \times V)^3)$. Moreover, it maps $\mathcal{L}((\mathbb{F}_2^6 \times \{0_6\})^3)$ to $\mathcal{L}(V^6)$ and vice versa. Since the encryption function is a composition of 21 round functions with a final round key addition, the same result holds for the whole encryption process.

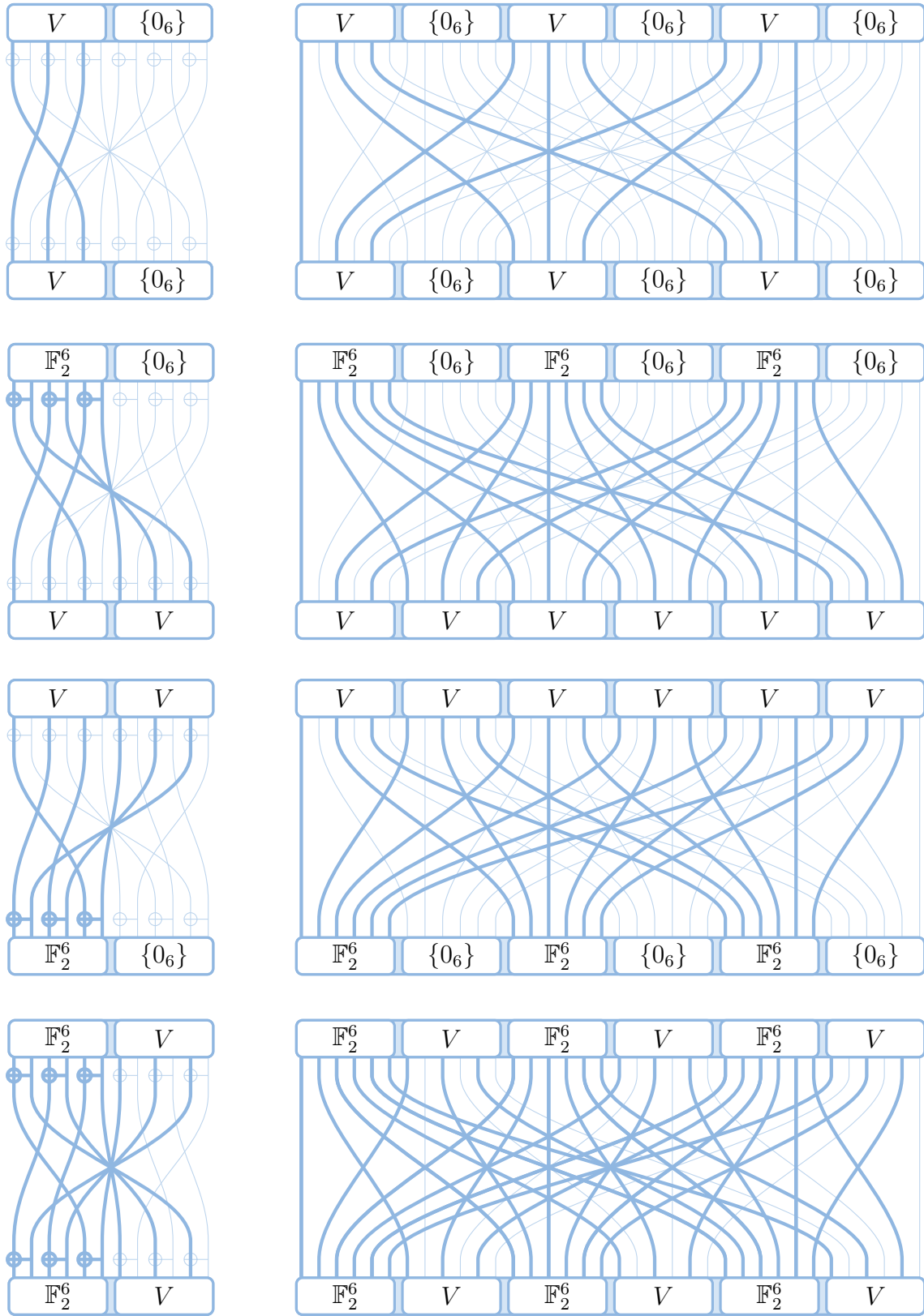


Figure 4.14: Spaces used by the linear mappings of TBC.

4.3.3.a. Basic and Multiple Partitions Attacks

In Chapter 3 Section 3.1.3, we have presented several ways to exploit such a backdoor following Paterson's work [88]. Denoting by G the group generated by the round functions, Theorem 4.29 ensures that $\mathcal{L}((V \times \{0_6\})^3)$ and $\mathcal{L}((\mathbb{F}_2^6 \times V)^3)$ are two G -invariant partitions. Consider for instance the partition $\mathcal{B} = \mathcal{L}((V \times \{0_6\})^3)$. First, we need an efficient description of \mathcal{B} , namely we require that

- it must be easy to give one representative message for each part of \mathcal{B} ,
- given a message x , it must be easy to enumerate all the messages lying in the same part of \mathcal{B} as x .

Because U is a complement subspace of V , the subspace $(U \times \mathbb{F}_2^6)^3$ is a complement of $(V \times \{0_6\})^3$ in $(\mathbb{F}_2^6)^6$. Hence, we have

$$\mathcal{L}((V \times \{0_6\})^3) = \{u + (V \times \{0_6\})^3 \mid u \in (U \times \mathbb{F}_2^6)^3\}.$$

Thus, each message in $(\mathbb{F}_2^6)^6$ can be written as

$$((v_{00}, u_{01}, v_{02}, u_{03}, v_{04}, u_{05}), (u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}), (v_{20}, u_{21}, v_{22}, u_{23}, v_{24}, u_{25}), \\ (u_{30}, u_{31}, u_{32}, u_{33}, u_{34}, u_{35}), (v_{40}, u_{41}, v_{42}, u_{43}, v_{44}, u_{45}), (u_{50}, u_{51}, u_{52}, u_{53}, u_{54}, u_{55})).$$

The bits u_i give the coset representative of the message and bits v_i represent its index within this coset.

We can now present a basic cryptanalysis. Let K be an known cipher key and denote by E_K the encryption function of TBC associated with K . For each u in $(U \times \mathbb{F}_2^6)^3$, require its encryption $c = E_K(u)$ and denote by c_u its coset representative in $(U \times \mathbb{F}_2^6)^3$. Thus, we have to store 2^{27} pairs (u, c_u) . Next, assume that we are given an unknown ciphertext c . We can then compute its coset representative c_u and obtain a representative of the corresponding plaintext. In other words, the plaintext lies in the set $c_u + (V \times \{0_6\})^3$.

To summarize, this cryptanalysis requires 2^{27} chosen plaintexts. Then, we recover 2^9 plaintext candidates for each unknown ciphertext, compromising seriously the security of TBC. Inversely, the same cryptanalysis based on the partition $\mathcal{L}((\mathbb{F}_2^6 \times V)^3)$ requires only 2^9 chosen plaintexts but yields 27 bits of uncertainty on each plaintext.

Finally, it is a simple matter to verify that this cryptanalysis can be generalized to any partition-based backdoor cipher. Knowing that the encryption function maps $\mathcal{L}((\mathbb{F}_2^6 \times \{0_6\})^3)$ to $\mathcal{L}(V^6)$, we can attack the cipher with 2^{18} chosen plaintexts. Then we can recover 18 bits of the plaintext corresponding to an unknown ciphertext. Since the encryption process also maps $\mathcal{L}(V^6)$ to $\mathcal{L}((\mathbb{F}_2^6 \times \{0_6\})^3)$, we deduce another cryptanalysis with the same parameters. Nonetheless, these two attacks can be combined as described in Section 3.1.3. The resulting cryptanalysis needs 2×2^{18} chosen plaintexts but can then recover 27 bits of any plaintext.

4.3.3.b. Key Schedule Dependent Attack

Even if this last cryptanalysis gives a clear advantage to any attacker aware of the backdoor, we can do much better using a key schedule dependent attack. As can be seen in Figure 4.14, the diffusion D preserves the partitions $\mathcal{L}(V \times \{0_6\})$ and $\mathcal{L}(\mathbb{F}_2^6 \times V)$, maps $\mathcal{L}(\mathbb{F}_2^6 \times \{0_6\})$ to $\mathcal{L}(V^2)$ and maps $\mathcal{L}(V^2)$ to $\mathcal{L}(\mathbb{F}_2^6 \times \{0_6\})$. Referring now to Figure 4.13, it is straightforward to check that Theorem 4.29 still holds if we consider the round function of the key schedule.

Corollary 4.30 (TBC Key Schedule). Each round G_i of TBC key schedule preserves $\mathcal{L}((V \times \{0_6\})^3)$ and $\mathcal{L}((\mathbb{F}_2^6 \times V)^3)$. In addition, it maps $\mathcal{L}((\mathbb{F}_2^6 \times \{0_6\})^3)$ to $\mathcal{L}(V^6)$ and inversely.

Assume that A and B are two subspaces of \mathbb{F}_2^{36} such that for each round key k the round function F_k maps $\mathcal{L}(A)$ to $\mathcal{L}(B)$. Consider two round keys k and k' lying in the same coset of A . Since the linear partition $\mathcal{L}(A)$ is equal to the quotient space \mathbb{F}_2^{36} / A and since the cosets $k + A$ and $k' + A$ are equal, the key additions α_k and $\alpha_{k'}$ induce the same permutation of $\mathcal{L}(A)$. As a consequence, for each message x in \mathbb{F}_2^{36} , it holds that

$$F_k(x + A) = F_{k'}(x + A) \quad \text{or equivalently} \quad F_k(x) + B = F_{k'}(x) + B.$$

Combining this observation with Corollary 4.30, we deduce the following theorem.

Theorem 4.31 (Key Schedule Dependent Attack). Denote by $(\mathcal{L}(A), \mathcal{L}(B))$ an input/output partition pair given by Theorem 4.29. If two cipher keys K and K' lie in the same coset of A , then for each message x in \mathbb{F}_2^{36} , we have

$$E_K(x) + B = E_{K'}(x) + B.$$

Let us now detail an efficient key schedule dependent cryptanalysis of TBC. Let K be the unknown cipher key. Then obtain a few plaintext/ciphertext pairs (p_i, c_i) . According to our experiments, only two or three pairs are sufficient. For simplicity of explanation, denote the cipher key K by

$$((v_{00}, u_{01}, v_{02}, u_{03}, v_{04}, u_{05}), (v_{10}, u_{11}, v_{12}, u_{13}, v_{14}, u_{15}), (v_{20}, u_{21}, v_{22}, u_{23}, v_{24}, u_{25}), \\ (v_{30}, u_{31}, v_{32}, u_{33}, v_{34}, u_{35}), (v_{40}, u_{41}, v_{42}, u_{43}, v_{44}, u_{45}), (v_{50}, u_{51}, v_{52}, u_{53}, v_{54}, u_{55})).$$

The first step considers the subspaces $A = B = (\mathbb{F}_2^6 \times V)^3$. According to Theorem 4.31, if a cipher key K' lies in the same coset of A as K , then for each index i , $E_{K'}(p_i)$ and c_i lie in the same coset of B . This property can be used in the following key recovery attack.

1. For each representative K' in $(\{0_6\} \times U)^3$, test whether $E_{K'}(p_i)$ and c_i lie in the same coset of $(\mathbb{F}_2^6 \times V)^3$ for each index i .

Among the 2^9 representatives tested, it should remain only one or two candidates. Thus, we can assume that we now know the bits u_{ij} for all i, j in $\{1, 3, 5\}$. For the second step, consider the subspaces $A = V^6$ and $B = (\mathbb{F}_2^6 \times \{0_6\})^3$.

2. For each candidate representative found in step 1,

Denote by $(u_{ij})_{i,j \in \{1,3,5\}}$ the bits found previously.

For each representative K' in U^6 such that $u'_{ij} = u_{ij}$ for all i, j in $\{1, 3, 5\}$, test whether $E_{K'}(p_i)$ and c_i lie in the same coset of $(\mathbb{F}_2^6 \times \{0_6\})^3$ for each index i .

After this step, we can assume that we now the bits u_{ij} for all $0 \leq i < 6$ and j in $\{1, 3, 5\}$. For the third step, consider the subspaces $A = B = (V \times \{0_6\})^3$.

3. For each candidate representative found in step 2,

Denote by $(u_{ij})_{0 \leq i < 6, j \in \{1,3,5\}}$ the bits found previously.

For each representative K' in $(U \times \mathbb{F}_2^6)^3$ such that $u'_{ij} = u_{ij}$ for all i, j , test whether $E_{K'}(p_i)$ and c_i lie in the same coset of $(V \times \{0_6\})^3$ for each index i .

Now, we know all the cipher key bits except $(v_{ij})_{i,j \in \{0,2,4\}}$. These nine remaining bits are found with an exhaustive search.

4. For each candidate representative found in step 3,

Denote by $(u_{ij})_{ij}$ and $(v_{ij})_{i,j \in \{0,2,4\}}$ the bits found previously.

For each cipher K' such that $u'_{ij} = u_{ij}$ and $v'_{ij} = v_{ij}$, test whether $E_{K'}(p_i) = c_i$ holds for each index i .

A candidate cipher key in step 4 is almost always equal to the true cipher key K .

Using two pairs (p_i, c_i) , each step requires at most 2×2^9 encryptions. Assuming that there is only one candidate after each step, this cryptanalysis computes $4 \times 2^{10} = 2^{12}$ encryptions to recover the cipher key. We found experimentally that on average, this attack performs almost 2^{10} encryptions.

4.3.4. The Flaws of This Cipher

The main flaw of TBC is that the S-boxes are incomplete, namely there are some output bits independent of some input bits. Unfortunately, the whole encryption function inherits this bad property. For instance, for any message x in $(\mathbb{F}_2^6)^6$ we have

$$E_K(x + (\mathbb{F}_2^6 \times \{0_6\})^3) = E_K(\{(y_0, x_1, y_2, x_3, y_4, x_5) \mid y_0, y_2, y_4 \in \mathbb{F}_2^6\}) = E_K(x) + V^6.$$

As a consequence, every output bits which has odd index is independent of the bundles 1, 3, and 5. This proves that TBC cannot seem to be secure, even when we are not aware of the backdoor.

The second flaw of TBC relies on an attack introduced by Knudsen in [60], called *truncated differential cryptanalysis*. An n -bit truncated difference pattern a is an element of $\{0, 1, \star\}^n$. By the set of the difference patterns associated with a , we mean

$$\{a\} = \{(x_0, \dots, x_{n-1}) \in \mathbb{F}_2^n \mid \forall i < n, (a_i \in \mathbb{F}_2 \Rightarrow x_i = a_i)\}.$$

For instance, the set associated with the truncated pattern $(1\star 01\star)$ is

$$\{1\star 01\star\} = \{10010, 10011, 11010, 11011\}.$$

Alternatively, a truncated difference pattern can be seen as a collection of difference patterns. Then an r -round truncated differential is a pair (a, b) of truncated patterns which predicts that if two plaintexts have a difference lying in $\{a\}$, then their corresponding ciphertexts have a difference lying in $\{b\}$ with some probability. As explained in [64, pp. 156], the term truncated draws attention to the fact that only some bits of the output difference are predicted.

Caranti et al. established a link between imprimitive ciphers and truncated differential cryptanalysis in [31, Corollary 4.1]. Their result can be easily generalized to partition-based backdoor ciphers with independent round keys.

Proposition 4.32. Let E be a partition-based backdoor cipher mapping $\mathcal{L}(V)$ to $\mathcal{L}(W)$. If the difference of two plaintexts lies in V , then the difference of their ciphertexts lies in W .

Proof. Consider two plaintexts p and p' such that $p + p'$ is in V . Then, there exists v in V such that $p = p' + v$ and hence p and p' lie in the same coset of V . Since E_K maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, $E_K(p)$ belongs to the same coset of W as $E_K(p')$. The result follows. ■

Now, observe that the subspace V used in the backdoor of TBC can be written as $V = \{\star 0 \star 0 \star 0\}$. Combining this observation with Proposition 4.32, it is easily seen that each input/output partitions pairs of Theorem 4.29 yields a truncated differential with probability 1. However, we can argue that finding such a truncated differential is equivalent to recover the backdoor.

This link with truncated differential cryptanalysis also affects the resistance of TBC with respect to classical differential cryptanalysis. Indeed, we have just seen that the truncated differential (a, b) where

$$a = b = (0 \star 0 \star 0 \star, 000000, 0 \star 0 \star 0 \star, 000000, 0 \star 0 \star 0 \star, 000000)$$

holds with probability ones over any number of rounds since the round function preserves the partition $\mathcal{L}((V \times 0_6)^3)$. Consequently, the (classical) differential (a, b) with

$$a = b = (20, 00, 00, 00, 00, 00)$$

also holds with high probability. Our experiments showed that this probability is close to 2^{-9} . Therefore, TBC is vulnerable to differential cryptanalysis, even if it seemed secure using the heuristic measure.

4.4. Preventing Partition-Based Backdoors

To conclude this theoretical treatment of partition-based backdoor ciphers, we will now present two criteria to prove that an SPN does not have such a backdoor. In the previous chapter, we have considered a generic SPN which maps a partition of the plaintext to a partition of the ciphertexts independently of the round keys used. We have then proven that when its diffusion layer is strongly proper, at least

one of its S-boxes must map a linear partition to another one. In this chapter, the differential and linear properties of such S-boxes have been studied. We then derives lower bounds on their resistance to these attacks in Corollaries 4.19, 4.26 and Remark 4.20. Therefore, if all the S-boxes of an SPN have a better resistance than what is possible to achieve using backdoor S-boxes, then the cipher does not have a partition-based backdoor. This proves the following theorem

Theorem 4.33. Consider an nm -bit substitution permutation network with m S-boxes over \mathbb{F}_2^n . Assume that its diffusion layer is strongly proper over r rounds. If each S-box S_i is such that for each $1 \leq d \leq n-1$, the values $LP_{S_i}^{\max}$ and $DP_{S_i}^{\max}$ are less than the bounds given in Figure 4.10, then the SPN does not have a partition-based backdoor holding with independent round keys.

Moreover, if the values $LP_{S_i}^{\max}$ and $DP_{S_i}^{\max}$ are *significantly* less than the bounds in Figure 4.10, then the SPN is unlikely to be a *probabilistic* partition-based backdoor cipher. For instance, this criteria can be used to prove that the AES [39] is not a (probabilistic) partition-based backdoor cipher. As explained in [23], its diffusion layer is strongly proper over 2 rounds. In addition, the maximum linear potential and differential probability of the AES S-box are far below the lower bounds given in Figure 4.10, no matter what the dimension d of the subspace V is. As a consequence, this S-box does not map any linear partition to another one.

The results of this chapter can also be used to recover a partition-based backdoor. Consider an S-box S mapping a linear partition $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Paying particular attention to the correlation and differential probability matrices of S , it should not be difficult to recover the subspaces V and W . Indeed, Theorems 4.17 and 4.25 suggest to consider the greatest coefficients of these matrices to recover the subspaces V^\perp and V respectively, provided that V is equal to W . However, using Proposition 4.1 and Equations (4.2) and (4.4) it is straightforward to generalize these theorems to S-boxes mapping a linear partition to another one.

The second criteria is due to Calderini and was introduced recently in [23]. In this paper, the author consider translation-based cipher, a family of ciphers introduced in [30] which generalizes our definition of SPN. Intuitively, a translation-based cipher is a substitution-permutation network in which the substitution and diffusion layers can be round-dependent. Moreover, the key-schedule must be surjective for at least one round, which is normally the case. We restate below Calderini's criteria for substitution-permutation networks.

Definition 4.34 (Strongly d -Anti-Invariant S-Box [30]). Let $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be a mapping satisfying $f(0_n) = 0_n$ and let $1 \leq d < n$. The mapping f is said to be *strongly r -anti-invariant* if any proper subspace V of \mathbb{F}_2^n whose image $f(V)$ is also a subspace of \mathbb{F}_2^n has dimension strictly less than $n - d$. An S-box S is said to be *strongly d -anti-invariant* if its equivalent S-box S' defined by the formula $S'(x) = S(x) + S(0)$ is strongly d -anti-invariant.

Remark 4.35. If S is strongly d -anti-invariant, then S is also i -anti-invariant for each $1 \leq i \leq d$. As proven in [4], S is strongly 1-anti-invariant if and only if $LP_S^{\max} < 1$.

Proposition 4.36. Assume that for each S-box S_i , there exists an integer $r_i < n$ such that the following two conditions hold:

- $2^n \times \text{DP}_{S_i}^{\max} \leq 2^{r_i}$,
- S_i is strongly $(r_i - 1)$ -anti-invariant.

The permutation σ maps a linear partition $\mathcal{L}(V)$ to another linear partition $\mathcal{L}(W)$ if and only if both V and W are walls.

Theorem 4.37. Suppose that the substitution layer satisfies the two conditions of Proposition 4.36 and that the diffusion layer is strongly proper over r rounds. Then, the SPN is not a partition-based backdoor cipher.

This result is complementary to our criteria since it gives other conditions. However, we will see in the next chapter that it gives no information about probabilistic partition-based backdoor ciphers.

Backdoored Encryption Algorithm 1

BEA-1 (standing for *Backdoored Encryption Algorithm*) is a real-size probabilistic partition-based backdoor ciphers whose design relies on the theory developed in Chapters 3 and 4. This cipher is largely inspired by the AES, the current standard of block ciphers, and is proven to be practically secure against linear and differential cryptanalysis. Nonetheless, the backdoor enables recovery of the full 120-bit cipher key in just a few seconds on a laptop computer using only 2^{16} chosen plaintext blocks. The success probability of this cryptanalysis was experimentally verified to be greater than 95%.

This chapter is organized as follows. First, the specification of the cipher BEA-1 and its security analysis against linear and differential cryptanalysis are given in Section 5.1. Next, Section 5.2 explores the hidden property of the algorithm and its design. Secondly, the main idea of the cryptanalysis is illustrated and formalized in Section 5.3. The full cryptanalysis of BEA-1 is then detailed in Section 5.4. To conclude, we compare our attack to Harpes' partitioning cryptanalysis and expose some advantages of probabilistic partition-based backdoors. Our cipher BEA-1 was introduced as a challenge in [10]. Its cryptanalysis was then outlined in [11] and described in [12].

5.1. Presentation of BEA-1

The cipher BEA-1 is directly inspired by *Rijndael* [39], the block cipher designed by Joan Daemen and Vincent Rijmen, now known as the AES [85]. Our algorithm encrypts 80-bit plaintext blocks using a 120-bit cipher key. Unlike the AES, the internal state is not seen as a matrix of bytes but as an array of 10-bit bundles. Therefore, the message and key spaces are respectively $(\mathbb{F}_2^{10})^8$ and $(\mathbb{F}_2^{10})^{12}$.

5.1.1. Specification of the Encryption Process

The encryption consists in applying eleven times a simple keyed operation called *round function* to the data block. A different 80-bit round key is used for each iteration of the round function. Since the last round is slightly different and uses two round keys, the encryption requires twelve 80-bit round keys. These round keys are derived from the 120-bit cipher key using a *key schedule*.

Algorithm 5 - ExpandKey

Input. The 120-bit cipher key $K = (K_0, \dots, K_{11}) \in (\mathbb{F}_2^{10})^{12}$.

Output. The twelve 80-bit round keys $k^{[0]}, \dots, k^{[11]} \in (\mathbb{F}_2^{10})^8$.

```

1   $(k_0, \dots, k_{11}) \leftarrow (K_0, \dots, K_{11})$ 
2  For  $i$  from 0 to 6 do
3       $x \leftarrow M(k_{12i+8}, \dots, k_{12i+11})$ 
4       $x \leftarrow (S_j(x_j))_{j<4}$ 
5       $x \leftarrow (x_0 \oplus (3^i \bmod 2^{10}), x_1, x_2, x_3)$ 
6       $(k_{12i+12}, \dots, k_{12i+15}) \leftarrow (k_{12i+0}, \dots, k_{12i+3}) \oplus x$ 
7       $(k_{12i+16}, \dots, k_{12i+19}) \leftarrow (k_{12i+4}, \dots, k_{12i+7}) \oplus (k_{12i+12}, \dots, k_{12i+15})$ 
8       $(k_{12i+20}, \dots, k_{12i+23}) \leftarrow (k_{12i+8}, \dots, k_{12i+11}) \oplus (k_{12i+16}, \dots, k_{12i+19})$ 
9  For  $r$  from 0 to 11 do
10      $k^{[r]} \leftarrow (k_{8r+i})_{i<8}$ 
11 Return  $k^{[0]}, \dots, k^{[11]}$ 
    
```

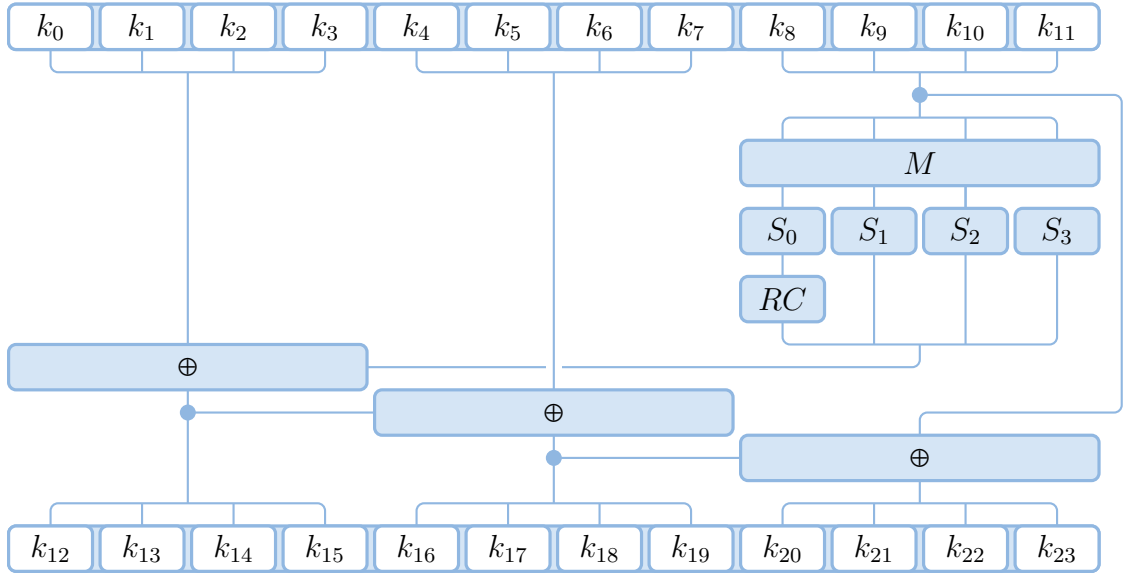


Figure 5.1: The key schedule of BEA-1.

Algorithm 6 – Encrypt

Input. The 120-bit master key $K \in (\mathbb{F}_2^{10})^{12}$ and the 80-bit plaintext block $p \in (\mathbb{F}_2^{10})^8$.

Output. The 80-bit ciphertext block $c \in (\mathbb{F}_2^{10})^8$.

```

1   $k^{[0]}, \dots, k^{[11]} \leftarrow \text{ExpandKey}(K)$ 
2   $x \leftarrow p$ 
3  For  $r$  from 0 to 9 do
4       $x \leftarrow x \oplus k^{[r]}$                                 AddRoundKey
5       $x \leftarrow (S_{i \bmod 4}(x_i))_{i < 8}$                 SubBundles
6       $x \leftarrow (x_0, x_5, x_2, x_7, x_4, x_1, x_6, x_3)$     ShiftRows
7       $x \leftarrow (M \parallel M)(x)$                         MixColumns
8   $x \leftarrow x \oplus k^{[10]}$                                 AddRoundKey
9   $x \leftarrow (S_{i \bmod 4}(x_i))_{i < 8}$                 SubBundles
10  $x \leftarrow (x_0, x_5, x_2, x_7, x_4, x_1, x_6, x_3)$     ShiftRows
11  $x \leftarrow x \oplus k^{[11]}$                                 AddRoundKey
12 Return  $x$ 

```

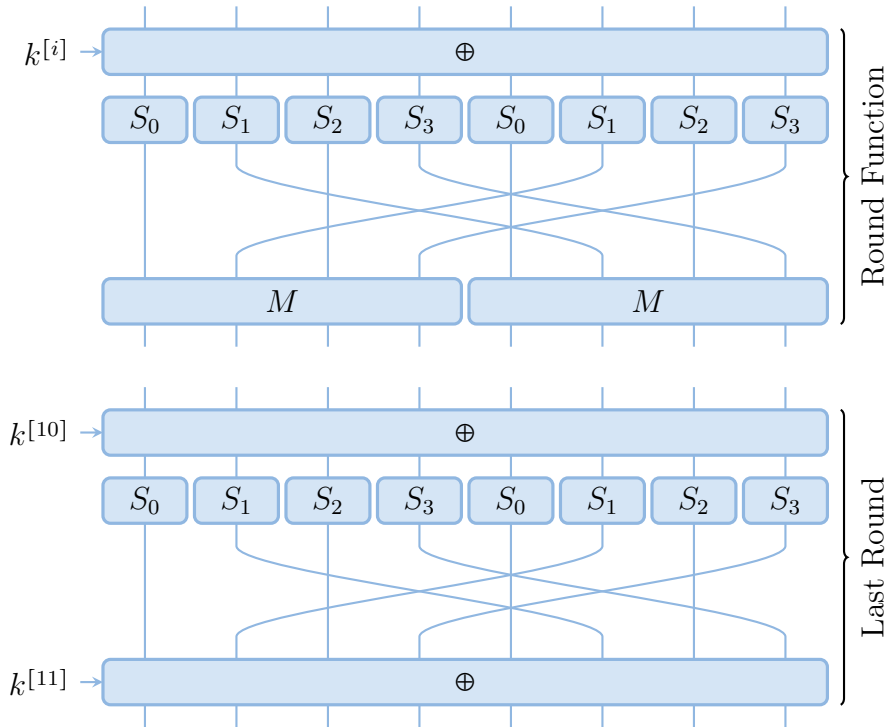


Figure 5.2: The round function of BEA-1.

Like any other Substitution-Permutation Network, the round function is made up of three stages: a *key addition*, a *substitution layer* and a *diffusion layer*.

- The key addition is just a bitwise “exclusive or” (XOR) between the data block and the round key.
- The substitution layer consists in the parallel evaluation of four different 10-bit S-boxes and is the only part of the cipher which is not affine. These S-boxes are referred to as S_0, S_1, S_2, S_3 and are defined in Figures 15, 17, 19 and 21 given in Appendix. They should not be confused with the secret S-boxes $\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2$ and \mathbf{S}_3 , only used in the design and the cryptanalysis of BEA-1.
- Following the design principles of the AES, the diffusion layer comes in two parts: the **ShiftRows** and the **MixColumns** operations. The first part is a bundle permutation. The second evaluates in parallel the linear transformation $M : (\mathbb{F}_2^{10})^4 \rightarrow (\mathbb{F}_2^{10})^4$ processing four 10-bit bundles. Because of its linearity, M is only defined over the standard basis of $(\mathbb{F}_2^{10})^4$ in Figure 13 in Appendix. For convenience, its inverse M^{-1} is also in the same figure.

The pseudo-code for the key schedule is given in Figure 5.1 together with an illustration providing an overview of its structure. This representation also emphasizes the similarities between the key schedules of BEA-1 and Rijndael. In the same way, Figure 5.2 describes the encryption process of BEA-1.

Remark 5.1. The decryption is straightforward from the encryption since all the primitives are bijective. Thus, to decrypt, we just have to apply the inverse operations in the reverse order. It should be stressed that the key addition and the **ShiftRows** are involutions, therefore the same operations are used in the decryption process. Finally, note that the inverse S-boxes are not given here but can be computed by using the equation $S_i^{-1}(S(x)) = x$ holding for each x in \mathbb{F}_2^{10} .

5.1.2. Differential and Linear Cryptanalysis

The differential and linear branch numbers of the linear transformation were recalled in Section 1.5.2. With an exhaustive search, it can be checked that the branch numbers of M are both equal to five, which is the maximum. According to Theorem 1.52, any 2-round trail activates at least five S-boxes. Thus, a 10-round trail activates at least 25 S-boxes.

It is not hard to verify with a computer that every S-box has a maximum differential probability less than $\frac{40}{2^{10}} = 1.25 \times 2^{-5}$ and a maximum linear potential equal to 2^{-4} . Therefore, the differential probability and linear potential of any 10-round trail are upper-bounded by $(1.25 \times 2^{-5})^{25} \approx 1.03 \times 2^{-117}$ and $(2^{-4})^{25} = 2^{-100}$ respectively. Consequently, a differential cryptanalysis of the 10-round version of our cipher would require at least 2^{117} chosen plaintext/ciphertext pairs and a linear cryptanalysis would require 2^{100} known plaintext/ciphertext pairs.

Even if this is a rough approximation since it does not take into account the inter-column diffusion provided by the **ShiftRows** operation, it suffices to prove the cipher’s practical resistance against classical differential and linear cryptanalysis. In fact, there are only 2^{80} different plaintext/ciphertext pairs for a fixed cipher key.

5.2. Design of the Backdoor

The presentation of secret structure of BEA-1 comes in two parts. First, Section 5.2.1 explains the nature of this backdoor and provides all the results needed to address the cryptanalysis. Then, the design of BEA-1's primitives is given in Sections 5.2.2 and 5.2.3. The reader who just wants to understand how the backdoor works can skip these two sections. Indeed, they are more technical and are also independent of the remainder of this chapter.

5.2.1. The Linear Partitions Throughout the Encryption

As said in introduction, the backdoor of BEA-1 relies on the theoretical framework developed in Chapters 3 and 4. Thus, it should not be surprising that linear partitions must play a key role in it. For this purpose, let us introduce the following 5-dimensional subspaces of \mathbb{F}_2^{10}

$$\begin{aligned} V_0 &= \text{span}(266, 343, 3ED, 354, 17F), & W_0 &= \text{span}(16A, 11B, 306, 05E, 0B8), \\ V_1 &= \text{span}(398, 229, 34C, 251, 37B), & W_1 &= \text{span}(04B, 3B7, 0D5, 027, 2C8), \\ V_2 &= \text{span}(0BA, 155, 307, 37E, 318), & W_2 &= \text{span}(1A9, 095, 107, 36F, 2A3), \\ V_3 &= \text{span}(1D1, 21E, 134, 0DC, 15A), & W_3 &= \text{span}(0F0, 2FE, 191, 332, 1A6). \end{aligned}$$

Denote by V and W the 40-dimensional subspaces $\prod_{i=0}^7 V_{i \bmod 4}$ and $\prod_{i=0}^7 W_{i \bmod 4}$ of message space $(\mathbb{F}_2^{10})^8$. Therefore, the linear partitions $\mathcal{L}(V)$ and $\mathcal{L}(W)$ are both made up with 2^{40} cosets, each containing 2^{40} elements.

The S-boxes S_0, S_1, S_2 and S_3 given in the specification of BEA-1 are actually derived from the *secret* S-boxes $\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2$ and \mathbf{S}_3 given in Figures 14, 16, 18 and 20 in Appendix. The relation between the secret S-boxes \mathbf{S}_i and their modified versions S_i will be detailed later in Section 5.2.2. In the first place, let us state the following theorem relating BEA-1 to the theory of partition-based backdoor ciphers.

Theorem 5.2. Consider the encryption function of BEA-1 where the *modified* S-boxes S_0, S_1, S_2 and S_3 are replaced with their *secret* counterparts $\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2$ and \mathbf{S}_3 . Then, the round function preserves the linear partition $\mathcal{L}(V)$ of $(\mathbb{F}_2^{10})^8$ and the last round maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$, no matter the round keys used. As a consequence, the full encryption maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$.

More precisely, Figure 5.3 depicts the evolution of the linear partition $\mathcal{L}(V)$ throughout each primitive of the *secret* encryption process. For instance, we can see that the S-box \mathbf{S}_i maps the linear partition $\mathcal{L}(V_i)$ to $\mathcal{L}(W_i)$ and hence the substitution layer maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Similarly, the diffusion layer comes back to the original partition since it maps $\mathcal{L}(W)$ to $\mathcal{L}(V)$.

Remark 5.3. Theorem 5.2, as well as Theorem 5.4 stated hereinafter, will be proven in Sections 5.2.2 and 5.2.3. Indeed, they establish the main properties of the backdoor and are hence closely related to the design of the cipher's primitives.

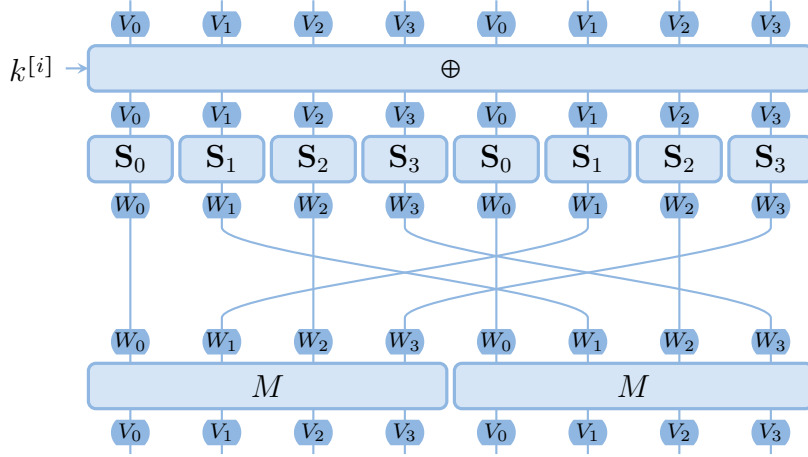


Figure 5.3: The linear partitions throughout the round function.

Thanks to Theorem 5.2, we can now explain our choices for the V_i and W_i . Each of these subspaces of \mathbb{F}_2^{10} is a 5-dimensional linear code whose minimal distance is equal to 4. This property ensures that the Hamming distance of any two different elements lying in the same coset is at least equal to 4. The subspaces V and W of \mathbb{F}_2^{80} inherit this property. Thus, if p is a plaintext, then any other plaintext p' lying in the same coset of V differs from p in at least four bits. Considering the secret encryption function, Theorem 5.2 establishes that their ciphertexts c and c' belong to the same coset of W . Thus, c and c' have at least four different bits. As it will become clear in the next two sections, the subspaces V_i and W_i could have been freely chosen among the 5-dimensional subspaces of \mathbb{F}_2^{10} . We surmised that using linear codes with high minimal distance should reduce the likelihood of observing the backdoor by accident, hence our choice for the V_i and W_i .

Having explained the main property of the secret encryption function, now is the time to introduce the following theorem establishing a link between the secret cipher and BEA-1.

Theorem 5.4. Let \mathbf{F} and \mathbf{E} denote the round function and the encryption function of BEA-1 using the secret S-boxes. Let $p = p^{[0]}$ be any plaintext. Define the following elements with respect to the round keys $k^{[0]}, \dots, k^{[10]}$:

$$p^{[i+1]} = F_{k^{[i]}}(p^{[i]}) \quad \text{and} \quad \mathbf{p}^{[i+1]} = \mathbf{F}_{k^{[i]}}(p^{[i]}) \quad \text{for } 0 \leq i < 11.$$

Assume that the round keys $k^{[0]}, \dots, k^{[10]}$ are independent and uniformly distributed. The probability that all the equations $p^{[i]} = \mathbf{p}^{[i]}$ hold for each $1 \leq i \leq 11$ is given by

$$\left(\left(\frac{944}{1024} \right)^6 \times \left(\frac{925}{1024} \right)^2 \right)^{11} \approx 2^{-11}.$$

Therefore, the probability that p is encrypted equally with E and \mathbf{E} can be approximated by 2^{-11} .

Remark 5.5. The fact that the MixColumns operation is replaced with a key addition in the last round of BEA-1 does not matter in Theorem 5.4. For the sake

of simplicity, we then ignore this detail. This explains why the last round key $k^{[11]}$ does not appear in the statement of this result.

Needless to say, the hypothesis that the round keys are independent and uniformly distributed is mathematically wrong in any practical cryptanalysis. Indeed, the twelve 80-bit round keys are all extracted from one 120-bit cipher key. However, the cipher key needs to have (at least) 960 bits to provide independence and uniform distribution to its round keys. Such a cipher key must be related to the concept of long-key cipher, see Definition 1.8. Nonetheless, if the cipher key is uniformly distributed, the same applies for each round key.

In our cryptanalysis of BEA-1, we are given plaintexts with their ciphertexts encrypted under a fixed cipher key. Even if we forget about the independence of the round keys, each plaintext must be encrypted with a random cipher key to make use of Theorem 5.4.

Fortunately, our experiments suggest that the proportion of the plaintexts encrypted equally with E_K and \mathbf{E}_K is approximately 2^{-11} , even when the round keys are derived from a fixed cipher key K . To put it another way, if \mathcal{P} is a subset of the plaintext space $(\mathbb{F}_2^{10})^8$, it seems reasonable to assume that

$$\#\{p \in \mathcal{P} \mid E_K(p) = \mathbf{E}_K(p)\} \approx \frac{\#\mathcal{P}}{2^{11}}. \quad (5.1)$$

Now, suppose that \mathcal{P} is included in a coset of V denoted by $x + V$. As the secret encryption function \mathbf{E}_K maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$ (see Theorem 5.2), we know that the image of \mathcal{P} under \mathbf{E}_K is included in a coset of W . More precisely, Lemma 3.18 establishes that $\mathbf{E}_K(\mathcal{P})$ is included in $y + W$ where $y = \mathbf{E}_K(x)$. Hence,

$$\{p \in \mathcal{P} \mid E_K(p) = \mathbf{E}_K(p)\} \subseteq \{p \in \mathcal{P} \mid E_K(p) \in (y + W)\}. \quad (5.2)$$

Combining (5.1) with (5.2), we conclude that approximately $\#\mathcal{P} \times 2^{-11}$ ciphertexts in $\mathcal{C} = E_K(\mathcal{P})$ belong to $y + W$. In addition, we have observed that the ciphertexts $c = E_K(p)$ such that $E_K(p) \neq \mathbf{E}_K(p)$ are spread over the 2^{40} cosets of W .

The backdoor of BEA-1 is hence the following. First, choose a set \mathcal{P} of 2^{16} plaintexts uniformly chosen in one coset $x + V$ and collect their ciphertexts $\mathcal{C} = E_K(\mathcal{P})$ encrypted under an unknown cipher key K . Then search for the most represented coset of W in \mathcal{C} and denote by y one of its representatives. According to our experiments, this coset should have roughly $2^{16-11} = 32$ elements and the second most represented coset is unlikely to have more than six elements. As a consequence of the preceding discussion, we know that the coset $x + V$ is mapped to $y + W$ by the secret encryption function \mathbf{E}_K . This information can then be used to recover the cipher key K with a low computation cost, as detailed later in Sections 5.3 and 5.4.

To conclude this section, observe that no particular property of the key schedule has been used. It can be proven that each round of the key schedule preserves the linear partition $\mathcal{L}(\prod_{i=0}^{11} W_{i \bmod 4})$, provided that the S-boxes S_i are replaced with their secret equivalents \mathbf{S}_i . This implies that if two cipher keys K and K' are in the

same coset of $\prod_{i=0}^{11} W_{i \bmod 4}$, then we can approximate the probability that each pair of round keys $k^{[i]}$ et $k'^{[i]}$ are in the same coset of W by $(944^3 \cdot 925 \cdot 2^{-40})^7 \approx 2^{-3.5}$. However, for this property to be easily exploitable, the round keys ought to stay in the same coset of V instead of W (which can be simply achieved by switching the mappings M and $(S_0 \parallel S_1 \parallel S_2 \parallel S_3)$ in the key schedule). Therefore, if compared with our cryptanalysis, this property appears not to be very useful and was intentionally left as a wrong track.

5.2.2. The Substitution Layer

The nature of the hidden property of BEA-1 having been emphasized, this and the following sections detail the design of the cipher's primitives and prove Theorems 5.2 and 5.4 stated above. As explained in introduction, these two sections are aimed at the reader who wants to understand how BEA-1 was made. For a first read, it is possible to jump directly to Section 5.3 explaining the basic principle of the cryptanalysis using the backdoor.

Let $\{0*\}$ and $\{*0\}$ denote respectively the subspaces $\{0_5\} \times \mathbb{F}_2^5$ and $\mathbb{F}_2^5 \times \{0_5\}$ of \mathbb{F}_2^{10} . It should be noted that $\{*0\}$ is a complement space of $\{0*\}$ in \mathbb{F}_2^{10} . The design of each secret S-box \mathbf{S}_i rests upon a permutation \mathbf{S}'_i of \mathbb{F}_2^{10} preserving the linear partition $\mathcal{L}(\{0*\})$. Following Theorem 4.4, we just need to choose a permutation ρ_i of $\{*0\}$ and a family $(\tau_{i,u})_{u \in \{*0\}}$ of permutations of $\{0*\}$. Then, we define \mathbf{S}'_i for all $x = u + v$ in \mathbb{F}_2^{10} to be

$$\mathbf{S}'_i(x) = \mathbf{S}'_i(u + v) = \rho_i(u) + \tau_{i,u}(v),$$

where u is in $\{*0\}$ and v in $\{0*\}$. The permutations ρ_i and $\tau_{i,u}$ were selected following the method given in Section 4.2.3, in order to maximize the resistance of \mathbf{S}'_i against both differential and linear cryptanalysis.

Figure 11 in Appendix defines the linear mappings L_{V_i} and L_{W_i} (for $0 \leq i < 4$) over the standard basis of \mathbb{F}_2^{10} . It is worthwhile to note that these mappings are automorphisms of \mathbb{F}_2^{10} . Moreover, $L_{V_i}(\{0*\}) = V_i$ and $L_{W_i}(\{0*\}) = W_i$. By virtue of Proposition 3.25, we know that L_{V_i} maps $\mathcal{L}(\{0*\})$ to $\mathcal{L}(V_i)$ and that L_{W_i} maps $\mathcal{L}(\{0*\})$ to $\mathcal{L}(W_i)$. Last, but not least, define for each $0 \leq i < 4$ the secret S-box \mathbf{S}_i to be

$$\mathbf{S}_i = L_{W_i} \circ \mathbf{S}'_i \circ (L_{V_i})^{-1}.$$

These S-boxes are given in Figures 14, 16, 18 and 20 in Appendix. Obviously, $(L_{V_i})^{-1}$ maps $\mathcal{L}(V_i)$ to $\mathcal{L}(\{0*\})$, then \mathbf{S}'_i preserves $\mathcal{L}(\{0*\})$, and L_{W_i} maps $\mathcal{L}(\{0*\})$ to $\mathcal{L}(W_i)$. This implies the following proposition.

Proposition 5.6. For each $0 \leq i < 4$, the secret S-box \mathbf{S}_i maps $\mathcal{L}(V_i)$ to $\mathcal{L}(W_i)$.

Remark 5.7. If the reader is interested in an explicit definition of the permutations ρ_i and the families of permutations $(\tau_{i,u})_{u \in \{*0\}}$, they can be recovered in the following way. First, compute $\mathbf{S}'_i = (L_{W_i})^{-1} \circ \mathbf{S}_i \circ L_{V_i}$ using the tables of Figures 11 and 14 (or 16, 18, 20). As noted previously, the permutation \mathbf{S}'_i preserves the linear

partition $\mathcal{L}(\{0*\})$. To obtain its decomposition, we just have to follow the proof of Theorem 4.4. Thus, for each u in $\{*0\}$, define $\rho_i(u)$ to be the unique element of $\{*0\} \cup (\mathbf{S}'_i(u) + \{0*\})$. It is not hard to see that $\rho_i(u)$ is simply equal to the element of \mathbb{F}_2^{10} where the five leftmost bits are exactly the ones of $\mathbf{S}'_i(u)$ and the five remaining bits are all zero. Lastly, for each u in $\{*0\}$, let $\tau_{i,u}$ be the permutation of $\{0*\}$ defined to be $\tau_{i,u}(v) = \mathbf{S}'_i(u + v) + \rho_i(u)$. Again, $\tau_{i,u}(v)$ is just the 10-bit vector having its five leftmost bits all zero and its five rightmost bits identical to the ones of $\mathbf{S}'_i(u + v)$. Naturally, the permutations ρ_i and $\tau_{i,u}$ can be seen as permutations of \mathbb{F}_2^5 (instead of $\{*0\}$ and $\{0*\}$) to obtain the more convenient definition

$$\mathbf{S}'_i(u \parallel v) = (\rho_i(u) \parallel \tau_{i,u}(v)).$$

The modified S-boxes S_i given in the specification of BEA-1 are such that $S_i(x) = \mathbf{S}_i(x)$ for almost all input x in \mathbb{F}_2^{10} . For instance, $S_0(x) = \mathbf{S}_0(x)$ for all except 80 elements x in \mathbb{F}_2^{10} . The images of these 80 particular points are emphasized in Figures 14 and 15. These modifications were chosen so as to improve the differential and linear resistances of S_0 compared to the original secret S-box \mathbf{S}_0 . More generally, S_i and \mathbf{S}_i have 80 different images for i in $\{0, 1, 2\}$. The last modified S-box S_3 is less close to its secret equivalent since S_3 and \mathbf{S}_3 have 99 different images.

Consequently, if x is uniformly distributed over \mathbb{F}_2^{10} , then the equality $S_i(x) = \mathbf{S}_i(x)$ holds with probability q_i where

$$q_0 = q_1 = q_2 = \frac{944}{1024} \quad \text{and} \quad q_3 = \frac{925}{1024}.$$

This implies that when x is uniformly distributed over $(\mathbb{F}_2^{10})^8$, the images of x under the secret and the modified substitution layers are equal with probability $q = (\prod_{i=0}^3 q_i)^2$.

Let $p = p^{[0]}$ be a plaintext. In the following, we use the notations of Theorem 5.4. If $k^{[i]}$ is uniformly distributed, then so is $p^{[i]} + k^{[i]}$. Thus, $p^{[i+1]} = F_{k^{[i]}}(p^{[i]})$ is equal to $\mathbf{p}^{[i+1]} = \mathbf{F}_{k^{[i]}}(p^{[i]})$ with probability q . Assuming moreover that the round keys are independent implies that the events $p^{[i]} = \mathbf{p}^{[i]}$ for each $1 \leq i \leq 11$ are independent. Therefore, the probability that the equalities $p^{[i]} = \mathbf{p}^{[i]}$ hold for all $1 \leq i \leq 11$ is given by q^{11} . This discussion proves Theorem 5.4.

5.2.3. The Diffusion Layer

Some components used to design the linear transformation M are defined over the finite field \mathbb{F}_{2^5} . In order to have an explicit construction of this field, we consider the irreducible polynomial $X^5 + X^2 + 1$ over \mathbb{F}_2 and define \mathbb{F}_{2^5} to be the quotient ring $\mathbb{F}_2[X]/(X^5 + X^2 + 1)$. Let α denote the equivalence class of X in \mathbb{F}_{2^5} . By construction, the equality $\alpha^5 + \alpha^2 + 1 = 0$ holds, or equivalently, $\alpha^5 = \alpha^2 + 1$. Each element of \mathbb{F}_{2^5} can hence be uniquely written as $\sum_{i=0}^4 x_i \alpha^i$ where (x_4, \dots, x_0) belongs to \mathbb{F}_2^5 . More precisely, the family $(\alpha^i)_{i < 5}$ is a basis of \mathbb{F}_{2^5} seen as a 5-dimensional vector space over \mathbb{F}_2 . The field \mathbb{F}_{2^5} will then be identified with $(\mathbb{F}_2)^5$ via the isomorphism from \mathbb{F}_2^5 to \mathbb{F}_{2^5} mapping (x_4, \dots, x_0) to $\sum_{i=0}^4 x_i \alpha^i$. For instance, the element $\alpha^2 + \alpha + 1$ in

\mathbb{F}_{2^5} is identified with 07 in \mathbb{F}_2^5 . Now define the 4×4 matrices M_U and M_V over \mathbb{F}_{2^5} to be

$$\begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix} \quad M_U : \begin{cases} a = \alpha^4 + \alpha^2, \\ b = \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1, \\ c = \alpha^3 + \alpha^2, \\ d = \alpha^4 + \alpha^2 + 1, \end{cases} \quad M_V : \begin{cases} a = \alpha^3 + \alpha^2 + 1, \\ b = \alpha^4 + \alpha^3 + \alpha^2 + \alpha, \\ c = \alpha^4 + \alpha^2 + \alpha, \\ d = \alpha^3. \end{cases}$$

It can be verified that these matrices are MDS. In other words, the $[8, 4]$ -linear code having $G = [\text{Id}_4, M_U]$ as generator matrix has minimal distance equals to 5, which is the maximum achievable.

Each of these matrices naturally induces an automorphism of $(\mathbb{F}_{2^5})^4$, and hence of $(\mathbb{F}_2^{10})^4$. For instance, M_U maps the element $x = (x_0, x_1, x_2, x_3)$ to $x \times M_U$. Observe that we chose to see elements of $(\mathbb{F}_2^{10})^4$ as row vectors to keep the common notations of linear codes.

Example 5.8. To illustrate these notations, let us compute the image of the element $x = (00, 02, 00, 00)$ of $(\mathbb{F}_2^{10})^4$ under the automorphism induced by M_U . First, x is identified with the element $(0, \alpha, 0, 0)$ of $(\mathbb{F}_{2^5})^4$. Then,

$$\begin{aligned} (0, \alpha, 0, 0) \times M_U &= (\alpha(\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1), \alpha(\alpha^4 + \alpha^2), \alpha(\alpha^4 + \alpha^2 + 1), \alpha(\alpha^3 + \alpha^2)) \\ &= (\alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha, \alpha^5 + \alpha^3, \alpha^5 + \alpha^3 + \alpha, \alpha^4 + \alpha^3) \\ &= (\alpha^4 + \alpha^3 + \alpha + 1, \alpha^3 + \alpha^2 + 1, \alpha^3 + \alpha^2 + \alpha + 1, \alpha^4 + \alpha^3). \end{aligned}$$

Therefore, $(00, 02, 00, 00) \times M_U = (1B, 0D, 0F, 18)$. ▀

As was the case for the secret S-boxes \mathbf{S}_i , the linear transformation M rests upon the linear transformation M' defined as follows:

$$\begin{aligned} M' : (\mathbb{F}_2^{10})^4 &\longrightarrow (\mathbb{F}_2^{10})^4 \\ (u_i \parallel v_i)_{i < 4} &\longmapsto (\rho(u)_i \parallel \tau_u(v)_i)_{i < 4} \end{aligned}$$

where $\rho(u) = u \times M_U$ and $\tau_u(v) = v \times M_V + P_{U \rightarrow V}(u)$. The strength of this construction is that M' inherits the linear and differential branch numbers of M_U and M_V , as stated in the proposition hereunder. But first, we introduce the following example.

Example 5.9. Let us compute the image of $x = (000, 070, 000, 000)$ under M' . As a first step, observe that x can be written as

$$x = (00 \parallel 00, 03 \parallel 10, 00 \parallel 00, 00 \parallel 00) = (u_i \parallel v_i)_{i < 4},$$

where $u = (00, 03, 00, 00)$ and $v = (00, 10, 00, 00)$. Let $e_9 = (00, 02, 00, 00)$ and $e_{10} = (00, 01, 00, 00)$. Then $u = e_9 + e_{10}$, it is indeed its decomposition over the standard basis of $(\mathbb{F}_2^5)^4$. Thus, for any linear mapping L , it holds that $L(u) = L(e_9) + L(e_{10})$. The image of u under ρ can hence be computed by

$$\rho(u) = \rho(e_9) + \rho(e_{10}) = (1B, 0D, 0F, 18) + (1F, 14, 15, 0C) = (04, 19, 1A, 14).$$

In the same way,

$$\begin{aligned} \tau_u(v) &= v \times M_V + P_{U \rightarrow V}(e_9) + P_{U \rightarrow V}(e_{10}) \\ &= (16, 0E, 14, 02) + (0F, 11, 0C, 16) + (11, 0E, 02, 0A) = (08, 11, 1A, 1E). \end{aligned}$$

Consequently, $M'(x) = (04 \parallel 08, 19 \parallel 11, 1A \parallel 1A, 14 \parallel 1E) = (088, 331, 35A, 29E)$. \blacktriangleleft

Proposition 5.10. The linear and the differential branch numbers of M' are both equal to 5. Thus, M' is a perfect diffusion layer.

Proof. Let $x = (u_i \parallel v_i)_{i < 4}$ be a nonzero element of $(\mathbb{F}_2^{10})^4$. In order to prove that the differential branch number of M' is equal to 5, we need to show that $w_{10}(x) + w_{10}(M'(x))$ is greater than or equal to 5. First, assume that $u = (u_i)_{i < 4}$ is nonzero. Using the fact that M_U is MDS, we obtain the inequality $w_5(u) + w_5(u \times M_U) \geq 5$. Next,

$$\begin{aligned} 5 \leq w_5(u) + w_5(\rho(u)) &= w_{10}((u_i \parallel 0)_{i < 4}) + w_{10}((\rho(u)_i \parallel 0)_{i < 4}) \\ &\leq w_{10}((u_i \parallel v_i)_{i < 4}) + w_{10}((\rho(u)_i \parallel \tau_u(v)_i)_{i < 4}) = w_{10}(x) + w_{10}(M'(x)). \end{aligned}$$

Now, suppose that $u = 0$. It must be the case that $v \neq 0$ as x is nonzero by definition. Again, it holds that $w_5(v) + w_5(v \times M_V) \geq 5$ because M_V is also MDS. Then,

$$\begin{aligned} 5 \leq w_5(v) + w_5(\tau_0(v)) &= w_{10}((0 \parallel v_i)_{i < 4}) + w_{10}((0 \parallel \tau_0(v)_i)_{i < 4}) \\ &= w_{10}(x) + w_{10}(M'(x)). \end{aligned}$$

We have proven that $w_{10}(x) + w_{10}(M'(x)) \geq 5$ for any nonzero element x of $(\mathbb{F}_2^{10})^4$. Consequently, the differential branch number of M' is greater than or equal to 5. The equality $\mathcal{B}_D(M') = 5$ follows as 5 is the maximum achievable. Similarly, it can be proven that M' has also the maximum linear branch number. It follows that M' is a perfect diffusion layer and the result is proven. \blacksquare

Recall that the notation $\{0*\}$ denotes the subspace $\{0_5\} \times \mathbb{F}_2^5$ and that the linear mappings L_{V_i} and L_{W_i} (see Figure 11) map respectively $\mathcal{L}(\{0*\})$ to $\mathcal{L}(V_i)$ and $\mathcal{L}(\{0*\})$ to $\mathcal{L}(W_i)$. It is then easily seen that M' maps $\{0*\}^4$ to itself. Thus, M' preserves the partition $\mathcal{L}(\{0*\}^4)$ by Proposition 3.25. Finally, define

$$M = (L_{V_0} \parallel L_{V_1} \parallel L_{V_2} \parallel L_{V_3}) \circ M' \circ (L_{W_0} \parallel L_{W_1} \parallel L_{W_2} \parallel L_{W_3})^{-1}.$$

From its definition, it is straightforward to check that M maps the linear partition $\mathcal{L}(\prod_{i=0}^3 W_i)$ to $\mathcal{L}(\prod_{i=0}^3 V_i)$.

Example 5.11. We are going to compute $M(000, 080, 000, 000)$. First, we have

$$\begin{aligned} &(L_{W_0} \parallel L_{W_1} \parallel L_{W_2} \parallel L_{W_3})^{-1}(000, 080, 000, 000) \\ &= (L_{W_0}^{-1}(000), L_{W_1}^{-1}(080), L_{W_2}^{-1}(000), L_{W_3}^{-1}(000)) = (000, 070, 000, 000). \end{aligned}$$

Then, the image of $(000, 070, 000, 000)$ under M' is $(088, 331, 35A, 29E)$, as already established in Example 5.9. Finally,

$$\begin{aligned} M(000, 080, 000, 000) &= (L_{V_0} \parallel L_{V_1} \parallel L_{V_2} \parallel L_{V_3})(088, 331, 35A, 29E) \\ &= (15E, 0BF, 1E2, 04F). \end{aligned}$$

Indeed, $L_{V_0}(088) = L_{V_0}(080) + L_{V_0}(008) = 21D + 343 = 15E$. The three other bundles are computed in the same manner. \blacktriangleleft

Because each mapping L_{V_i} or L_{W_i} operates on different bundles and is invertible, it is clear that the linear and differential branch numbers of M are the same as M' . This discussion completes the proof of the following corollary.

Corollary 5.12. The linear mapping M is a perfect diffusion layer which maps $\mathcal{L}(\prod_{i=0}^3 W_i)$ to $\mathcal{L}(\prod_{i=0}^3 V_i)$.

In conclusion, Proposition 3.23 ensures that any key addition preserves all the linear partitions, and hence it preserves $\mathcal{L}(V)$. Next, it has been proven in Section 5.2.2 that every secret S-box \mathbf{S}_i maps $\mathcal{L}(V_i)$ to $\mathcal{L}(W_i)$. Thus, the secret substitution layer maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. It is clear that the **ShiftRows** operation is linear and maps W to itself. According to Proposition 3.25, this mapping preserves $\mathcal{L}(W)$. Finally, the **MixColumn** operation maps $\mathcal{L}(W)$ to $\mathcal{L}(V)$ by Corollary 5.12. This discussion is summarized in Figure 5.3 and proves Theorem 5.2 previously given in Section 5.2.1.

5.3. Main Idea of the Cryptanalysis

As we have seen in Section 5.2.1, the cipher BEA-1 does not map a linear partition to another one, but behaves as though it did for a non-negligible fraction of the message space. This non-trivial property can be used to recover the cipher key in an operational cryptanalysis close to Harpes's basic partitioning cryptanalysis [52]. But before considering the full cipher, we give the main idea of this attack.

5.3.1. A Detailed Example

To explain how to take advantage of this backdoor, we introduce a toy example. First, let us mention that all the notations of this section are independent of the remainder of this chapter. The message space of this toy cipher is simply \mathbb{F}_2^6 . Then, consider the subspaces V and W of \mathbb{F}_2^6 defined to be

$$\begin{aligned} V &= \text{span}(01, 02, 10, 20) = \{(x_3, x_2, 0, 0, x_1, x_0) \mid x \in \mathbb{F}_2^4\}, \\ W &= \text{span}(01, 02, 04, 10) = \{(0, x_3, 0, x_2, x_1, x_0) \mid x \in \mathbb{F}_2^4\}. \end{aligned}$$

Thus, $\mathcal{L}(V) = \{x + V \mid x \in \{00, 04, 08, 0C\}\}$ and $\mathcal{L}(W) = \{y + W \mid y \in \{00, 08, 20, 28\}\}$.

Let \mathbf{S} be the permutation of \mathbb{F}_2^6 given in Figure 5.4. We define another permutation S of \mathbb{F}_2^6 satisfying $S(x) = \mathbf{S}(x)$ for any input x in \mathbb{F}_2^6 except 00, 01, 04, 05, 08, 09, 0C and 0D. The images of these eight specific points under S are also given in Figure 5.4. By analogy with Section 5.2, the permutation \mathbf{S} represents the *secret* S-box used to design the trapdoor whereas S represents the *modified* S-box given in the specification of the algorithm. Lastly, define the following keyed mappings

$$\begin{aligned} \mathbf{F}_k : \mathbb{F}_2^6 &\longrightarrow \mathbb{F}_2^6 & F_k : \mathbb{F}_2^6 &\longrightarrow \mathbb{F}_2^6 \\ x &\longmapsto \mathbf{S}(x) + k, & x &\longmapsto S(x) + k, \end{aligned}$$

		.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
$\mathbf{S}(x)$	0.	1C	1E	1F	08	39	3A	3C	2A	13	05	02	03	37	20	24	31
	1.	0D	18	0A	1A	3B	2D	29	3E	14	07	11	10	25	26	21	35
	2.	1B	19	0B	1D	2B	2F	2C	28	15	01	16	06	27	36	30	32
	3.	0C	09	0F	0E	3F	2E	3D	38	00	17	04	12	22	23	33	34
$S(x)$	0.	39	05			13	1C			37	20			1E	3A		

Figure 5.4: The secret and modified S-boxes.

representing respectively the secret and the modified round functions. Naturally, the key k can be any element of \mathbb{F}_2^6 .

It can be easily verified that the secret S-box \mathbf{S} maps $\mathcal{L}(V)$ to $\mathcal{L}(W)$. In fact, we have

$$\begin{aligned} \mathbf{S}(00 + V) &= 08 + W, & \mathbf{S}(08 + V) &= 00 + W, \\ \mathbf{S}(04 + V) &= 28 + W, & \mathbf{S}(0C + V) &= 20 + W. \end{aligned}$$

In contrast with the secret permutation \mathbf{S} , the modified S-box S does not map $\mathcal{L}(V)$ to $\mathcal{L}(W)$. However the equality $S(x) = \mathbf{S}(x)$ holds with probability $56/64$ assuming that x is uniformly distributed over \mathbb{F}_2^6 . This can be stated equivalently as

$$\#\{x \in \mathbb{F}_2^6 \mid S(x) = \mathbf{S}(x)\} = 2^6 - 8 = 56.$$

It should also be noted that this statement remains valid when considering their inverse mappings, that is $\#\{y \in \mathbb{F}_2^6 \mid S^{-1}(y) = \mathbf{S}^{-1}(y)\} = 56$. Indeed, if x is an element of \mathbb{F}_2^6 such that $S(x) = \mathbf{S}(x)$, then $y = S(x)$ satisfies the equality $S^{-1}(y) = \mathbf{S}^{-1}(y)$. As a consequence,

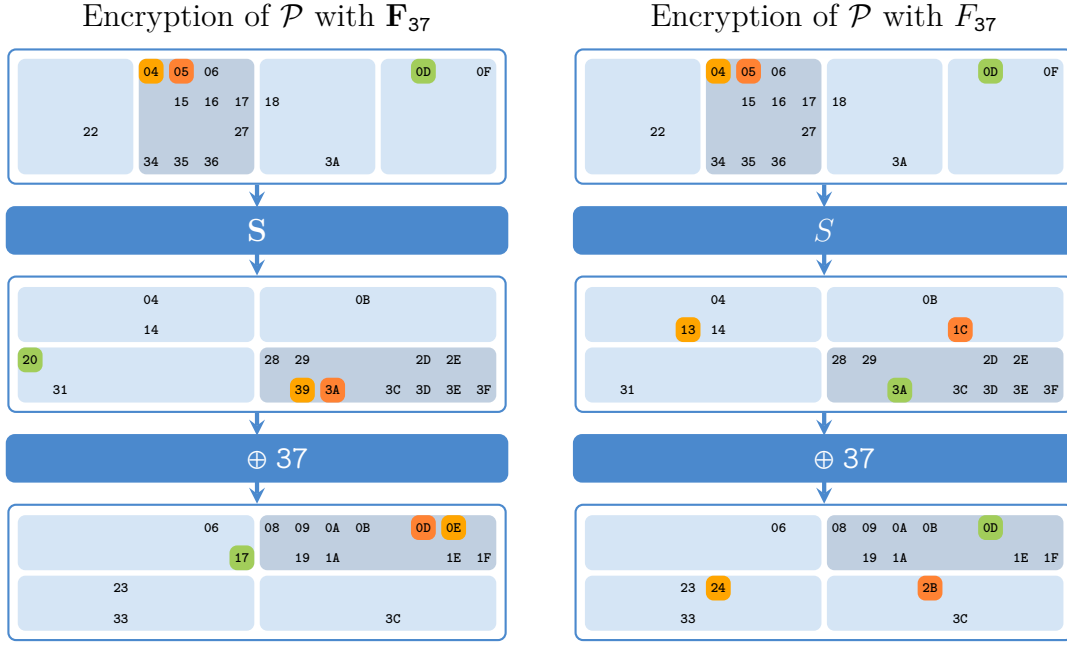
$$\#\{x \in \mathbb{F}_2^6 \mid S(x) = \mathbf{S}(x)\} \leq \#\{y \in \mathbb{F}_2^6 \mid S^{-1}(y) = \mathbf{S}^{-1}(y)\}.$$

The converse inequality can be proven in the same way, establishing the equality.

Now, consider the subset \mathcal{P} of \mathbb{F}_2^6 defined hereinafter. We assume that the round key is $k = 37$. The image of \mathcal{P} under \mathbf{S} and its encryption with \mathbf{F}_{37} are given below.

$$\begin{aligned} & \begin{array}{cccc} \underbrace{\in (00+V)} & \underbrace{\in (04+V)} & \underbrace{\in (08+V)} & \underbrace{\in (0C+V)} \\ \mathcal{P} = \{ & 22, & 04, 05, 06, 15, 16, 17, 27, 34, 35, 36, & 18, 3A, & 0D, 0F \}, \\ \mathbf{S}(\mathcal{P}) = \{ & 0B, & 39, 3A, 3C, 2D, 29, 3E, 28, 3F, 2E, 3D, & 14, 04, & 20, 31 \}, \\ \mathbf{F}_{37}(\mathcal{P}) = \{ & 3C, & 0E, 0D, 0B, 1A, 1E, 09, 1F, 08, 19, 0A, & 23, 33, & 17, 06 \}. \end{array} \\ & \begin{array}{cccc} & \underbrace{\in (28+W)} & \underbrace{\in (08+W)} & \underbrace{\in (20+W)} & \underbrace{\in (00+W)} \end{array} \end{aligned}$$

It should be stressed that the coset $04 + V$ is significantly more represented in \mathcal{P} than any other coset of V . Since \mathbf{F}_{37} maps the linear partition $\mathcal{L}(V)$ to $\mathcal{L}(W)$, the messages belonging to the same coset of V are all mapped to the same coset of W . Therefore, the most represented coset of W in $\mathbf{F}_{37}(\mathcal{P})$ has also ten elements.


 Figure 5.5: Encryption with \mathbf{F}_{37} and F_{37} .

As we have seen above, the modified round function F_{37} does not map $\mathcal{L}(V)$ to $\mathcal{L}(W)$. Figure 5.5 displays the differences between the encryption of \mathcal{P} with \mathbf{F}_{37} and its encryption with F_{37} by highlighting the messages x in \mathcal{P} such that $S(x) \neq \mathbf{S}(x)$ (that is 04, 05 and 0D) and their images throughout the encryption.

To explain these differences, let us first consider the set \mathcal{Q} of the ten messages lying in both \mathcal{P} and $04 + V$. Knowing that the equality $S(x) = \mathbf{S}(x)$ holds with probability $56/64$ when x is uniformly distributed, it seems reasonable to assume that only $10 \times 56/64 = 8.75$ messages of \mathcal{Q} will remain in the same coset when computing their images under S . By comparing with the actual messages in \mathcal{Q} , we can see that this is a good approximation since eight messages in $S(\mathcal{Q})$ belong to the same coset of W .

$$\begin{aligned} \mathcal{Q} &= \{ \text{04, 05}, 06, 15, 16, 17, 27, 34, 35, 36 \} = \mathcal{P} \cap (04 + V), \\ S(\mathcal{Q}) &= \underbrace{\{ \text{13, 1C} \}}_{\notin (28+W)} \cup \underbrace{\{ 3C, 2D, 29, 3E, 28, 3F, 2E, 3D \}}_{\in (28+W)}. \end{aligned}$$

Needless to say, there are also eight messages in $F_{37}(\mathcal{Q})$ lying in the same coset of W because the key addition preserves $\mathcal{L}(W)$.

We focus now to the set \mathcal{P} as a whole. According to the discussion above, we know that the most represented coset of W in $F_{37}(\mathcal{P})$ has at least eight elements. We have seen that the images under S of messages lying in the same coset may not stay together. Nonetheless, the converse can also be true and messages in different cosets may end up in the same coset. This is exactly what happens with the message 0D, as illustrated in Figure 5.5. Consequently, the most represented coset in $F_{37}(\mathcal{P})$ has actually nine elements.

The fact that the most represented coset may not only lose but occasionally

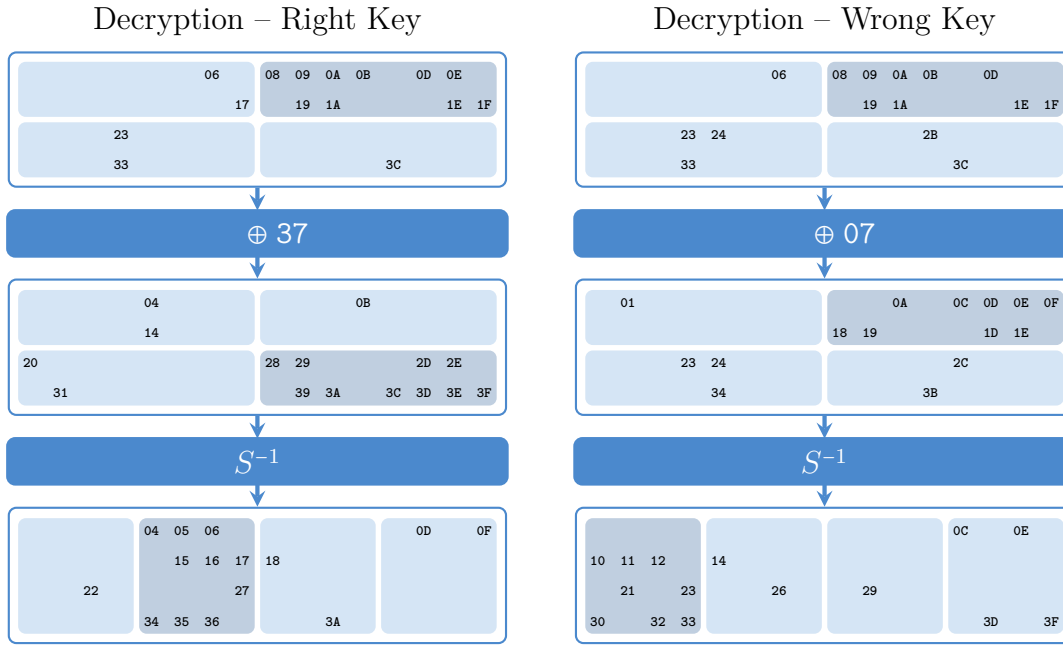


Figure 5.6: Decryption with the right key and with a wrong key.

retrieve elements, should be seen as a side effect. Its impact remains low when

- one coset has significantly more elements than all other cosets (say at least 5 times more), and
- when the number of messages is less than the total number of cosets.

We must nevertheless keep this fact in mind to understand why the right key will not necessarily have the best score.

It is now time to explain how to recover the round key using only the set $\mathcal{C} = F_{37}(\mathcal{P})$ of encrypted messages. First, we have to determine the most represented coset in \mathcal{C} . In our example, this coset is $08 + W$ with nine messages, and $u = 08$ is one of its representatives.

Now, assume that k is the round key used to encrypt \mathcal{C} . We need to find the coset of V which is mapped to $\mathbf{u} + W$ by the secret round function \mathbf{F}_k . According to Lemma 3.18, F_k maps $\mathbf{t} + V$ to $\mathbf{F}_k(\mathbf{t}) + W$. A representative of this coset of V is then $\mathbf{t} = \mathbf{S}^{-1}(\mathbf{u} + k)$. Finally, the *score* of the guessed key k is the number of messages $F_k^{-1}(c) = S^{-1}(c + k)$ which belong to the theoretical coset $\mathbf{t} + V$, that is to say

$$\text{score}(k) = \#\{c \in \mathcal{C} \mid S^{-1}(c + k) \in (\mathbf{t} + V)\}.$$

Figure 5.6 illustrates the scoring process applied to the right key (37) and to a wrong key (07). We naturally recover the set \mathcal{P} and the coset $\mathbf{t} + V = 34 + V = 04 + V$ when using the right key. Thus, the score of $k = 37$ is equal to 10. In the same way, the score of $k = 07$ is the number of decrypted messages in the coset $\mathbf{t} + V = 32 + V = 00 + V$, so $\text{score}(07) = 8$.

Let us now explain why a wrong key tends to have a lower score than the right key. First, the addition of the wrong key randomizes the cosets and the messages within.

Key	0B	12	1C	37	03	05	10	1D	20	21	22	2C	2F	35	36	38
Score	11	10	10	10	9	9	9	9	9	9	9	9	9	9	9	9
Key	3B	3C	3D	00	01	02	04	06	07	08	09	0A	0E	0F	11	13
Score	9	9	9	8	8	8	8	8	8	8	8	8	8	8	8	8
Key	18	19	1E	1F	24	25	26	27	2A	2B	2D	2E	30	34	39	3A
Score	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Key	0C	0D	14	15	16	17	1A	1B	23	28	29	31	32	33	3E	3F
Score	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

Figure 5.7: The scores for each key.

Recall that when the input x is uniformly distributed, the equality $S^{-1}(x) = \mathbf{S}^{-1}(x)$ holds with probability $56/64$. The most represented coset after the addition of the wrong key should then lose some elements by applying S^{-1} . Thus, the score of any wrong key should be less than or equal to 8.

It goes without saying that the previous discussion gives just the main idea of the cryptanalysis. For some wrong keys, the side effects are significant and their scores can even be higher than the score of the right key, as shown in Figure 5.7. Indeed, the key 37 is one the four best keys, but is not the one which has the highest score (0B). For this reason, we will not only return the best key but also the **NbCand** candidate keys having the highest scores when running this cryptanalysis.

5.3.2. Formalization of the Attack

The aim of this section is to formalize and to generalize the cryptanalysis introduced previously in Section 5.3.1. As we have just seen, this attack really begins in Figure 5.6. The very first data needed is the set \mathcal{C} containing the encrypted messages under the unknown key, given by

$$\mathcal{C} = \{04, 05, 06, 0D, 0F, 15, 16, 17, 18, 22, 27, 34, 35, 36, 3A\}.$$

Naturally, \mathcal{C} is included in the set $\mathcal{C} = \mathbb{F}_2^6$ of all possible ciphertexts. Similarly, the set of all possible round keys is denoted by $\mathcal{K} = \mathbb{F}_2^6$. Next, define the keyed mapping

$$\begin{aligned} G : \mathcal{K} \times \mathcal{C} &\longrightarrow \mathbb{F}_2^6 \\ (k, c) &\longmapsto S^{-1}(c + k). \end{aligned}$$

Each mapping $G_k : c \mapsto G(k, c)$ is the inverse of the round function F_k . The secret counterpart of G is $\mathbf{G} : (k, c) \mapsto \mathbf{S}^{-1}(c + k)$. Observe that for each round key k , the mapping \mathbf{G}_k maps $\mathcal{L}(W)$ to $\mathcal{L}(V)$. It is also necessary to know the most represented coset $\mathbf{u} + W$ in \mathcal{C} . Using these notations, the cryptanalysis is formalized in Algorithm 7. Finally, to include potential information on the round keys, this attack processes only a subset \mathcal{K} of \mathcal{K} .

More generally, the parameters can be outlined as follows.

Algorithm 7 – SelectKeys($G, \mathcal{G}, \mathcal{K}, \mathcal{C}, \mathbf{u}, V, \text{NbCand}$)**Input.** See Section 5.3.2.**Output.** The set **Cand** containing the **NbCand** best keys together with their scores.

```

1  Cand  $\leftarrow$  []
2  For each  $k \in \mathcal{K}$  do
    Computation of the score of  $k$ 
3  Score  $\leftarrow$  0
4  For each  $c \in \mathcal{C}$  do
5       $\mathbf{t} \leftarrow \mathbf{G}(k, \mathbf{u})$ 
6      If  $G(k, c)$  lies in  $\mathbf{t} + V$  then
7          Score  $\leftarrow$  Score + 1
    Saving  $k$  if it is one of the  $\text{NbCand}$  best keys
8  If the cardinality of Cand is lower than NbCand then
9      Insert  $(k, \text{Score})$  in Cand
10 Else if Score is greater than the lowest score in Cand then
11     Remove the lowest scored key of Cand
12 Insert  $(k, \text{Score})$  in Cand
13 Return Cand

```

- The sets of all possible keys and ciphertexts are referred to as \mathcal{K} and \mathcal{C} .
- The keyed mapping $G: \mathcal{K} \times \mathcal{C} \rightarrow E$ typically undoes (or partially undoes) one or two rounds of the encryption process.
- Its secret counterpart is denoted by $\mathbf{G}: \mathcal{K} \times \mathcal{C} \rightarrow E$. It is assumed that \mathbf{G}_k maps a linear partition $\mathcal{L}(W)$ to another partition $\mathcal{L}(V)$ no matter the key k used.
- The set of the given ciphertexts is denoted by \mathcal{C} . The set of the keys that must be scored by this attack is denoted by \mathcal{K} .
- It is assumed that there is a coset of W containing significantly more ciphertexts than any other coset. The element \mathbf{u} of \mathcal{C} is a representative of this coset.
- Finally, **NbCand** is the number of candidate keys to return.

Remark 5.13. Taking a closer look at Algorithm 7, we can see that the structure **Cand** requires an efficient way to remove the lowest scored key. In our implementation, **Cand** is a sorted array of couples (s, L) where L is a list containing the keys having the score s . Since there are very few different scores, the sorted insertion in **Cand** is (almost) in constant time. Removing the lowest scored key is also in constant time. Thus, the time complexity of this cryptanalysis is $O(\#\mathcal{K} \times \#\mathcal{C})$.

5.4. Cryptanalysis of BEA-1 Using the Backdoor

The algorithm **SelectKeys** (see Algorithm 7) detailed into the previous section enables recovery of information on the last round key, using the fact that the round function acts as a function mapping a linear partition to another one with high

probability. In this section, we explain how this algorithm can be used to recover the full 120-bit cipher key in just a few seconds on a laptop computer.

This cryptanalysis requires $N = 2^{16}$ chosen plaintexts and their corresponding ciphertexts encrypted under one unknown cipher key K . As BEA-1 operates on 80-bit blocks, this amounts to 2×640 KiB of data. The plaintexts only need to be uniformly chosen in one coset of V and there is no requirement on the cipher key.

Our cryptanalysis is naturally divided in five distinct parts. First, we give a brief overview of each part. By hypothesis, all the plaintexts are in the same coset of V . As explained in Section 5.2.1, a coset of W should be more represented among the ciphertexts. The first part is aimed at finding a representative \mathbf{u} of this coset. The second part consists in using the algorithm **SelectKeys** to find 2^{15} candidates for the full 80-bit last round key $k^{[11]}$. Next, relying on a property of the key schedule, **SelectKeys** is applied to these 2^{15} candidates to find the right last key in a third part. So far, we have recovered 80 bits of the cipher key. Knowing the last round key, it is then possible to undo the last round of each ciphertext. The fourth part is really close to the first one and provides 2^{15} candidates for the 40 remaining bits. Finally, deduce the 2^{15} candidate cipher keys from $k^{[11]}$ and the preceding candidates. The last part involves testing these cipher keys on the plaintext/ciphertext pairs available to find the right one.

The presentation of our cryptanalysis is structured as follows. First, we provide the full attack in Algorithm 5.4. Then, each part of this algorithm is detailed in one dedicated section. It should be noted that we keep the notations of Section 5.2 (and not those of Section 5.3) in the remainder of this chapter.

5.4.1. Part 1: Finding the Right Output Coset

Let \mathcal{P} denote the set of the 2^{16} plaintexts uniformly chosen in one coset of V and let $\mathcal{C} = \{E_K(p) \mid p \in \mathcal{P}\}$ denote the set of their ciphertexts. As said previously, we first need to find the most represented coset of W in \mathcal{C} . Let U_i be the subspace of \mathbb{F}_2^{10} defined to be $U_i = L_{W_i}(\{*\mathbf{0}\})$ for each $0 \leq i < 3$. Since $\{*\mathbf{0}\}$ is a complement space of $\{0*\}$ and L_{W_i} is an automorphism, we know that U_i is a complement space of $L_{W_i}(\{0*\}) = W_i$. Define U to be the subspace $\prod_{i=0}^7 U_{i \bmod 4}$ of $(\mathbb{F}_2^{10})^8$. Of course, U is a complement space of W .

Let c be a ciphertext and $u = (u_i)_{i < 8}$ be in U . Because both U and W are product spaces, it is easily seen that u is the unique representative in U of the coset $c + W$ if, and only if, c_i and u_i are in the same coset of $W_{i \bmod 4}$ for each $i < 8$. We deduce the following efficient way to compute the representative in U of the coset $c + W$. First, precompute the four tables $\text{Rep}W_i$ such that, for each x in \mathbb{F}_2^{10} , $\text{Rep}W_i[x]$ gives the representative in U_i of $x + W_i$. These tables are just arrays of 1024 integers. Then, the representative of $c = (c_i)_{i < 8}$ is just $u = (\text{Rep}W_{i \bmod 4}[c_i])_{i < 8}$.

To find the most represented coset of W in \mathcal{C} , we first compute the representative in U of each ciphertext as described above. Then, we search for the representative which occurs the most. Any naive algorithm should work since there are only 2^{15} representatives.

Algorithm 8 – Cryptanalysis of BEA-1 Using the Backdoor

Input. The number N of plaintext/ciphertext pairs (typically, $N \approx 2^{15}$).

- A set \mathcal{P} of N plaintexts uniformly chosen in one coset of V .
- The corresponding ciphertexts encrypted under one (unknown) cipher key K .
The set $\{E_K(p) \mid p \in \mathcal{P}\}$ of these ciphertexts is denoted by \mathcal{C} .

Output. The cipher key K or "Failure" in case of failure.

1 $\text{NbCand} \leftarrow 2^{15}$

Part 1: find the representative of the output coset.

2 $\lfloor \mathbf{u} \leftarrow$ the element $u \in U$ maximizing the cardinality of $\mathcal{C} \cap (u + W)$

Part 2: find the 2^{15} best candidates for $k^{[11]}$.

3 $E \leftarrow \{3\}$

4 $\text{Cand} \leftarrow \{(k_i)_{i \in E} \mid k_3 \in \mathbb{F}_2^{10}\}$

5 **For each** $\text{idx} \in [7, 0, 4, 1, 5, 2, 6]$ **do**

6 $E \leftarrow E \cup \{\text{idx}\}$

7 Define \mathbf{G}_E , G_E , \mathcal{C}_E and V_E as in Section 5.4.2

8 $\mathcal{K}_E \leftarrow \{(k_i)_{i \in E} \mid k_{\text{idx}} \in \mathbb{F}_2^{10} \text{ and } (k_i)_{i \in E \setminus \{\text{idx}\}} \in \text{Cand}\}$

9 $\lfloor \text{Cand} \leftarrow \text{SelectKeys}(\mathbf{G}_E, G_E, \mathcal{K}_E, \mathcal{C}_E, (\mathbf{u}_i)_{i \in E}, V_E, \text{NbCand})$

Part 3: find $k^{[11]}$ among its candidates.

10 $E \leftarrow \{0, 2, 5, 7\}$

11 Define \mathbf{G} , G and V' as in Section 5.4.3

12 $\text{Cand} \leftarrow \text{SelectKeys}(\mathbf{G}, G, \text{Cand}, \mathcal{C}_E, (\mathbf{u}_i)_{i \in E}, V, \text{NbCand})$

13 $\lfloor k^{[11]} \leftarrow$ the key with the highest score in Cand

Part 4: find the 2^{15} best candidates for $(k'_i)^{[10]}_{4 \leq i < 8}$.

14 Define \mathcal{C}' and \mathbf{u}' as in Section 5.4.4

15 $E \leftarrow \{4\}$

16 $\text{Cand} \leftarrow \{(k'_i)_{i \in E} \mid k'_4 \in \mathbb{F}_2^{10}\}$

17 **For each** $\text{idx} \in [7, 5, 6]$ **do**

18 $E \leftarrow E \cup \{\text{idx}\}$

19 Define \mathbf{G}_E , G_E , \mathcal{C}'_E and V_E as in Section 5.4.4

20 $\mathcal{K}'_E \leftarrow \{(k'_i)_{i \in E} \mid k'_{\text{idx}} \in \mathbb{F}_2^{10} \text{ and } (k'_i)_{i \in E \setminus \{\text{idx}\}} \in \text{Cand}\}$

21 $\lfloor \text{Cand} \leftarrow \text{SelectKeys}(\mathbf{G}_E, G_E, \mathcal{K}'_E, \mathcal{C}'_E, (\mathbf{u}'_i)_{i \in E}, V_E, \text{NbCand})$

Part 5: find the cipher key K .

22 **For each** $(k'_i)^{[10]}_{4 \leq i < 8} \in \text{Cand}$ **do**

23 $(k_i)^{[10]}_{4 \leq i < 8} \leftarrow M((k'_i)^{[10]}_{4 \leq i < 8})$

24 $K \leftarrow$ the cipher key corresponding to $(k_i)^{[10]}_{4 \leq i < 8}$ and $k^{[11]}$

25 **If** $E_K(p) = c$ for all plaintext/ciphertext pairs (p, c) **then**

26 $\lfloor \lfloor$ **Return** K

27 **Return** "Failure"

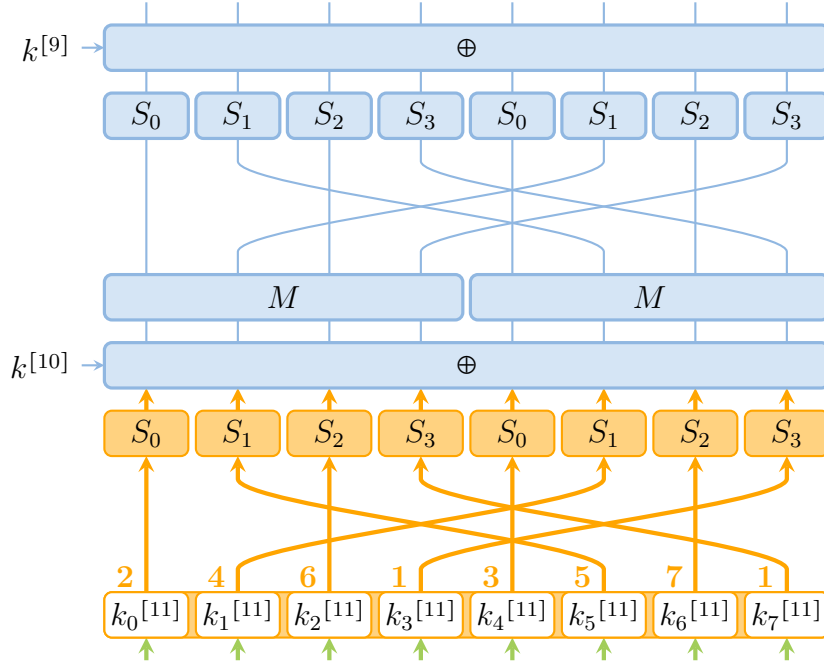


Figure 5.8: Cryptanalysis using the backdoor (Part 2).

5.4.2. Part 2: Obtaining Candidates for the Last Round Key

This part is intended to find candidates for the last round key $k^{[11]}$ using the algorithm **SelectKeys** (see Algorithm 7) to undo the last round of BEA-1. However, if this algorithm is naively applied, then the last round has to be undone for each of the 2^{16} ciphertexts and 2^{80} possible values of $k^{[11]}$, yielding an order of 2^{96} time complexity.

To solve this problem, the 2^{15} candidates for $k^{[11]}$ are obtained bundle by bundle, as illustrated in Figure 5.8. First, we partially decrypt the bundles of index 3 and 7. We begin by these bundles since they both involve the S-box S_3 , being the most different from its secret equivalent. Following the notations of **SelectKeys**, the set containing the ciphertexts is $\mathcal{C}_{\{3,7\}} = \{(c_3, c_7) \mid c \in \mathcal{C}\}$ and the set of the keys is $\mathcal{K}_{\{3,7\}} = \{(k_3, k_7) \mid k_3, k_7 \in \mathbb{F}_2^{10}\}$. The mapping used to partially decrypt the last round of these ciphertexts is

$$G_{\{3,7\}} : (\mathbb{F}_2^{10})^2 \times (\mathbb{F}_2^{10})^2 \longrightarrow (\mathbb{F}_2^{10})^2 \\ ((k_3, k_7), (c_3, c_7)) \longmapsto (S_3^{-1}(c_3 + k_3), S_3^{-1}(c_7 + k_7)).$$

Its secret equivalent $\mathbf{G}_{\{3,7\}}$ is obtained by replacing S_3 with \mathbf{S}_3 . The two remaining inputs of the algorithm are the representative $\mathbf{u} = (\mathbf{u}_3, \mathbf{u}_7)$ of the most represented coset of $(W_3)^2$, and the subspace $(V_3)^2$ of $(\mathbb{F}_2^{10})^2$. It is worth observing that $\mathbf{G}_{\{3,7\}}$ maps $\mathcal{L}((W_3)^2)$ to $\mathcal{L}((V_3)^2)$ as required by the algorithm. Running **SelectKeys** with these arguments generates a set **Cand** containing 2^{15} candidates for $(k_3^{[11]}, k_7^{[11]})$ instead of 2^{20} .

From now on, each step seeks to add a new bundle to our candidates for the last round key $k^{[11]}$. The next bundle to add has index 0. Let E denote the set $\{0, 3, 7\}$

of the current bundle's indices. Since we have no information on the value of $k_0^{[11]}$, the set of the possible values for $(k_i^{[11]})_{i \in E}$ is

$$\mathcal{K}_E = \{(k_i)_{i \in E} \mid k_0 \in \mathbb{F}_2^{10}, (k_3, k_7) \in \mathbf{Cand}\}.$$

Following the idea of the first step, we define $\mathcal{C}_E = \{(c_i)_{i \in E} \mid (c_i)_{i < 8} \in \mathcal{C}\}$ and

$$\begin{aligned} G_E : (\mathbb{F}_2^{10})^E \times (\mathbb{F}_2^{10})^E &\longrightarrow (\mathbb{F}_2^{10})^E \\ ((k_i)_{i \in E}, (c_i)_{i \in E}) &\longmapsto (S_{i \bmod 4}^{-1}(c_i + k_i))_{i \in E}. \end{aligned}$$

Then, define \mathbf{G}_E by replacing S_i with \mathbf{S}_i and let V_E denote the subspace $\prod_{i \in E} V_{i \bmod 4}$ of $(\mathbb{F}_2^{10})^E$. The set \mathbf{Cand} obtained by running **SelectKeys** with these parameters contains 2^{15} candidates for $(k_0^{[11]}, k_3^{[11]}, k_7^{[11]})$.

According to Algorithm 5.4, the index of the next bundle is 4. Actually, the order of the bundle's indices were chosen such as to involve the S-boxes S_3 , then S_0 , S_1 and finally S_2 . The current indices are in the set $E = \{0, 3, 4, 7\}$. Similarly, we define

$$\mathcal{K}_E = \{(k_i)_{i \in E} \mid k_4 \in \mathbb{F}_2^{10}, (k_0, k_3, k_7) \in \mathbf{Cand}\}$$

to include the information on $k^{[11]}$ gathered by the previous step. Finally, define \mathcal{C}_E , G_E , \mathbf{G}_E and V_E as above. Again, the algorithm **SelectKeys** yields 2^{15} candidates for $(k_i^{[11]})_{i \in E}$.

This time, let us take a closer look at the implementation of this step. Because $\#\mathcal{K}_E = 2^{25}$ and $\#\mathcal{C}_E = 2^{16}$, a straightforward implementation of **SelectKeys** requires 2^{41} partial round decryptions, as explained by Remark 5.13. Algorithm 9 provides our implementation of **SelectKeys** for this step. As we can see, the previous candidates are used to filter the ciphertexts before attacking k_4 by brute force. For each of the 2^{15} candidates, initializing the filter requires 2^{16} partial decryptions. On average, it remains roughly 2^6 ciphertexts after the filtering process. The loop over k_4 hence requires 2^{16} partial decryptions. Consequently, this implementation performs about 2^{32} partial decryptions instead of 2^{41} .

Naturally, the 2^{15} candidates for the full round key $k^{[11]}$ are obtained by repeating this method for the four remaining bundles. We will conclude by observing that the complexity of each step decreases since the filtering process improves as the algorithm progresses.

5.4.3. Part 3: Finding the Last Round Key

So far, we have found 2^{15} candidates for the 80-bits key $k^{[11]}$. This part intends to recover the right key among these candidates, relying on the key schedule's structure. Let us consider the last round of the key schedule in order to derive a relation between $k^{[10]}$ and $k^{[11]}$. In Figure 5.1:

- $k^{[9]} = (k_0^{[9]}, \dots, k_7^{[9]})$ corresponds with (k_0, \dots, k_7) ,
- $k^{[10]} = (k_0^{[10]}, \dots, k_7^{[10]})$ corresponds with (k_8, \dots, k_{15}) ,
- $k^{[11]} = (k_0^{[11]}, \dots, k_7^{[11]})$ corresponds with (k_{16}, \dots, k_{23}) .

Algorithm 9 - An implementation of the step $\text{idx}=4$ in part 2.

```

1  Cand  $\leftarrow []$ 
2  For each of the  $2^{15}$  candidates  $(k_0, k_3, k_7)$  for  $(k_0^{[11]}, k_3^{[11]}, k_7^{[11]})$  do
    Initialization of the filter over the ciphertexts
3  Filter  $\leftarrow \emptyset$ 
4   $(\mathbf{t}_0, \mathbf{t}_3, \mathbf{t}_7) \leftarrow (S_0^{-1}(k_0 + \mathbf{u}_0), S_3^{-1}(k_3 + \mathbf{u}_3), S_3^{-1}(k_7 + \mathbf{u}_7))$ 
5  For each  $c \in \mathcal{C}$  do
6       $(t_0, t_3, t_7) \leftarrow (S_0^{-1}(k_0 + c_0), S_3^{-1}(k_3 + c_3), S_3^{-1}(k_7 + c_7))$ 
7      If  $t_0 \in (\mathbf{t}_0 + V_0)$  and  $t_3 \in (\mathbf{t}_3 + V_3)$  and  $t_7 \in (\mathbf{t}_7 + V_3)$  then
8          Filter  $\leftarrow \text{Filter} \cup \{c\}$ 
    Loop over the new bundle of the key
9  For each  $k_4 \in \mathbb{F}_2^{10}$  do
10     Score  $\leftarrow 0$ 
11      $\mathbf{t}_4 \leftarrow S_0^{-1}(k_4 + u_4)$ 
12     For each  $c \in \text{Filter}$  do
13          $t_4 \leftarrow S_0^{-1}(k_4 + u_4)$ 
14         If  $t_4 \in (\mathbf{t}_4 + V_0)$  then
15             Score  $\leftarrow \text{Score} + 1$ 
    Saving  $(k_0, k_3, k_4, k_7)$  if its score is high enough
16     If  $\#\text{Cand} \leq 2^{15}$  then
17         Insert  $((k_0, k_3, k_4, k_7), \text{Score})$  in Cand
18     Else if Score is greater than the lowest score in Cand then
19         Remove the lowest scored key of Cand
20         Insert  $((k_0, k_3, k_4, k_7), \text{Score})$  in Cand
21 Return Cand
    
```

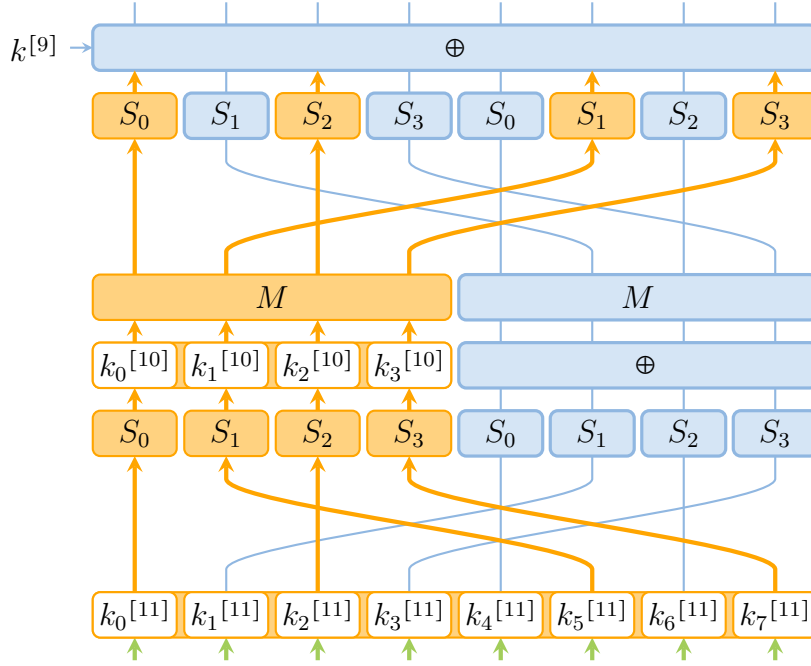


Figure 5.9: Cryptanalysis using the backdoor (Part 3).

It is then easily seen that

$$(k_0^{[10]}, k_1^{[10]}, k_2^{[10]}, k_3^{[10]}) = (k_0^{[11]}, k_1^{[11]}, k_2^{[11]}, k_3^{[11]}) + (k_4^{[11]}, k_5^{[11]}, k_6^{[11]}, k_7^{[11]}).$$

Thus, the 40 leftmost bits of $k^{[10]}$ are determined by $k^{[11]}$. Using this equality, it is possible to partially decrypt the last two rounds for every candidate for $k^{[11]}$. Again, the algorithm **SelectKeys** is used to distinguish between candidates.

Instead of wasting time understanding the definition of G stated hereinafter, we encourage the reader to compare it with Figure 5.9 which speaks for itself. Let us consider

$$\begin{aligned} G' : (F_2^{10})^8 \times (\mathbb{F}_2^{10})^{\{0,2,5,7\}} &\mapsto (\mathbb{F}_2^{10})^4 \\ ((k_i)_{i<8}, (c_i)_{i \in \{0,2,5,7\}}) &\mapsto (S_0^{-1}(c_0 + k_0) + k_0 + k_4, S_1^{-1}(c_5 + k_5) + k_1 + k_5, \\ &\quad S_2^{-1}(c_2 + k_2) + k_2 + k_6, S_3^{-1}(c_7 + k_7) + k_3 + k_7). \end{aligned}$$

Then, let G be the mapping from $(F_2^{10})^8 \times (\mathbb{F}_2^{10})^{\{0,2,5,7\}}$ to $(\mathbb{F}_2^{10})^4$ given by

$$G = (S_0 \parallel S_1 \parallel S_2 \parallel S_3)^{-1} \circ M^{-1} \circ G'.$$

Define \mathbf{G} in the same way as before and let $V' = \prod_{i=0}^3 V_i$. Finally, run **Selectkeys** as in line 12 of Algorithm 5.4. The candidate which has the highest score is then the last round key $k^{[11]}$.

To explain why Parts 2 and 3 of this cryptanalysis are complementary, let us take a closer look at the 2^{15} candidates obtained previously. Most of them are in fact really close to $k^{[11]}$, more precisely, they have at most three bundles different from $k^{[11]}$. This observation is not surprising because when decrypting the last

round, each bundle of the key affects only one bundle of the output. As a direct consequence, close candidates give rise to close one-round decrypted ciphertexts. This explains why the algorithm **SelectKeys**, as used in Part 2, may assign similar scores to close candidates.

By contrast, the mapping G defined above yields very different outputs when used with close candidate keys. Such a property comes from the high diffusion provided by M^{-1} . Thus, this part is more effective where the previous part has its main weakness. Moreover, the side effects are limited here since we decrypt two rounds instead of one.

5.4.4. Part 4: Obtaining Candidates for the Remaining Bits

The round function of the key schedule being bijective, it is sufficient to know the 120 output bits of the last round to compute the cipher key. Until now, we have recovered the last round key $k^{[11]}$, accounting for 80 of these 120 bits. The 40 remaining bits are the 40 rightmost bits of $k^{[10]}$, also denoted by $(k_i^{[10]})_{4 \leq i < 8}$. This fourth part intends to find 2^{15} candidates for these unknown bits.

Since the key $k^{[11]}$ is now known, it is possible to undo the last round for every ciphertext. The cryptanalysis is then reduced to the attack of the second to last round. However, the method used in Part 2 cannot be directly applied here since the second to last round involves the MDS mapping M . Let x and k be elements of $(\mathbb{F}_2^{10})^4$ and observe that

$$M(x) + k = M(x) + M(M^{-1}(k)) = M(x + M^{-1}(k)) = M(x + k')$$

where $k' = M^{-1}(k)$. Thus, the key addition and the mapping M can be switched provided that the key is replaced. According to this observation, define

$$(k_i'^{[10]})_{4 \leq i < 8} = M^{-1}((k_i^{[10]})_{4 \leq i < 8}).$$

Therefore, the last two rounds of BEA-1 can equivalently be represented as in Figure 5.10.

Thanks to this representation, candidates for the key $(k_i'^{[10]})_{4 \leq i < 8}$ can be obtained using **SelectKeys** as in Part 2. To this end, we first need to partially undo the last round using $k^{[11]}$. Following Figure 5.10, define

$$\begin{aligned} f : (\mathbb{F}_2^{10})^{\{1,3,4,6\}} &\longrightarrow (\mathbb{F}_2^{10})^4 \\ (c_i)_{i \in \{1,3,4,6\}} &\longmapsto M^{-1}(S_0^{-1}(c_4 + k_4^{[11]}), S_1^{-1}(c_1 + k_1^{[11]}), \\ &\quad S_2^{-1}(c_6 + k_6^{[11]}), S_3^{-1}(c_3 + k_3^{[11]})). \end{aligned}$$

The set $\{f((c_i)_{i \in \{1,3,4,6\}}) \mid c \in \mathcal{C}\}$ of these “new” ciphertexts is denoted by \mathcal{C}' and the corresponding coset representative is $\mathbf{u}' = \mathbf{f}((\mathbf{u}_i)_{i \in \{1,3,4,6\}})$. To be more consistent with Figure 5.10, the bundles of \mathbf{u}' and of the elements of \mathcal{C}' are indexed from 4 to 7 included. The remainder of the attack is similar to Part 2 as the candidates are obtained bundle by bundle. The first step gets candidates for the bundle’s indices

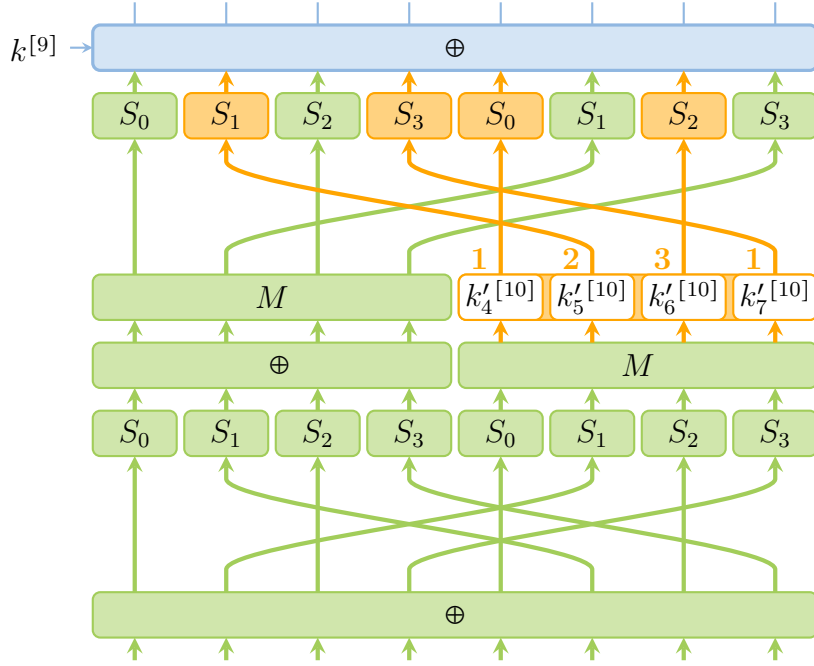


Figure 5.10: Cryptanalysis using the backdoor (Part 4).

4 and 7. The second and the third steps add the indices 5 and 6 respectively. If E denotes the set of the current bundle's indices, then the parameters of **SelectKeys** are the set $\mathcal{C}'_E = \{(c'_i)_{i \in E} \mid (c'_i)_{4 \leq i < 8} \in \mathcal{C}'\}$, the mapping

$$\begin{aligned} G_E : (\mathbb{F}_2^{10})^E \times (\mathbb{F}_2^{10})^E &\longrightarrow (\mathbb{F}_2^{10})^E \\ ((k'_i)_{i \in E}, (c'_i)_{i \in E}) &\longmapsto (S_{i \bmod 4}^{-1}(c'_i + k'_i))_{i \in E}, \end{aligned}$$

its equivalent \mathbf{G}_E and the subspace $V_E = \prod_{i \in E} V_{i \bmod 4}$ of $(\mathbb{F}_2^{10})^E$. The other details are given in Algorithm 5.4. At the end of this part, every candidate $k' = (k'_i)_{4 \leq i < 8}$ for $(k'_i)^{[10]}_{4 \leq i < 8}$ gives rise to a candidate $k = M(k')$ for $(k_i)^{[10]}_{4 \leq i < 8}$.

5.4.5. Part 5: Deducing the Cipher Key

Concatenating the candidates for $(k_i)^{[10]}_{4 \leq i < 8}$ with $k^{[11]}$ yields 2^{15} candidates for the output of the key schedule's last round. To obtain the corresponding candidates for the cipher key, we need to reverse the rounds of the key schedule.

Referring to Figure 5.1, the i -th round of the key schedule maps the element (X_0, X_1, X_2) of $(\mathbb{F}_2^{40})^3$ to (Y_0, Y_1, Y_2) according to the following equalities

$$Y_0 = X_0 + f_i(X_2), \quad Y_1 = Y_0 + X_1, \quad Y_2 = Y_1 + X_2,$$

where f_i denotes the permutation of $(\mathbb{F}_2^{10})^4$ defined for each X to be

$$f_i(X) = (3^i \bmod 2^{10}, 0, 0, 0) + (S_0 \parallel S_1 \parallel S_2 \parallel S_3) \circ M(X).$$

Using these notations, it easily seen that

$$X_0 = Y_0 + f_i(Y_1 + Y_2), \quad X_1 = Y_0 + Y_1, \quad X_2 = Y_1 + Y_2.$$

These equalities describe how to reverse each round of the key schedule, and thus how to recover the 2^{15} candidate cipher keys.

Finally, it just remains to test these candidate cipher keys to complete the cryptanalysis. To be efficient, choose one plaintext/ciphertext pair (p, c) and check whether or not the encryption of p under the candidate K is equal to c . In case of equality, repeat this process for all pairs available to prevent false positive results. Otherwise, the candidate is discarded. Obviously, the right cipher key is the one which passes all tests.

5.5. Conclusion

When parallelized and optimized as described in Algorithm 9, our cryptanalysis of BEA-1 recovers the full 120-bit cipher key in about 20 seconds on a laptop computer. Thanks to its small computing time, we performed several times this attack and verified experimentally that its success probability is greater than 95%. When this attack fails, the cryptanalyst can generally still recover the cipher key with the same data but needs more than 2^{15} candidates in each step. Thus, using more candidates increases the success probability but also the time-complexity of the cryptanalysis.

As noted in Section 5.3, the main idea our cryptanalysis is really close to Harpes' partitioning cryptanalysis [52]. However, some significant differences emerge. First, the number of parts in the output partition is assumed to be small in a partitioning cryptanalysis. Typically, this number is equal to 2, 4 or 8. In contrast, the output partition used in our cryptanalysis consists of all the 2^{40} cosets of the subspace W .

Second, partitioning cryptanalysis considers classes of the last round keys where only a few bits influence the output cosets of the messages. Because its complexity is proportional to the number of key classes, a partitioning cryptanalysis is efficient only if this number is reasonably small. In the case of BEA-1, each bit of the last round key impacts the output cosets of the messages. In other words, whenever one bit of the last round key is changed, at least one plaintext is encrypted in another coset of W . Since there are 2^{80} possible last round keys, a basic partitioning cryptanalysis is ineffective on BEA-1. This problem was addressed in the second part of our cryptanalysis (see Section 5.4.2) as we introduced a trick to compute the best round keys bundle by bundle.

Additionally, a partitioning cryptanalysis updates for each key class as many counters as there are cosets in the output partition. In our algorithm **SelectKeys** presented in Section 5.3.2, we manage only one counter per key as we exploit the secret structure of the round function and the output coset of the ciphertext space. Lastly, our attack recovers the full 120-bit cipher key whereas a partitioning cryptanalysis recovers only a few bits of the last round key.

We should now compare probabilistic and non-probabilistic partition-based backdoor ciphers. By virtue of Theorem 5.2, BEA-1 can be transformed into a partition-based backdoor cipher by simply replacing its S-boxes S_i with their secret counterparts \mathbf{S}_i . Now, assume further that

- the last round of BEA-1 includes the **MixColumns** operation,
- the key schedule also uses the secret S-boxes and
- the mappings M and $(\mathbf{S}_0 \parallel \dots \parallel \mathbf{S}_3)$ are switched in the round function of the key schedule (see Figure 5.1).

It is easily seen that the round function of this new version of BEA-1 preserves the linear partition $\mathcal{L}(V)$ and the same applies to the whole encryption function. As observed at the end of section 5.2.1, when two cipher keys K and K' are in the same cosets of $\prod_{i=0}^{11} V_{i \bmod 4}$, then the derived round keys $k^{[i]}$ and $k'^{[i]}$ are pairwise in the same coset of V . As a consequence, this backdoor cipher is vulnerable to the key schedule dependent attack presented in Section 3.1.3. Using one or two known plaintexts, the coset containing the cipher key is obtained with at most 2^{60} encryptions and then the right cipher key is searched within the 2^{60} elements of this coset. Summarizing, this cryptanalysis recovers the cipher key using at most 2^{61} encryptions or 2^{60} on average. By comparison, our cryptanalysis of the probabilistic version requires 2^{16} chosen plaintexts but has a much better time complexity. Moreover, the differential probability and linear potential matrices of the modified S-boxes S_i are much less suspicious than the ones of the secret S-boxes \mathbf{S}_i .

Before concluding this thesis, let us consider the two criteria to prevent partition-based backdoors given in Section 4.4. We begin with Calderini's criterion. As well as the AES, the diffusion layer of BEA-1 is strongly proper over two rounds. It can be proven with an exhaustive search that for each secret S-box \mathbf{S}_i , any subspace V of \mathbb{F}_2^{10} such that $W = \mathbf{S}_i(V) + \mathbf{S}_i(0)$ is also a subspace of \mathbb{F}_2^{10} is at most 6-dimensional. Consequently, the secret S-boxes are strongly 3-anti-invariant. Since the smallest integer r_i such that $2^{10} \times \text{DP}_{\mathbf{S}_i}^{\max} \leq 2^{r_i}$ is 6, the conditions of Theorem 4.37 are not fulfilled and we hopefully cannot prove that non-probabilistic version of BEA-1 is not a partition-based backdoor cipher, as it is one. Nonetheless, it can be verified that every modified S-box S_i is strongly 7-anti-invariant. Calderini's criterion then proves that BEA-1 *is not* a partition-based backdoor cipher. This fact does not contradict Theorem 4.37 since BEA-1 is a *probabilistic* partition-based backdoor cipher. However, this proves that Calderini's criterion does not apply to this broader family of backdoor ciphers.

Let us now consider our criterion given in Theorem 4.33. By simply looking at the maximum differential probability and linear potential of each S-box, we can see that the conditions are not fulfilled neither for BEA-1 or its non-probabilistic version. Moreover, since these maximum values are really close to the bounds of Figure 4.10, our criterion suggests that each S-box *might almost* maps a linear partition associated with a 5-dimensional subspace to another one.

To conclude let us motivate future research around backdoor ciphers. Even if by-design backdoors are undesirable in block ciphers, their study can contribute to design better ciphers and to improve our understanding of classical cryptanalysis. In fact, partition-based backdoor ciphers are closely related to invariant subspace, constant-dimensional subspace trail and partitioning cryptanalysis. We proved in Chapter 3 that plausible partition-based backdoor ciphers must have S-boxes equivalent to imprimitive S-boxes. Then, we showed in Chapter 4 that such S-boxes are either highly resistant to differential cryptanalysis or to linear cryptanalysis but

not both. As a consequence, our study yields unexpected links between differential, linear and partitioning cryptanalysis. Combined with the complementary work of Calderini, we have now two criteria to prove that a cipher does not have a partition-based backdoor but further interesting researches should be dedicated to prove other criteria for the probabilistic version. Along a similar line, a new variation of differential cryptanalysis was recently proposed by Blondeau, Civino and Sala [16]. This new perspective is directly inspired the family of backdoor ciphers based on hidden sums [19]. In addition, searching for backdoors naturally implies to consider different properties than the ones addressed by classical cryptanalysis, thereby increasing the chance of discovering new effective attacks. Finally, it is worth recalling that the question whether backdoors that are both efficient and undetectable can be inserted in practical block ciphers remains open.

Appendix

In this appendix, we give the specifications of the mappings used in our probabilistic partition-based cipher BEA-1.

x	200	100	080	040	020	010	008	004	002	001
$L_{V_0}(x)$	334	259	21D	0E4	193	266	343	3ED	354	17F
$L_{V_1}(x)$	3DA	306	39E	262	080	398	229	34C	251	37B
$L_{V_2}(x)$	295	237	131	3D1	26B	0BA	155	307	37E	318
$L_{V_3}(x)$	290	15D	0F8	2BE	25F	1D1	21E	134	0DC	15A
$L_{W_0}(x)$	3E8	386	067	19C	158	16A	11B	306	05E	0B8
$L_{W_1}(x)$	364	33E	3A7	119	1D2	04B	3B7	0D5	027	2C8
$L_{W_2}(x)$	324	188	3CB	1B0	131	1A9	095	107	36F	2A3
$L_{W_3}(x)$	262	1A5	34E	0B7	3ED	0F0	2FE	191	332	1A6
$(L_{V_0})^{-1}(x)$	3BF	268	0BB	379	17B	055	061	2F9	354	1F2
$(L_{V_1})^{-1}(x)$	13D	0AD	020	2C7	36D	2B4	314	047	0D7	14C
$(L_{V_2})^{-1}(x)$	361	070	133	02A	2B8	3CC	0DC	21A	08B	184
$(L_{V_3})^{-1}(x)$	1E9	3D1	0BE	245	0F6	357	1DA	074	318	26D
$(L_{W_0})^{-1}(x)$	026	0E9	104	29D	351	053	207	3F9	332	187
$(L_{W_1})^{-1}(x)$	142	1B0	070	3D3	196	088	2E0	0B7	2BB	398
$(L_{W_2})^{-1}(x)$	02D	0AA	205	0F1	375	19A	3AF	1F2	339	265
$(L_{W_3})^{-1}(x)$	0A6	3B3	045	32B	1E4	29A	2AD	27A	069	168

Figure 11: The transformation mappings given over the standard basis of \mathbb{F}_2^{10} .

x	$x \times M_U$	$x \times M_V$	$P_{U \rightarrow V}(x)$
(10, 00, 00, 00)	(07, 06, 1E, 17)	(0E, 16, 02, 14)	(07, 01, 1C, 18)
(08, 00, 00, 00)	(11, 03, 0F, 19)	(07, 0B, 01, 0A)	(05, 16, 14, 03)
(04, 00, 00, 00)	(1A, 13, 15, 1E)	(11, 17, 12, 05)	(0A, 01, 1C, 1C)
(02, 00, 00, 00)	(0D, 1B, 18, 0F)	(1A, 19, 09, 10)	(02, 1F, 1E, 1C)
(01, 00, 00, 00)	(14, 1F, 0C, 15)	(0D, 1E, 16, 08)	(01, 1B, 13, 04)
(00, 10, 00, 00)	(06, 07, 17, 1E)	(16, 0E, 14, 02)	(07, 08, 01, 11)
(00, 08, 00, 00)	(03, 11, 19, 0F)	(0B, 07, 0A, 01)	(02, 1E, 1B, 1F)
(00, 04, 00, 00)	(13, 1A, 1E, 15)	(17, 11, 05, 12)	(16, 06, 1E, 0D)
(00, 02, 00, 00)	(1B, 0D, 0F, 18)	(19, 1A, 10, 09)	(0F, 11, 0C, 16)
(00, 01, 00, 00)	(1F, 14, 15, 0C)	(1E, 0D, 08, 16)	(11, 0E, 02, 0A)
(00, 00, 10, 00)	(1E, 17, 07, 06)	(02, 14, 0E, 16)	(1F, 0C, 08, 1B)
(00, 00, 08, 00)	(0F, 19, 11, 03)	(01, 0A, 07, 0B)	(17, 15, 17, 16)
(00, 00, 04, 00)	(15, 1E, 1A, 13)	(12, 05, 11, 17)	(1D, 04, 0E, 00)
(00, 00, 02, 00)	(18, 0F, 0D, 1B)	(09, 10, 1A, 19)	(11, 0E, 19, 15)
(00, 00, 01, 00)	(0C, 15, 14, 1F)	(16, 08, 0D, 1E)	(16, 1F, 06, 14)
(00, 00, 00, 10)	(17, 1E, 06, 07)	(14, 02, 16, 0E)	(0F, 03, 16, 03)
(00, 00, 00, 08)	(19, 0F, 03, 11)	(0A, 01, 0B, 07)	(0B, 12, 03, 0D)
(00, 00, 00, 04)	(1E, 15, 13, 1A)	(05, 12, 17, 11)	(1F, 1D, 1B, 02)
(00, 00, 00, 02)	(0F, 18, 1B, 0D)	(10, 09, 19, 1A)	(18, 12, 0A, 15)
(00, 00, 00, 01)	(15, 0C, 1F, 14)	(08, 16, 1E, 0D)	(17, 05, 05, 05)

Figure 12: The linear mappings over $(\mathbb{F}_2^{10})^4$ associated with M_U , M_V and the linear mapping $P_{U \rightarrow V}$.

x	$M(x)$	$M^{-1}(x)$
(200, 000, 000, 000)	(13E, 20F, 253, 0BC)	(2D8, 209, 353, 243)
(100, 000, 000, 000)	(35C, 13E, 212, 110)	(0F5, 1BD, 210, 210)
(080, 000, 000, 000)	(32C, 199, 2C5, 07A)	(1E9, 3FE, 238, 329)
(040, 000, 000, 000)	(3C6, 010, 0EC, 261)	(002, 246, 2E2, 380)
(020, 000, 000, 000)	(231, 120, 322, 016)	(322, 3FD, 3D5, 0E5)
(010, 000, 000, 000)	(2D9, 10A, 0C4, 095)	(0AD, 337, 3C5, 2D4)
(008, 000, 000, 000)	(215, 11F, 1E0, 2E7)	(08D, 04D, 016, 34C)
(004, 000, 000, 000)	(23F, 15B, 0C7, 0A7)	(1AB, 11E, 05F, 3A4)
(002, 000, 000, 000)	(344, 394, 342, 165)	(1AE, 1E9, 2CB, 245)
(001, 000, 000, 000)	(112, 1BC, 36C, 0C5)	(10B, 221, 09D, 398)
(000, 200, 000, 000)	(0E6, 0ED, 314, 289)	(395, 295, 38D, 129)
(000, 100, 000, 000)	(17E, 011, 198, 3C5)	(2D7, 1F4, 378, 157)
(000, 080, 000, 000)	(15E, 0BF, 1E2, 04F)	(0BD, 1B1, 18E, 2AB)
(000, 040, 000, 000)	(006, 131, 32E, 12B)	(3AA, 29E, 239, 1C0)
(000, 020, 000, 000)	(39A, 062, 38C, 2EB)	(3D9, 069, 21B, 11B)
(000, 010, 000, 000)	(1F4, 1C5, 1FF, 31D)	(06D, 1BE, 3EB, 0BE)
(000, 008, 000, 000)	(022, 37D, 08D, 3D4)	(3D1, 236, 09D, 2F1)
(000, 004, 000, 000)	(13B, 2FA, 328, 38C)	(0EB, 2FD, 3C3, 176)
(000, 002, 000, 000)	(0CC, 32A, 01A, 2DB)	(055, 128, 25A, 17F)
(000, 001, 000, 000)	(237, 252, 004, 0F8)	(07D, 2BB, 037, 3C8)
(000, 000, 200, 000)	(009, 175, 254, 3ED)	(0A6, 050, 36D, 016)
(000, 000, 100, 000)	(2D5, 29F, 072, 04D)	(263, 36C, 361, 369)
(000, 000, 080, 000)	(09A, 1DD, 336, 34B)	(0C8, 111, 34B, 38E)
(000, 000, 040, 000)	(269, 2CC, 27E, 1CD)	(169, 1A1, 02D, 39B)
(000, 000, 020, 000)	(1B2, 0A7, 178, 208)	(009, 1D9, 3CC, 131)
(000, 000, 010, 000)	(189, 2AB, 1A6, 39D)	(141, 222, 031, 28A)
(000, 000, 008, 000)	(0DC, 0B1, 061, 3DE)	(1C7, 3F1, 063, 33C)
(000, 000, 004, 000)	(019, 08E, 280, 1A7)	(084, 128, 167, 20B)
(000, 000, 002, 000)	(38B, 1A6, 221, 260)	(0D0, 34D, 18C, 354)
(000, 000, 001, 000)	(075, 380, 371, 2E9)	(15E, 23B, 378, 376)
(000, 000, 000, 200)	(099, 176, 3BC, 031)	(03D, 208, 27E, 249)
(000, 000, 000, 100)	(38E, 3D2, 2CD, 21C)	(005, 38F, 215, 2DF)
(000, 000, 000, 080)	(1C7, 259, 17E, 0BE)	(14F, 3D2, 0E2, 1C7)
(000, 000, 000, 040)	(165, 3BA, 19B, 0F7)	(211, 2D9, 1B2, 362)
(000, 000, 000, 020)	(37F, 282, 3A4, 3D8)	(13C, 355, 058, 07F)
(000, 000, 000, 010)	(256, 130, 382, 067)	(19A, 0E6, 364, 0F2)
(000, 000, 000, 008)	(370, 1D0, 3CD, 07F)	(322, 319, 244, 300)
(000, 000, 000, 004)	(22D, 1C8, 221, 18B)	(2BE, 1DD, 223, 1FA)
(000, 000, 000, 002)	(058, 044, 3A0, 281)	(04A, 1EC, 1B6, 3B4)
(000, 000, 000, 001)	(28D, 172, 3EA, 24E)	(015, 371, 2DC, 0E2)

Figure 13: Specification of the diffusion M and its inverse M^{-1} .

CHAPTER 5 – BACKDOORED ENCRYPTION ALGORITHM 1

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	0BA	026	0A0	1E1	183	3DB	1A4	084	110	350	085	2E5	3B4	195	359	2E6
01.	33A	26B	209	07E	1CE	2E3	0C0	136	129	0C8	3D6	054	040	3F2	09F	322
02.	11B	07F	139	07D	2CF	02A	268	227	10A	1C5	12B	016	16C	20D	1E7	35B
03.	313	0CD	11E	1E6	117	355	182	0E6	094	1B9	19C	28C	255	336	0AF	19D
04.	2BC	1A9	31B	02E	282	2AE	272	2E9	3AA	1DD	013	2D3	30F	35A	159	1BB
05.	3DD	12A	248	3C7	28B	191	025	173	018	38D	1A1	185	007	156	378	312
06.	10B	143	05D	3FA	038	3DE	081	0F9	2D1	3FB	1C7	302	1DC	16A	2D8	23F
07.	030	1EB	3AF	311	36D	3BD	3C9	348	261	1AF	071	3EE	3BA	3AB	1B8	3CA
08.	290	118	21B	0F6	3FF	122	1B2	360	1D6	1B6	3D4	3BB	3B3	0EA	097	308
09.	3A9	086	0AE	15A	253	058	0BB	3D5	14B	1A3	23E	053	35D	277	384	0E2
0A.	233	2B8	2AF	0D0	1B1	105	0B3	215	2A2	27F	2DB	17E	12C	3A2	18E	2AC
0B.	321	09C	294	04C	036	2F1	3D2	18D	14C	304	128	069	198	2F4	3DC	370
0C.	138	324	23C	1FD	082	247	005	0A3	0F0	273	152	17B	1A0	1C8	04E	132
0D.	12F	0CC	075	10E	3E0	021	1AE	211	3E6	17A	276	289	0A9	123	01F	048
0E.	201	08F	0B1	002	179	32E	120	1AC	1E3	109	079	37C	297	096	12D	323
0F.	165	0AC	18B	0AB	1FF	13D	25B	3D3	111	22B	21C	1BE	187	30E	34A	318
10.	269	343	29F	395	1AD	1D2	023	3ED	1B3	35E	2D7	044	0F1	3F1	310	0A7
11.	287	3C3	2A5	213	3E4	3DA	0FD	140	38E	2C2	154	254	15F	02C	1FB	1ED
12.	1C6	051	062	090	214	230	190	15D	0A1	186	032	0B9	1DA	239	3D1	383
13.	331	06D	02D	009	2FC	3AD	2AA	363	1EF	38F	39A	2DC	3BF	106	39B	31F
14.	03E	0DE	1BC	067	0CF	155	2CE	240	05E	0E8	0C4	149	08C	3E5	2A1	150
15.	1D1	228	3DF	0E0	3F6	193	19B	27D	2B0	35C	0E3	171	180	022	00E	358
16.	161	0EE	365	15B	0C3	2CD	3E1	06C	119	283	31E	2B9	212	226	076	382
17.	38C	1D3	15C	0B2	22C	314	056	216	364	11C	1E9	020	176	389	2F2	073
18.	06F	27E	027	14E	177	26D	1BA	0EC	033	194	3C6	2F9	221	0E1	3F4	0B7
19.	14F	293	144	0FB	2F0	2F3	0F4	1CC	0C6	065	028	315	3E2	2DD	274	0FA
1A.	17C	041	080	2C5	072	08D	339	2A3	1F1	1DF	2F5	28A	015	188	246	206
1B.	091	03F	259	18F	1C3	27B	319	153	0D2	0BD	2D0	064	000	379	2F6	2A9
1C.	142	0F5	3EF	03B	3F8	344	3BC	265	0E7	334	238	08E	0AA	174	267	162
1D.	112	1D0	01C	292	2C0	0E9	2B6	301	0C1	30D	369	1C0	1E4	1F7	08A	2FA
1E.	3CB	34D	2BE	28F	09A	39D	232	262	333	2F8	397	2C4	06E	27A	317	017
1F.	327	26C	325	167	05B	36C	362	004	3F7	0F7	20B	22D	222	2D2	0CA	196
20.	33F	347	17D	349	146	170	367	18A	1DE	0B5	099	3BE	2C1	0BC	2A0	01B
21.	11D	010	342	169	366	2EC	088	361	291	131	2FF	199	18C	3B0	00D	24F
22.	031	063	3B6	281	0A5	070	1CB	07B	270	2CC	398	32B	1C1	396	278	39E
23.	160	0FF	1A2	0D4	024	24B	178	1BD	326	2EF	28D	299	21F	24A	103	042
24.	141	256	229	218	0EB	260	145	050	035	0E5	300	3AE	1E2	34E	223	20A
25.	164	02F	0C5	210	1A6	258	3F5	32D	1B4	2EA	1C4	3D0	381	371	392	101
26.	3C8	3F3	1F2	10F	0D1	1BF	2D6	320	390	25E	249	341	33B	203	3B5	23A
27.	09E	095	2C8	3A6	0F2	263	108	3B9	3E8	3C4	2BB	2B7	36E	13E	2C9	376
28.	014	00F	0DA	133	163	05C	0A4	1E5	019	37D	043	1FC	184	07A	3FE	03D
29.	0FE	25F	26E	3B7	21A	2E8	3B1	1B7	012	2CA	0C2	113	001	271	1D8	275
2A.	16F	1C9	0AD	236	2AB	3CF	3EC	24E	3F0	1D4	3CC	2BF	2C3	338	1B5	25C
2B.	181	052	243	1F3	11F	2EE	332	32C	1CD	3A8	2B4	34F	0D3	305	006	124
2C.	13F	13C	19E	3A0	2DE	2E2	3CE	345	3CD	0EF	205	31A	23D	34C	059	19F
2D.	1E0	307	3F9	217	337	0DB	14D	353	127	0CE	385	114	107	3D7	057	288
2E.	04F	2B2	2CB	039	234	2B5	2E1	32A	2FB	115	116	37B	3A5	092	373	17F
2F.	21E	06B	087	2FD	2ED	2BA	1EA	125	208	16E	2FE	2E0	0B4	3C2	3C1	21D
30.	11A	01D	3EA	047	157	25D	1D9	37F	16D	20E	098	2B1	340	22E	241	078
31.	1F5	0ED	192	298	3A7	30C	1CF	05F	351	0E4	335	046	151	24C	1EE	235
32.	12E	2A6	1A5	061	3A1	29C	011	066	093	03A	38A	1F8	1F0	3B2	134	356
33.	225	20C	3D9	2E4	0A8	0BE	1FE	0FC	0C7	377	2F7	07C	074	045	1E8	05A
34.	36B	36F	37E	375	04D	1FA	257	13B	089	220	399	00B	158	2D5	068	280
35.	357	0DD	0BF	1B0	2A7	23B	1CA	3FC	00A	330	2A4	200	3EB	008	126	0A6
36.	09B	37A	284	2D4	0F3	28E	237	31D	0DF	368	386	060	374	31C	29A	26A
37.	100	394	1F9	04B	391	39F	30B	00C	077	2EB	01A	231	29B	049	202	224
38.	0C9	2DA	2A8	286	06A	189	130	279	1EC	29D	104	387	32F	316	207	137
39.	0DC	02B	1D7	034	354	39C	0B6	329	3E3	3A4	0D9	245	2B3	0D6	33E	252
3A.	0CB	1DB	172	296	14A	04A	244	250	1F6	2AD	2C6	346	09D	388	328	3A3
3B.	2C7	3E7	29E	3C0	0D5	22A	1F4	168	3FD	242	102	3C5	0F8	251	264	2DF
3C.	27C	029	003	38B	10C	380	10D	295	303	197	33C	219	13A	306	166	2D9
3D.	175	19A	0D8	3D8	0A2	26F	3B8	1C2	148	30A	0B8	24D	1A7	121	15E	372
3E.	25A	266	22F	135	0B0	055	01E	3AC	083	285	34B	1D5	3E9	393	2E7	037
3F.	20F	0D7	1A8	1AB	16B	36A	352	204	2BD	08B	147	1AA	35F	03C	309	33D

Figure 14: Specification of the secret S-box S_0 .

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	0BA	026	0A0	1E1	183	3DB	1A4	083	110	350	085	2E5	3B4	195	359	2E6
01.	33A	26B	209	217	1CE	2E3	0C0	136	129	0C8	3D6	054	040	3F2	09F	322
02.	11B	07F	139	07D	2CF	02A	268	227	246	1C5	12B	3B6	16C	20D	1E7	35B
03.	313	0CD	11E	1E6	117	355	182	0E6	094	1B9	19C	28C	2B9	336	0AF	19D
04.	2BC	1A9	31B	02E	282	2AE	272	2E9	3AA	1DD	013	2D3	30F	35A	159	1BB
05.	11C	12A	248	3C7	28B	191	025	173	018	38D	1A1	185	007	156	378	312
06.	0C9	143	05D	3FA	038	3DE	081	0F9	2D1	3FB	1C7	3E0	1DC	16A	2D8	23F
07.	030	1EB	3AF	311	36D	3BD	3C9	348	261	1AF	071	3EE	3BA	3AB	1B8	3CA
08.	22B	118	279	0F6	3FF	122	1B2	360	1D6	1B6	3D4	3BB	3B3	0EA	097	308
09.	3A9	086	0AE	15A	253	058	0BB	3D5	01D	1A3	23E	053	35D	277	384	0E2
0A.	233	2B8	2AF	0D0	1B1	105	0B3	215	2A2	27F	2DB	17E	12C	3A2	18E	2AC
0B.	321	09C	294	04C	036	2F1	3D2	18D	188	349	128	069	198	2F4	3DC	370
0C.	138	324	23C	1FD	082	247	005	0A3	0F0	273	152	17B	1A0	1C8	04E	34C
0D.	12F	0CC	075	10E	290	021	1AE	211	3E6	17A	276	289	3B5	123	01F	048
0E.	201	08F	29A	002	179	32E	120	1AC	1E3	109	079	37C	297	096	12D	323
0F.	165	0AC	18B	0AB	1FF	230	25B	3D3	111	07E	21C	1BE	187	30E	34A	318
10.	269	343	29F	395	1AD	1D2	023	2DE	1B3	35E	2D7	044	206	3F1	310	0A7
11.	287	3C3	2A5	213	3E4	3DA	0FD	140	38E	2C2	154	254	15F	02C	1FB	1ED
12.	1C6	051	062	090	214	14B	190	15D	0A1	186	032	0B9	1DA	239	3D1	383
13.	331	06D	02D	009	2FC	3AD	2AA	363	1EF	38F	39A	2DC	3BF	106	39B	31F
14.	03E	0DE	1BC	067	0CF	155	2CE	240	05E	0E8	0C4	149	08C	3E5	2A1	150
15.	1D1	228	3DF	0E0	3F6	193	19B	27D	2B0	35C	0E3	171	180	022	00E	358
16.	161	0EE	365	15B	0C3	2CD	3E1	06C	119	283	0F1	3B9	212	226	076	382
17.	38C	1D3	15C	0B2	22C	314	056	216	364	3DD	1E9	020	176	389	2F2	073
18.	06F	27E	027	14E	177	26D	1BA	0EC	25A	194	3C6	2F9	221	0E1	3F4	0B7
19.	14F	293	144	0FB	2F0	3ED	0F4	1CC	0C6	065	028	315	3E2	2DD	274	0FA
1A.	0D3	041	080	2C5	072	08D	339	2A3	1F1	1DF	2F5	267	015	0B1	275	21B
1B.	091	03F	259	18F	1C3	27B	319	153	0D2	0BD	2D0	064	000	379	2F6	2A9
1C.	142	0F5	3EF	03B	3F8	344	3BC	265	0E7	334	238	08E	347	174	18C	162
1D.	112	1D0	01C	292	2C0	0E9	2B6	301	0C1	30D	369	1C0	1E4	1F7	08A	2FA
1E.	3CB	34D	2BE	28F	09A	39D	232	262	333	2F8	397	2C4	06E	27A	317	017
1F.	327	26C	325	167	05B	36C	362	004	3F7	0F7	20B	22D	222	2D2	0CA	196
20.	33F	3B2	17D	302	146	170	367	18A	1DE	0B5	099	3BE	2C1	0BC	2A0	01B
21.	11D	010	342	169	366	2EC	088	361	291	131	2FF	199	1CA	3B0	00D	24F
22.	2B7	063	3EB	281	0A5	070	1CB	07B	270	2CC	398	32B	1C1	396	278	39E
23.	160	0FF	1A2	0D4	024	24B	178	1BD	326	2EF	28D	392	21F	24A	10B	042
24.	141	256	229	218	0EB	260	145	050	035	0E5	300	3AE	1E2	34E	223	20A
25.	164	02F	0C5	210	1A6	258	3F5	32D	1B4	2EA	1C4	3D0	381	371	2D9	101
26.	3C8	3F3	1F2	10F	0D1	1BF	2D6	320	390	25E	249	341	33B	203	087	23A
27.	09E	095	2C8	3A6	0F2	263	108	307	3E8	3C4	2BB	14C	36E	13E	2C9	376
28.	014	00F	0DA	133	163	05C	0AA	1E5	019	37D	043	1FC	184	07A	3FE	03D
29.	0FE	25F	26E	3B7	135	2E8	3B1	1B7	012	2CA	0C2	113	001	271	1D8	01A
2A.	16F	1C9	0AD	236	299	3CF	3EC	24E	3F0	1D4	3CC	2BF	2C3	338	1B5	25C
2B.	181	052	243	1F3	11F	2EE	332	32C	034	3A8	2B4	34F	031	305	006	124
2C.	13F	13C	19E	3A0	17C	2E2	3CE	345	3CD	0EF	205	31A	23D	06B	059	19F
2D.	1E0	3D8	3F9	103	337	0DB	14D	353	127	0CE	385	114	107	3D7	057	288
2E.	04F	2B2	2CB	039	234	2B5	2E1	32A	2FB	115	116	37B	3A5	092	373	17F
2F.	21E	2AB	37F	2FD	2ED	2BA	1EA	125	208	16E	33C	0A9	2F3	3C2	3C1	21D
30.	11A	0A4	3EA	047	157	25D	1D9	10A	16D	20E	098	2B1	340	22E	241	078
31.	1F5	0ED	31E	298	3A7	30C	1CF	05F	351	0E4	335	046	151	24C	1EE	235
32.	12E	2A6	1A5	061	3A1	29C	011	066	093	03A	38A	1F8	1F0	084	134	356
33.	225	20C	3D9	2E4	0A8	0BE	1FE	0FC	0C7	377	2F7	07C	074	045	1E8	05A
34.	36B	36F	37E	375	04D	1FA	257	13B	089	220	399	00B	158	2D5	068	280
35.	357	0DD	0BF	1B0	2A7	23B	255	3FC	00A	330	2A4	200	016	008	126	0A6
36.	09B	37A	284	2D4	0F3	28E	237	31D	0DF	368	386	060	374	31C	033	26A
37.	100	394	1F9	04B	391	39F	30B	00C	077	2EB	3E3	231	29B	049	202	224
38.	132	2DA	2A8	286	06A	189	130	13D	1EC	29D	104	387	32F	316	207	137
39.	0DC	02B	1D7	21A	354	39C	0B6	329	285	3A4	0D9	245	2B3	0D6	33E	252
3A.	0CB	1DB	172	296	192	04A	244	250	1F6	2AD	2C6	346	09D	388	328	3A3
3B.	2C7	3E7	29E	3C0	0D5	22A	1F4	168	3FD	242	102	3C5	0F8	251	264	2DF
3C.	27C	029	003	38B	10C	380	10D	295	303	197	1CD	219	13A	306	166	304
3D.	175	19A	0D8	28A	0A2	26F	3B8	1C2	148	30A	0B8	24D	1A7	121	15E	372
3E.	0B4	266	22F	2FE	0B0	055	01E	3AC	14A	2E0	34B	1D5	3E9	393	2E7	037
3F.	20F	0D7	1A8	1AB	16B	36A	352	204	2BD	08B	147	1AA	35F	03C	309	33D

Figure 15: Specification of the modified S-box S_0 .

CHAPTER 5 – BACKDOORED ENCRYPTION ALGORITHM 1

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	021	09B	37A	3AB	0DF	016	1FE	004	07C	3BE	141	397	300	185	00C	1A7
01.	2FA	3AA	235	0B9	003	3CF	14A	18F	356	363	055	2E4	168	0CF	373	379
02.	2CA	33B	16B	393	283	2E0	2B9	3E9	12F	247	3AD	07B	288	146	30F	3C8
03.	15C	01F	22C	0F8	10F	35D	367	343	1EC	047	008	062	2CF	019	36B	148
04.	0B4	2E3	25E	234	0D2	1F8	184	2FF	2EB	2BB	3A1	34F	312	10B	2EA	04D
05.	1B1	2FE	084	3CC	216	337	0D4	08D	21F	035	1F5	32A	1AA	182	24B	1BF
06.	245	257	01E	34E	375	197	292	1DD	14D	190	27E	18D	137	3A3	228	392
07.	010	34C	389	114	3B9	28B	325	210	1E7	30B	388	335	094	088	038	1C2
08.	305	38E	112	0AA	01B	260	3C1	104	30E	3D4	0EF	079	347	382	22E	09D
09.	1E6	087	278	20D	25B	060	215	2C6	3E0	0A1	3F9	179	252	1B5	105	368
0A.	029	1E9	2C4	2C5	037	233	204	133	3BD	20B	37D	1AE	115	116	1B2	2F3
0B.	266	333	08F	050	1B9	328	26F	1EA	1A9	0E6	291	2ED	05E	162	1EE	362
0C.	15B	351	20F	17D	08B	2D5	259	271	14F	2F5	011	3E7	14B	391	248	0B2
0D.	119	3CD	160	23E	06A	0D0	3C3	01C	171	3D3	349	061	16F	0FB	1DF	342
0E.	082	074	218	2E9	3B3	225	2F9	230	020	223	151	0C5	2A9	0FE	096	045
0F.	0F2	0DA	03A	015	049	370	14C	255	369	193	344	20E	164	3A6	03D	387
10.	24C	030	315	3CA	2EE	0C6	02C	203	107	0F1	3FE	244	26C	264	1C6	1C9
11.	0B1	090	36F	28F	1A3	19D	0BE	317	19B	25C	117	0ED	395	0BF	37E	3E4
12.	35C	3FB	103	2E6	36E	11E	213	279	316	38C	277	286	081	068	3D1	1F7
13.	3C5	095	2FC	09F	2B5	332	05C	38A	3B8	09E	2DD	358	19F	111	2A7	2B0
14.	091	329	106	10E	012	273	2EC	033	080	174	2DB	1C7	102	2D3	123	1B0
15.	03F	2D4	364	131	0A6	275	00A	386	052	3DC	339	11A	211	02A	27F	0DD
16.	318	27B	17B	2D7	1E4	285	144	269	3F4	1EF	093	3BB	307	08E	3B0	0EB
17.	209	2CB	0BB	3A5	129	0AC	027	028	3E6	0E5	221	125	159	2B7	0F9	37C
18.	054	32D	3F6	031	053	29F	23C	2A1	0D9	237	11D	232	1B3	1C1	380	2C1
19.	0C7	360	0D6	265	34A	17F	296	3E1	20C	0A2	1F6	207	0CB	040	1D5	026
1A.	200	121	134	2AB	2FB	272	0D7	07E	001	262	27A	1FF	299	3EB	1FA	0A8
1B.	253	006	128	195	14E	289	0F6	3A8	3D2	261	178	3E5	2C0	0B7	303	181
1C.	097	22A	32E	166	306	0FC	139	138	0F7	1AC	1FD	29B	0AF	041	2CC	0CA
1D.	23B	1F2	25D	0EC	314	20A	03C	338	3C6	0C0	158	28C	3E8	21E	06E	263
1E.	0C4	085	1BD	051	3E2	153	013	0F3	2B6	1A8	17C	2DC	2C7	3B7	33C	29E
1F.	0B5	27C	3F2	398	194	099	0A9	320	35A	366	2C2	05D	1F9	226	098	04E
20.	05A	3AC	33E	0E8	0A7	186	100	17E	126	32B	110	05F	1A5	390	3CE	1FC
21.	11F	3D6	3D5	13C	2BD	251	355	065	336	3DF	152	07A	086	1B6	308	188
22.	0DC	124	15F	075	2E7	39E	046	302	32C	2CE	1DA	3AF	267	066	394	12B
23.	06D	371	2AF	12A	378	319	24D	1D7	37F	3A2	21D	157	31A	3FF	238	2DA
24.	071	31B	256	3F3	33D	280	30C	08C	21C	058	1CD	2D6	165	3A0	077	354
25.	022	32F	359	2BC	374	1EB	30A	192	1CF	1BA	06B	0A0	177	183	28E	2A8
26.	29C	130	323	122	331	201	3B1	0BC	25A	0D8	34B	11B	24F	2E8	1F1	3F5
27.	31C	254	346	376	11C	000	243	0C8	381	0E9	22D	01A	161	3D0	07F	1E0
28.	295	175	04F	3C4	1AF	2A2	191	2F7	34D	36C	2E2	3D7	02E	3CB	0F5	2F6
29.	0C1	30D	025	1F3	01D	1D3	06C	13B	109	2DF	38B	31F	18C	0E1	231	10D
2A.	36D	3DB	377	1DB	16D	09C	024	242	072	39B	31D	2C9	149	206	089	0A3
2B.	0EA	057	250	2CD	38F	2A0	0B3	169	12D	309	2D8	2AD	3F0	3F1	1C8	043
2C.	268	2A3	1D6	28A	1CA	324	2AA	02F	1DE	3C7	0D3	274	147	219	02D	2B2
2D.	1D8	13F	383	3DA	3ED	26A	0AE	1DC	301	2A4	350	2F2	0AB	2A6	3D8	014
2E.	2D2	352	108	0E3	270	229	1A1	29D	1BE	06F	002	059	0A4	198	23A	044
2F.	064	258	348	39C	176	2B4	007	3C2	33F	217	287	073	3B2	15E	03B	167
30.	2B8	2D0	340	0F4	0BD	2F0	353	39A	18A	29A	399	246	1CB	02B	1A2	2E1
31.	3FC	212	1B7	032	281	357	120	048	322	3A9	3B6	33A	196	1BB	1FB	19A
32.	1E2	0AD	101	0F0	22F	227	0B6	345	0C2	220	07D	298	3EF	0B8	2F1	0DE
33.	304	0E4	202	0D1	21B	005	12C	0EE	13A	3BF	092	00D	05B	009	37B	365
34.	0DB	2AC	27D	39D	3A7	214	0CC	1AD	2E5	2DE	1D9	1E5	1C0	3DE	140	24A
35.	2B3	26B	1F0	3C0	3A4	04A	039	2C3	0BA	078	1D4	1E3	16A	145	170	2C8
36.	00B	35B	1AB	127	2BF	16E	2BE	241	1E1	063	334	2B1	136	3EE	384	1C5
37.	23D	2D1	042	372	3BA	1ED	0FA	327	0C9	018	1C3	396	3F8	26E	1BC	187
38.	034	3FD	310	118	1D1	076	22B	143	208	38D	39F	0D5	3B4	199	3C9	3B5
39.	0E2	13D	10A	284	156	150	173	155	3DD	15D	0CD	163	1A0	0C3	10C	341
3A.	180	1A6	321	00E	276	03E	25F	3EC	189	3E3	1D0	1CC	26D	205	17A	3FA
3B.	35E	036	35F	2F8	067	2BA	2A5	16C	3D9	2FD	297	18E	113	0FD	313	0E7
3C.	15A	1B8	08A	239	04B	326	083	385	2F4	19C	12E	017	3BC	224	135	290
3D.	09A	311	240	13E	0A5	24E	069	18B	0FF	236	36A	1A4	04C	3AE	1E8	31E
3E.	132	23F	222	070	2AE	3EA	249	023	293	0B0	330	21A	28D	1CE	154	172
3F.	1F4	056	00F	2EF	361	1D2	0E0	1C4	19E	282	1B4	3F7	294	142	2D9	0CE

Figure 16: Specification of the secret S-box S_1 .

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	021	09B	37A	3AB	0DF	016	1FE	004	07C	3BE	141	397	300	185	00C	1A7
01.	2FA	3AA	235	0B9	003	3CF	14A	18F	356	363	173	2E4	168	0CF	373	379
02.	2CA	326	16B	393	283	2E0	2B9	3E9	12F	247	3D8	07B	288	146	30F	267
03.	15C	01F	22C	0F8	10F	35D	367	343	1EC	047	008	062	2CF	3D6	36B	148
04.	0B4	2E3	25E	234	0D2	1F8	184	2FF	2EB	2BB	3A1	34F	312	10B	2EA	04D
05.	1B1	2FE	084	229	216	337	0D4	08D	21F	035	164	32A	1AA	182	24B	1BF
06.	245	257	01E	34E	375	197	292	1DD	14D	190	27E	13D	137	3A3	228	392
07.	010	34C	389	114	3B9	28B	325	210	1E7	30B	388	1A1	094	088	038	1C2
08.	305	38E	112	0AA	01B	260	3C1	104	30E	3D4	0EF	079	347	382	22E	09D
09.	1E6	087	278	20D	25B	060	215	2C6	3E0	055	3F9	179	252	1B5	105	368
0A.	029	1E9	2C4	2C5	037	233	204	133	3BD	20B	37D	1AE	03D	116	1B2	2F3
0B.	266	333	08F	050	1B9	328	26F	1EA	1A9	0E6	291	2ED	05E	162	1EE	362
0C.	15B	351	20F	17D	08B	2D5	259	271	14F	2F5	011	3E7	14B	391	248	0B2
0D.	119	3CD	160	23E	06A	0D0	3C3	01C	171	3D3	349	061	16F	0FB	1DF	342
0E.	082	068	218	2E9	3B3	225	2F9	230	020	223	151	0C5	2A9	0FE	096	045
0F.	0F2	0DA	03A	015	049	370	14C	255	369	193	38A	20E	0B1	3A6	039	387
10.	24C	030	315	3CA	0A1	0C6	02C	203	107	115	3FE	244	26C	264	1C6	1C9
11.	123	090	36F	28F	1A3	19D	0BE	317	19B	25C	117	0ED	395	0BF	37E	3E4
12.	04C	3FB	103	2E6	3C8	11E	3D1	279	316	38C	277	286	081	074	213	1F7
13.	3C5	095	2FC	09F	2B5	332	05C	31F	324	09E	2DD	3FC	19F	111	2A7	2B0
14.	091	329	106	10E	012	273	2EC	341	080	174	2DB	1C7	102	2D3	2EE	1B0
15.	03F	2D4	364	131	0A6	275	00A	386	052	3DC	339	11A	211	02A	27F	0DD
16.	318	27B	17B	2D7	1E4	285	0AC	269	3F4	1EF	093	3BB	307	08E	3B0	0EB
17.	209	2CB	0BB	3A5	129	1CA	027	028	3E6	064	221	125	159	2B7	0F9	37C
18.	054	32D	3F6	031	053	29F	23C	2A1	0D9	237	336	232	1B3	1C1	380	2C1
19.	1DA	360	30C	265	34A	17F	296	3E1	20C	0A2	1F6	207	0F1	040	1D5	026
1A.	200	121	134	2AB	2FB	272	0D7	07E	001	262	27A	1FF	299	3EB	1FA	39F
1B.	253	006	128	36E	14E	289	0F6	3A8	3D2	261	178	3E5	2C0	0B7	303	181
1C.	097	22A	32E	166	306	0FC	139	138	3BF	1AC	1FD	29B	0AF	041	2CC	0CA
1D.	23B	1F2	25D	0EC	314	20A	03C	120	3C6	0C0	158	28C	3E8	21E	06E	263
1E.	0C4	085	1BD	051	3E2	153	013	0F3	2B6	1A8	17C	2DC	2C7	3B7	33C	29E
1F.	0B5	27C	3F2	398	194	099	0A9	320	35A	366	2C2	05D	1F9	226	09E	04E
20.	05A	3AC	33E	0E8	0A7	186	1D8	17E	126	32B	110	05F	1A5	390	3C8	1FC
21.	11F	019	3D5	13C	2BD	251	355	065	1F5	3DF	152	07A	086	1B6	308	188
22.	0DC	124	15F	075	2E7	39E	046	302	32C	2CE	3CC	3AF	208	066	394	12B
23.	06D	371	2AF	12A	378	319	24D	1D7	37F	3A2	21D	157	31A	3FF	3B2	2DA
24.	071	31B	256	3F3	33D	280	144	08C	21C	058	1CD	2D6	165	3A0	077	354
25.	022	32F	359	2BC	374	1EB	30A	192	1CF	1BA	06B	0A0	177	183	28E	2A8
26.	29C	130	323	122	331	201	3B1	0BC	25A	0D8	34B	11B	24F	2E8	1F1	3F5
27.	31C	254	346	376	11C	000	243	0C8	381	0E9	22D	01A	161	3D0	07F	1E0
28.	295	175	04F	3C4	1AF	2A2	191	2F7	34D	36C	2E2	3D7	0F7	18B	0F5	2F6
29.	0C1	30D	025	1F3	01D	1D3	06C	13B	109	2DF	38B	2E5	18C	0E1	231	10D
2A.	36D	3DB	377	1DB	16D	09C	024	242	072	39B	31D	2C9	149	0F0	089	0A3
2B.	0EA	057	250	2CD	38F	2A0	0B3	169	12D	309	2D8	2AD	358	3F1	1C8	043
2C.	268	2A3	1D6	28A	3EC	18D	2AA	02F	1DE	3C7	0D3	274	147	219	02D	2B2
2D.	0CC	13F	383	3DA	3ED	26A	0AE	1DC	301	2A4	350	2F2	0AB	2A6	39A	014
2E.	2D2	352	108	0E3	270	3E3	02E	29D	1BE	06F	002	059	0A4	198	23A	044
2F.	0CB	258	348	39C	176	2B4	007	3C2	33F	217	287	073	238	15E	03B	167
30.	2B8	2D0	340	0F4	0BD	2F0	353	100	18A	29A	399	246	1CB	02B	1A2	2E1
31.	3F0	212	1B7	032	281	357	3AD	048	322	3A9	3B6	33A	196	1BB	1FB	19A
32.	1E2	0AD	101	033	22F	227	0B6	345	0C2	220	07D	298	3EF	0B8	2F1	0DE
33.	304	0E4	202	0D1	21B	005	12C	0EE	13A	0C7	092	00D	05B	009	37B	365
34.	0DB	2AC	27D	3A7	214	338	1AD	335	2DE	1D9	1E5	1C0	3DE	140	24A	
35.	2B3	26B	1F0	3C0	3A4	04A	0A8	2C3	0BA	078	1D4	1E3	16A	145	170	2C8
36.	00B	35B	1AB	127	2BF	16E	2BE	241	1E1	063	334	2B1	136	3EE	3B8	1C5
37.	23D	2D1	042	372	3BA	1ED	0FA	327	0C9	018	1C3	396	3F8	26E	1BC	187
38.	034	3FD	310	118	1D1	076	22B	143	38D	33B	0E5	0D5	3B4	199	3C9	3B5
39.	0E2	195	10A	284	156	150	11D	155	3DD	15D	0CD	163	1A0	0C3	10C	35C
3A.	180	1A6	321	00E	276	03E	25F	0D6	189	206	1D0	1CC	26D	205	17A	3FA
3B.	35E	036	35F	2F8	067	2BA	2A5	16C	3D9	2FD	297	18E	113	0FD	313	0E7
3C.	15A	1B8	08A	239	04B	384	083	385	2F4	19C	12E	017	3BC	224	135	290
3D.	09A	311	240	13E	0A5	24E	069	3CB	0FF	236	36A	1A4	344	3AE	1E8	31E
3E.	132	23F	222	070	2AE	3EA	249	023	293	0B0	330	21A	28D	1CE	154	172
3F.	1F4	056	00F	2EF	361	1D2	0E0	1C4	19E	282	1B4	3F7	294	142	2D9	0CE

Figure 17: Specification of the modified S-box S_1 .

CHAPTER 5 – BACKDOORED ENCRYPTION ALGORITHM 1

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	12E	38B	18E	131	039	10D	2DE	246	286	2BE	315	384	21D	1A5	06D	0CA
01.	2A2	2CE	264	085	374	3BB	3B9	1B7	0DE	3BC	207	002	392	1B5	0BA	318
02.	39C	2EE	13C	125	227	063	27E	126	0AA	082	305	15C	206	0A0	009	3C6
03.	100	3F3	2AD	199	102	108	1DB	30E	310	245	0A8	116	022	3C1	028	332
04.	1E1	2E7	0DA	255	0CB	07C	2A0	240	150	165	258	2C8	0C4	334	36B	2D1
05.	1E0	138	39A	0FF	1A7	10C	353	19B	171	038	3BD	000	3A2	1B8	282	2EB
06.	1D4	3D4	20F	23C	0D7	154	012	0DF	10F	237	04E	155	2E2	189	01E	121
07.	1B4	381	273	123	052	12C	158	033	2D2	3D3	23B	3B8	2F7	160	341	124
08.	337	1E6	3BE	327	0BD	096	2E4	107	1C2	263	2A4	2CF	244	196	36A	16D
09.	0A1	2C3	004	049	303	3A3	09E	361	065	1B0	05D	319	21B	249	2B2	399
0A.	198	26A	080	1B1	340	28A	33C	316	0FC	37F	1A8	134	17F	3DF	34F	3E5
0B.	2D9	32A	34A	1D1	09D	3FB	0BE	3A8	383	036	3B6	222	22E	2B6	3A6	0FA
0C.	1C1	0B2	113	3E8	129	34C	153	333	07E	01F	01D	213	299	0F8	130	1B9
0D.	182	0A2	1A1	1CD	119	210	24C	020	097	3F0	280	112	04C	14D	1EB	307
0E.	386	0AE	322	2FE	217	3D7	1AF	345	05B	3F5	110	1C8	03C	1C5	35A	0C0
0F.	3F1	238	338	1CB	0F4	2B4	00F	3A1	242	03D	1DA	1B3	003	114	3FA	313
10.	35F	0C5	261	2C1	15D	28F	390	1C9	1DD	3C7	14F	11D	066	04D	03B	0E9
11.	2BA	2FD	347	191	044	0B8	194	148	256	360	326	257	1AE	396	09B	2CD
12.	1E7	3CD	1FF	269	040	3E7	08A	216	0C9	33B	3D9	1BC	2B1	325	11B	16F
13.	053	22A	186	180	27D	11F	2A9	13E	3E1	0D4	24E	1D2	2FC	3C9	1FB	31A
14.	3DE	1D7	025	372	339	2C7	2ED	25F	3E6	098	2EF	247	0E8	2D3	105	09F
15.	2CC	36D	31F	24B	1D8	241	068	211	2AF	3EA	355	35C	026	2BD	0B5	0EF
16.	35B	233	05A	1BE	291	368	137	035	298	140	26B	1E4	379	07F	3EB	164
17.	20B	12D	375	1BF	12F	1AA	18B	268	3F4	364	0F7	057	0B9	3C5	060	19D
18.	22B	17C	11C	0B1	23A	3B4	05F	2F5	219	224	3CC	042	06F	39D	218	023
19.	215	177	190	395	274	359	0E2	2E9	397	0F1	010	099	17D	08E	314	317
1A.	0DC	03F	1AC	1A6	132	152	195	3AD	3E9	3C2	019	0F0	0CD	074	178	174
1B.	184	3E0	084	2FB	1A9	0B7	250	27B	06C	13B	0FB	296	297	30F	350	14E
1C.	007	10E	19C	055	351	034	175	103	272	02D	2C0	21C	047	20D	0E3	29B
1D.	13F	1DF	162	376	0BF	1CA	3EC	2B9	3FE	388	133	29D	33A	304	1FE	059
1E.	13D	19A	294	02B	127	1E8	275	07B	14C	018	031	15B	0A3	0EC	27C	087
1F.	38D	3B0	284	1FA	1F5	00A	3E2	02E	228	285	34B	311	075	2F1	1C4	094
20.	3FF	202	27F	2F9	30D	135	33F	301	3D8	2C6	3D2	309	0EA	073	1F1	289
21.	3B5	093	111	0B4	20E	1BA	1F7	24A	394	157	366	336	39B	017	25C	3C4
22.	1EC	2BC	144	1E9	193	16A	33D	344	295	079	2B7	2D4	38A	17A	292	0AC
23.	0F2	35E	1EF	0BB	1BB	071	2DA	3F7	3D1	037	2AB	330	0B0	2DB	07A	22D
24.	00C	149	0AF	290	2E0	122	283	32E	3AE	3C3	1D9	2E5	37B	0BC	265	32D
25.	089	2CB	115	081	18A	0E5	05C	1A3	287	0D0	276	32C	0C3	30B	226	1C0
26.	2F3	0A5	062	2AA	185	091	208	156	230	320	385	28E	21E	3F9	11E	05E
27.	159	281	0C8	37C	0DD	188	04F	26E	33E	2F8	3A0	3B3	3C8	0A9	1A2	3AA
28.	302	36E	38F	19E	212	142	24D	0B3	141	3EF	1CE	262	145	362	346	176
29.	1E3	14B	3A9	3DD	1C6	3F8	070	0D3	1EA	3BA	248	146	201	243	1F6	205
2A.	0A7	20A	2F6	00E	267	26D	2A7	31D	2D0	0D1	38E	006	30A	3E3	2C5	28B
2B.	2D5	3A7	1D5	3A4	101	2D7	34E	2B5	072	26C	090	1F8	1F9	3AF	1F0	0C2
2C.	2C2	21A	06A	0AB	1FC	109	16B	15E	161	38C	1CC	271	279	369	342	1D6
2D.	01A	016	352	173	34D	354	181	235	254	23F	16C	030	03E	1C3	2EA	0CF
2E.	18C	078	18D	01B	117	393	3F2	39E	37D	1BD	24F	10A	29A	0A4	08D	187
2F.	015	046	0C1	251	00D	348	014	21F	001	008	2CA	321	1B2	1B6	043	147
30.	223	0B6	054	1AB	0FD	373	31E	323	20C	151	10B	288	045	041	349	2E3
31.	32F	0ED	277	179	278	3F6	23E	252	077	04A	120	200	308	300	312	04B
32.	1ED	048	30C	183	0D2	39F	3B7	0AD	3FD	204	050	1C7	197	2F2	221	209
33.	1F3	343	051	1F2	169	266	25A	26F	0F3	2B0	095	17B	31B	0F6	0E6	2DC
34.	225	36C	1EE	253	058	0E1	021	31C	3D6	1AD	167	2AC	06B	23D	398	032
35.	35D	2FA	00B	391	239	0F5	335	02C	083	143	02A	29E	36F	214	104	14A
36.	0E4	1E5	19F	2E1	1FD	356	28D	07D	11A	0EE	0EB	370	358	1DC	163	056
37.	367	03A	2D6	363	229	3DA	08B	382	270	2E8	2FF	168	027	2AE	170	28C
38.	2A3	08C	1CF	076	389	32B	2EC	2A5	2C9	2A6	29C	3ED	09C	0CC	2A8	203
39.	0FE	293	29F	2F4	0E7	232	0CE	3AB	13A	011	3DB	220	0D8	1F4	22F	236
3A.	17E	1A4	27A	128	329	324	067	365	024	2B8	3E4	0D6	3AC	3A5	172	306
3B.	16E	0DB	3C0	25B	088	2D8	3BF	380	259	2A1	1E2	377	02F	029	2F0	2BF
3C.	2C4	136	2BB	3B1	1DE	3D5	01C	25E	2B3	069	0A6	106	0D5	3DC	118	2E6
3D.	2DD	1D3	18F	371	064	260	0C7	0E0	0C6	1D0	3EE	0D9	37A	387	3CB	234
3E.	3D0	15F	3B2	08F	15A	013	331	328	06E	25D	0F9	092	166	378	3CE	139
3F.	005	09A	12B	061	231	1A0	3CF	3CA	2DF	192	086	357	22C	12A	3FC	37E

Figure 18: Specification of the secret S-box S_2 .

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	12E	38B	18E	131	039	10D	2DE	246	286	2BE	315	384	21D	142	06D	0CA
01.	2A2	2CE	264	085	374	3BB	3B9	1B7	3E6	3BC	207	002	392	1B5	0BA	318
02.	39C	2EE	1DA	125	019	063	27E	126	19A	082	305	0E3	206	0A0	009	3C6
03.	100	3F3	2AD	199	102	108	1DB	2F0	310	245	0A8	116	022	3C1	028	332
04.	1E1	2E7	0DA	2B7	0CB	07C	2A0	240	150	165	258	2C8	0C4	334	36B	2D1
05.	1E0	138	39A	0FF	1A7	10C	353	19B	171	038	3BD	000	3A2	1B8	282	2EB
06.	1D4	3D4	20F	23C	0D7	154	012	0DF	3A8	237	09E	155	2E2	189	2F2	136
07.	1B4	381	273	123	052	12C	158	033	2D2	3D3	23B	3B8	2F7	160	341	124
08.	337	1E6	3BE	327	1D3	045	2E4	107	1C2	263	2A4	2CF	244	196	36A	16D
09.	0A1	2C3	004	049	209	3A3	221	361	01E	1B0	05D	319	21B	249	2B2	399
0A.	198	26A	080	1B1	340	28A	33C	316	0FC	37F	1A8	134	17F	3DF	34F	3E5
0B.	2D9	32A	34A	1D1	09D	3FB	0BE	3EA	383	036	3B6	222	22E	2B6	3A6	0FA
0C.	1C1	0B2	113	3E8	129	34C	153	333	07E	01F	01D	213	299	0F8	130	1B9
0D.	182	0A2	1A1	3D5	119	10F	24C	020	097	3F0	280	112	04C	14D	1EB	307
0E.	386	0AE	322	2FE	0C5	3D7	1AF	345	05B	3F5	110	1C8	03C	1C5	35A	0C0
0F.	3F1	15B	338	1CB	0F4	2B4	00F	3A1	242	03D	29D	1B3	003	114	3FA	313
10.	35F	217	261	2C1	15D	28F	390	1C9	1DD	3C7	14F	11D	066	04D	03B	0E9
11.	2BA	2FD	347	191	044	0B8	194	148	256	360	326	257	1AE	396	09B	2CD
12.	1E7	3CD	1FF	269	040	3E7	08A	216	0C9	33B	3D9	1BC	2B1	325	11B	16F
13.	053	22A	186	180	27D	11F	2A9	13E	3E1	0D4	24E	1D2	2FC	3C9	1FB	31A
14.	3DE	1D7	025	372	339	2C7	2ED	25F	0A7	098	2EF	247	0E8	2D3	105	09F
15.	2CC	36D	31F	24B	1D8	241	068	211	2AF	0AA	355	35C	026	2BD	238	0EF
16.	35B	233	05A	1BE	291	368	137	035	298	140	26B	1E4	379	07F	3EB	164
17.	20B	12D	375	1BF	12F	1AA	18B	268	3F4	364	0F7	1CC	0B9	3C5	060	19D
18.	22B	17C	11C	0B1	23A	3B4	05F	2F5	219	224	0E5	042	06F	39D	218	023
19.	1DE	177	190	395	274	359	0E2	2E9	397	0F1	010	099	17D	08E	314	317
1A.	0DC	03F	1AC	1A6	132	152	195	3AD	3E9	3C2	1BB	0F0	0CD	074	178	174
1B.	184	3E0	389	2FB	1A9	0B7	250	27B	06C	13B	0FB	296	297	30F	350	14E
1C.	007	10E	19C	055	351	034	175	103	272	02D	2C0	21C	047	20D	0EA	29B
1D.	13F	1DF	162	376	0BF	1CA	3EC	2B9	3FE	388	133	0A9	33A	304	1FE	059
1E.	13D	0BD	294	02B	127	1E8	275	07B	14C	018	031	1C6	0A3	0EC	27C	087
1F.	38D	3B0	284	1FA	1F5	00A	3E2	02E	228	285	34B	311	075	2F1	1C4	094
20.	3FF	202	27F	2F9	30D	135	33F	301	3D8	2C6	3D2	309	057	073	1F1	289
21.	3B5	3CE	111	0B4	20E	1BA	1F7	24A	394	157	366	336	39B	017	25C	3C4
22.	1EC	2BC	144	1E9	193	16A	33D	344	295	079	027	2D4	38A	17A	292	0AC
23.	0F2	35E	1EF	0BB	106	071	2DA	3F7	084	037	2AB	330	0B0	2DB	07A	22D
24.	00C	149	0AF	290	2E0	122	283	32E	3AE	3C3	1D9	2E5	37B	0BC	265	32D
25.	089	2CB	115	081	18A	255	05C	1A3	287	0D0	276	32C	0C3	30B	226	1C0
26.	2F3	0A5	121	2AA	210	091	208	3EE	230	320	385	28E	21E	3F9	11E	05E
27.	159	281	0C8	37C	0DD	188	04F	26E	33E	2F8	3A0	3B3	3C8	227	1A2	3AA
28.	302	36E	38F	19E	212	13C	24D	0B3	141	3EF	1CE	262	145	362	346	176
29.	1E3	14B	3A9	3DD	093	3F8	070	0D3	1EA	3BA	248	146	201	243	1F6	205
2A.	1CD	20A	2F6	00E	267	26D	2A7	1FC	2D0	0D1	38E	006	30A	3E3	2C5	28B
2B.	2D5	3A7	1D5	3A4	101	2D7	34E	2B5	072	26C	090	1F8	1F9	3AF	1F0	0C2
2C.	2C2	21A	06A	0AB	1EE	109	16B	15E	161	38C	156	271	279	369	342	1D6
2D.	01A	016	352	173	34D	354	181	185	1A5	23F	16C	030	215	1C3	2EA	0CF
2E.	0DE	078	18D	01B	117	393	3F2	39E	37D	1BD	24F	18C	29A	0A4	08D	187
2F.	015	065	0C1	251	00D	348	014	21F	001	008	2CA	321	1B2	1B6	043	147
30.	223	0B6	054	1AB	0FD	373	31E	323	20C	151	10B	288	2DF	041	349	2E3
31.	32F	0ED	277	179	278	3F6	23E	252	077	04A	120	200	308	300	312	04B
32.	1ED	048	30C	183	0D2	39F	3B7	0AD	3FD	204	050	1C7	197	3BF	046	04E
33.	1F3	343	051	1F2	169	266	25A	26F	0F3	2B0	095	17B	31B	0F6	0E6	2DC
34.	225	36C	377	253	058	0E1	021	31C	3D6	1AD	167	2AC	06B	23D	398	032
35.	35D	2FA	00B	391	239	0F5	335	02C	083	143	02A	29E	36F	214	104	14A
36.	0E4	096	19F	2E1	1FD	30E	28D	07D	11A	0EE	0EB	370	358	1DC	163	056
37.	367	03A	2D6	363	229	3DA	08B	1E5	270	2E8	2FF	168	10A	2AE	170	28C
38.	2A3	08C	1CF	076	3D1	32B	2EC	2A5	2C9	2A6	29C	3ED	09C	0CC	2A8	203
39.	0FE	293	29F	2F4	0E7	232	0CE	3AB	13A	011	3DB	220	0D8	1F4	22F	236
3A.	062	1A4	27A	128	329	324	067	365	024	2B8	3E4	0D6	3AC	3A5	172	306
3B.	16E	0DB	3C0	25B	088	2D8	303	380	259	2A1	1E2	0B5	02F	029	356	2BF
3C.	2C4	03E	2BB	3B1	17E	3CC	01C	25E	2B3	069	0A6	15C	0D5	3DC	118	2E6
3D.	2DD	235	18F	371	064	260	0C7	0E0	0C6	1D0	254	0D9	37A	387	3CB	234
3E.	3D0	15F	3B2	08F	15A	013	331	328	06E	25D	0F9	092	166	378	31D	139
3F.	005	09A	12B	061	231	1A0	3CF	3CA	382	192	086	357	22C	12A	3FC	37E

Figure 19: Specification of the modified S-box S_2 .

CHAPTER 5 – BACKDOORED ENCRYPTION ALGORITHM 1

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	1AD	084	1B5	30A	25A	151	174	3F9	113	3B4	35B	291	332	170	021	31E
01.	00E	2FC	023	0B0	376	259	2BC	378	031	050	359	1FF	26C	0D5	214	0BD
02.	1AB	0AB	3AC	036	0E2	2F6	07A	0EA	2CB	0FE	24E	280	057	073	219	3EA
03.	2E2	27C	032	162	285	13C	0B6	1ED	0B3	2F5	2C6	34B	335	093	298	37A
04.	273	17E	30F	2E7	14B	3BC	1CE	039	315	01A	144	1C4	20A	3A9	362	10D
05.	235	1D9	2F9	0A4	052	0E3	17F	061	02C	140	0E1	156	10E	250	288	1BE
06.	07C	2B8	05D	242	192	0A8	3B0	0DB	129	2AF	063	3AF	3D1	0C8	0A6	029
07.	2B9	3B8	092	078	2A2	06E	2CF	3CF	0EF	0E7	019	1F1	07E	1BB	2C7	251
08.	36A	2CA	076	216	2E5	0E6	1DD	2FE	390	277	1D2	394	2C5	022	05A	396
09.	0F4	265	0FD	150	027	111	2EC	29C	3DF	11F	2A1	158	388	1D3	3C8	386
0A.	38B	279	064	1A4	028	34F	1D5	352	2C8	257	3C4	355	0B7	322	2C1	317
0B.	1DF	1A9	137	3DC	015	096	2AA	2A4	3F6	1A3	3DA	086	2E8	343	36F	11A
0C.	0A5	38D	328	348	292	308	3F4	059	31C	1AC	1E4	3BF	1C2	36D	1D8	0ED
0D.	191	3D3	3D4	046	0E8	373	034	23C	102	3C5	11E	393	00B	2D7	2DA	00F
0E.	209	230	19D	184	1B8	339	360	240	011	305	17A	324	344	045	3F0	0F3
0F.	1C6	0B4	08D	18E	035	0C9	345	0D3	37D	3CA	284	3EF	00D	197	36B	06B
10.	08B	10B	18A	218	3DE	32A	2CC	0AE	254	3FC	066	246	24D	232	0A2	145
11.	2DB	199	37F	1E1	392	3F3	1C8	1CD	136	2D0	325	27B	068	1F5	077	22D
12.	1E2	2F4	0B2	2E9	3CC	296	2EA	116	30D	276	02D	11B	09C	25E	157	195
13.	3A1	3F2	3D7	130	258	227	0D4	26B	1FB	1EA	379	329	179	2D5	0C4	09F
14.	39E	09E	1FD	15B	126	2B3	15E	012	21A	372	356	154	042	017	217	19B
15.	1F8	261	3ED	14A	22F	110	037	1B7	079	201	3C9	0EE	2B6	107	3CB	302
16.	19E	21D	1E5	205	25F	3BD	196	198	337	069	32E	0DF	3EE	272	0BC	3C2
17.	01D	37B	3C0	0A3	22E	123	2A9	0DE	2A7	2FF	3A5	05B	38F	047	1B4	350
18.	1EE	0A1	29D	1FC	024	29B	3A3	115	3BE	215	09A	37E	2A0	0C2	377	0B1
19.	149	04A	0E9	365	3D6	2E3	200	35A	17B	0D7	134	3D0	36E	336	334	1F6
1A.	1DC	3B2	2B1	2AB	3F1	1FA	067	06C	020	211	233	28D	0DA	34C	20C	1A8
1B.	28F	389	349	3F5	2FB	1CF	383	387	0CB	08F	0C0	135	3A7	2B0	346	30E
1C.	163	33C	32F	1EF	0FA	125	244	226	1C1	35E	1A2	252	1E9	3A0	146	3D8
1D.	148	353	0FF	37C	09D	2C0	268	048	117	1E3	2A6	003	323	0AD	1D7	313
1E.	072	18D	297	39A	0C7	12D	016	222	056	27A	287	095	366	293	3E0	354
1F.	369	299	190	10F	25B	183	080	1B6	361	3AA	3E1	318	2BA	15C	0D8	1DB
20.	342	2EE	1AE	04F	1A7	2CD	2F8	03A	01F	0BA	188	090	2B7	382	16F	0C5
21.	1B0	2D2	1CC	3B9	267	153	24B	1E7	0D2	0FC	33B	0F7	3BB	25C	1C7	0D9
22.	00C	271	0AA	1C5	357	1E8	01E	3FE	081	245	314	294	164	13F	212	340
23.	141	1B9	2C3	02E	087	0E4	13E	26A	171	249	22B	206	138	001	0A0	23F
24.	02B	2BB	06F	05E	275	20E	3B3	12A	28A	100	2AC	22A	263	0F9	1C0	21B
25.	203	303	35C	295	088	008	3E5	0DD	307	105	121	185	0A7	3EC	11C	347
26.	094	39D	1B2	02A	3B1	204	114	312	167	131	304	290	231	3E7	2D3	3D2
27.	2C2	32C	3E6	2F1	009	10C	327	2F2	2D1	1BA	2FD	35D	253	2EF	282	3D9
28.	338	14F	1B1	28B	330	2F0	18C	175	12E	169	2D9	223	2F3	255	0C3	13D
29.	398	15F	16D	2DC	2BD	0FB	3FA	2ED	147	161	01B	04B	17D	28C	058	3C1
2A.	1A6	21F	1DA	27F	124	2BF	39C	005	054	35F	143	3CE	19A	043	12F	104
2B.	0CF	286	18B	243	006	106	333	152	1BF	3FF	3B7	1EC	30B	098	08E	1D1
2C.	089	3CD	1F0	210	2EB	309	2F7	13B	20D	3AD	02F	0EC	11D	1BD	3A8	38E
2D.	311	1E6	3FB	0AF	2E1	12B	220	03D	0F1	2FA	208	16C	28E	181	33A	119
2E.	033	10A	0CA	2A5	010	31F	3BA	1F3	3B5	2DD	193	2AD	283	085	00A	32B
2F.	0F5	3AB	1D0	2E6	0EB	2DF	2B2	06A	065	38C	18F	364	33E	0B8	2CE	1F2
30.	289	142	266	132	3E2	24C	101	24A	39B	09B	097	1A0	229	375	320	062
31.	33D	118	3EB	03C	15A	281	1A1	207	3C7	331	319	082	127	34E	07B	239
32.	23A	300	0B5	01C	2B5	1E0	39F	180	321	133	26F	371	1B3	363	26D	23E
33.	0C6	165	0F6	19C	070	0E0	367	1DE	247	213	053	109	2D6	2B4	3DB	29E
34.	30C	1CB	0E5	1F7	15D	2E0	013	236	2A3	228	0BB	14D	018	278	155	1C9
35.	178	0CD	370	07D	3A6	23B	049	2D4	397	0BF	1BC	21E	399	3F8	0AC	004
36.	34A	055	04D	14C	33F	13A	301	05C	3A2	112	2DE	075	3F7	391	040	326
37.	2D8	000	0DC	0D1	041	14E	20B	1A5	166	168	051	22C	31B	0CC	16E	172
38.	1CA	2C4	3C3	0A9	03F	173	27E	08A	25D	1F9	014	17C	044	16B	380	176
39.	31D	3E9	108	139	2C9	04C	187	071	29F	381	316	038	0CE	06D	34D	1AA
3A.	122	2A8	1C3	04E	368	351	202	38A	225	189	194	306	026	0F0	248	08C
3B.	05F	19F	2BE	270	060	083	186	3DD	2AE	0C1	23D	160	241	0F8	1EB	385
3C.	374	177	3E4	358	1FE	099	120	1D4	3A4	310	030	1AF	1F4	0BE	074	16A
3D.	274	1D6	21C	3FD	3C6	238	234	262	3D5	31A	395	27D	3E8	128	002	29A
3E.	32D	0D0	341	26E	0B9	224	237	0F2	2E4	12C	103	025	20F	260	3AE	269
3F.	07F	03B	03E	007	182	159	091	3B6	3E3	384	264	0D6	36C	256	221	24F

Figure 20: Specification of the secret S-box S_3 .

	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
00.	200	084	1B5	30A	25A	151	174	3F9	113	3B4	35B	291	332	170	021	31E
01.	00E	2FC	023	0B0	3A9	259	2BC	378	031	050	0D0	1FF	26C	0D5	214	23E
02.	1AB	0AB	3AC	036	0E2	2F6	07A	0EA	2CB	0FE	24E	280	138	073	219	3EA
03.	2E2	27C	032	162	285	13C	0B6	1ED	0B3	2F5	2C6	34B	335	1EF	26E	37A
04.	273	17E	30F	2E7	14B	3BC	1CE	039	315	01A	144	1C4	20A	17B	362	10D
05.	235	1D9	2F9	0A4	052	0E3	0BD	061	02C	140	0E1	156	10E	250	288	1BE
06.	07C	2B8	05D	242	192	0A8	3B0	0DB	129	2AF	063	3AF	3D1	0C8	0A6	029
07.	2B9	3B8	0D2	078	2A2	06E	2CF	3CF	0EF	0E7	019	1F1	07E	1BB	2C7	251
08.	36A	2CA	076	216	2E5	0E6	1DD	2FE	390	277	1D2	394	2C5	022	05A	396
09.	0F4	265	0FD	150	057	111	2EC	29C	3DF	11F	13A	158	388	1D3	3C8	386
0A.	38B	279	064	1A4	028	22F	1D5	352	2C8	257	3C4	355	104	322	2C1	382
0B.	1DF	1A9	137	3DC	015	096	2AA	2A4	3F6	1A3	3DA	086	2E8	343	233	11A
0C.	0A5	38D	328	348	292	132	3F4	059	31C	1AC	1C6	3BF	1C2	36D	1D8	0ED
0D.	191	3D3	3D4	3DE	0E8	373	034	23C	224	3C5	11E	393	00B	308	2DA	00F
0E.	209	230	19D	184	1B8	339	360	2D7	011	305	17A	324	344	128	3F0	0F3
0F.	317	0B4	08D	18E	035	0C9	345	0D3	37D	3CA	284	3EF	00D	197	36B	06B
10.	08B	10B	18A	218	046	32A	2CC	0AE	254	3FC	066	246	24D	232	0A2	145
11.	2DB	199	37F	1E1	392	3F3	1C8	1CD	136	2D0	325	27B	068	1F5	077	22D
12.	12F	2F4	0B2	2E9	3CC	296	2EA	116	30D	276	02D	266	09C	25E	157	195
13.	3A1	3F2	3D7	130	258	227	0D4	26B	027	1EA	379	329	179	2D5	0C4	09F
14.	39E	09E	1FD	15B	126	2B3	15E	012	21A	372	356	154	042	017	217	19B
15.	1F8	261	3ED	14A	1FB	110	037	1B7	079	045	3C9	0EE	2B6	107	3CB	302
16.	19E	21D	1E5	205	25F	3BD	196	198	337	069	32E	0DF	3EE	201	0BC	3C2
17.	01D	37B	3C0	0A3	22E	123	2A9	0DE	2A7	2FF	3A5	05B	38F	047	1B4	350
18.	0CB	0A1	29D	1FC	024	29B	3A3	2A1	3BE	215	09A	37E	2A0	0C2	377	0B1
19.	149	33B	323	365	3D6	2E3	082	35A	38C	0D7	134	3D0	36E	336	334	1F6
1A.	1DC	3B2	2B1	213	3F1	1FA	380	06C	020	211	033	28D	0DA	34C	20C	1A8
1B.	28F	369	349	3F5	2FB	1CF	383	387	35E	08F	29A	135	3A7	2B0	346	30E
1C.	163	33C	32F	093	0FA	125	244	226	1C1	1EE	1A2	252	1E9	3A0	146	3D8
1D.	148	353	0FF	37C	09D	2C0	268	048	117	1E3	2A6	003	11B	0AD	1D7	313
1E.	072	18D	297	39A	0C7	12D	016	222	056	1CB	287	095	366	293	3E0	354
1F.	13E	299	190	10F	25B	183	080	1B6	361	3AA	3E1	318	2BA	15C	0D8	1DB
20.	342	2EE	1AE	04F	1A7	2CD	2F8	03A	06A	0BA	188	090	2B7	1E4	16F	0C5
21.	1B0	2D2	1CC	3B9	267	153	24B	1E7	20B	0FC	2E6	0F7	3BB	376	1C7	0D9
22.	00C	271	0AA	1C5	357	1E8	01E	3FE	081	245	314	294	164	13F	212	340
23.	141	1B9	120	02E	34F	0E4	092	26A	171	249	22B	206	0C0	001	0A0	23F
24.	02B	2BB	06F	05E	275	20E	3B3	12A	28A	100	2AC	22A	263	0F9	1C0	21B
25.	203	303	35C	295	088	008	3E5	0DD	307	105	121	185	0A7	3EC	11C	347
26.	094	39D	1B2	02A	3B1	204	114	312	167	131	304	290	231	3E7	2D3	3D2
27.	2C2	32C	3E6	04A	009	10C	327	1BD	2D1	1BA	2FD	35D	253	2EF	282	3D9
28.	338	14F	1B1	28B	330	2F0	18C	175	12E	169	2D9	223	2F3	255	0C3	13D
29.	398	15F	16D	2DC	2BD	0FB	3FA	2ED	147	161	01B	04B	17D	28C	058	3C1
2A.	1A6	21F	1DA	0E9	124	2BF	39C	005	054	35F	143	3CE	19A	043	36F	1F3
2B.	0CF	286	18B	243	006	106	333	152	1BF	3FF	3B7	1EC	30B	098	08E	1D1
2C.	089	3CD	1F0	210	2EB	309	2F7	13B	20D	3AD	02F	0EC	11D	06D	3A8	38E
2D.	311	1E6	3FB	0AF	2E1	12B	220	03D	0F1	2FA	208	16C	28E	181	33A	119
2E.	109	10A	0CA	2A5	010	31F	3BA	0B7	3B5	2DD	193	2AD	283	085	00A	32B
2F.	2A8	3AB	1D0	2F1	0EB	2DF	298	1DE	065	17C	18F	364	33E	0B8	2CE	1F2
30.	289	142	2B2	0D6	3E2	24C	101	24A	39B	09B	097	1A0	229	375	320	062
31.	33D	118	3EB	03C	15A	281	1A1	207	3C7	331	319	27A	127	34E	07B	239
32.	23A	300	0B5	01C	2B5	1E0	39F	180	321	133	26F	371	1B3	363	26D	0F5
33.	0C6	165	0F6	19C	070	0E0	367	002	247	389	053	27F	2D6	2B4	3DB	29E
34.	30C	1AD	0E5	22C	15D	2E0	013	236	2A3	228	0BB	14D	018	278	155	1C9
35.	178	0CD	370	07D	3A6	23B	049	2D4	397	0BF	1BC	21E	399	3F8	0AC	004
36.	34A	055	04D	14C	33F	1F7	301	05C	3A2	112	2DE	075	3F7	391	040	326
37.	2D8	000	0DC	0D1	041	14E	067	160	166	168	051	0A9	31B	0CC	16E	172
38.	1CA	2C4	3C3	1E2	03F	173	27E	08A	25D	1F9	014	17F	044	16B	2F2	176
39.	31D	3E9	108	139	2C9	04C	187	071	29F	381	316	038	0CE	2C3	34D	1AA
3A.	122	225	1C3	04E	368	351	202	38A	102	189	194	306	026	0F0	248	08C
3B.	05F	19F	2BE	270	060	083	186	3DD	2AE	0C1	23D	272	241	0F8	1EB	01F
3C.	374	177	3E4	358	1FE	099	2AB	1D4	3A4	310	030	1AF	1F4	0BE	074	16A
3D.	274	1D6	21C	3FD	3C6	238	234	262	3D5	31A	395	27D	3E8	240	1A5	087
3E.	32D	359	341	25C	0B9	115	237	0F2	2E4	12C	103	025	20F	260	3AE	269
3F.	07F	03B	03E	007	182	159	091	3B6	3E3	384	264	385	36C	256	221	24F

Figure 21: Specification of the modified S-box S_3 .

Bibliography

- [1] Kashif Ali and Howard M Heys. An Algorithm to Analyze Block Cipher Resistance to Linear and Differential Cryptanalysis. In *Proceedings of Newfoundland Electrical and Computer Engineering Conference (NECEC 2006)*, 2006.
- [2] Vesela Angelova and Yuri Borissov. Plaintext recovery in des-like cryptosystems based on s-boxes with embedded parity check. *Serdica Journal of Computing*, 7(3):257p–270p, 2013.
- [3] Kazumaro Aoki, Kunio Kobayashi, and Shiho Moriai. Best differential characteristic search of FEAL. In *Fast Software Encryption*, pages 41–53. Springer, 1997.
- [4] Riccardo Aragona, Marco Calderini, Antonio Tortora, and Maria Tota. On the primitivity of present and other lightweight ciphers. *arXiv preprint arXiv:1611.01346*, 2016.
- [5] Riccardo Aragona, Andrea Caranti, Francesca Dalla Volta, and Massimiliano Sala. On the group generated by the round functions of translation based ciphers over arbitrary finite fields. *Finite Fields and Their Applications*, 25:293–305, 2014.
- [6] John Bamberg. *Permutation Group Theory*. RMIT Summer Course notes, 2006.
- [7] Arnaud Bannier and Nicolas Bodin. A new drawing for simple venn diagrams based on algebraic construction. *Journal of Computational Geometry*, 8(1):153–173, 2017.
- [8] Arnaud Bannier, Nicolas Bodin, and Eric Filiol. Automatic Search for a Maximum Probability Differential Characteristic in a Substitution-Permutation Network. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 5165–5174. IEEE, 2015. Best Paper Awards.
- [9] Arnaud Bannier, Nicolas Bodin, and Eric Filiol. Partition-based trapdoor ciphers. Cryptology ePrint Archive, Report 2016/493, 2016. <http://eprint.iacr.org/2016/493>.
- [10] Arnaud Bannier and Eric Filiol. Mathematical Backdoors in Symmetric Encryption Systems – Proposal for a Backdoored AES-like Block Cipher. In *1st*

BIBLIOGRAPHY

- International Workshop on Formal methods for Security Engineering (ForSE)*, February 2017.
- [11] Arnaud Bannier and Eric Filiol. Operational Cryptanalysis Based on Backdoors Exploitation in an AES-like Cipher. In *RusCrypto'17*, March 2017.
 - [12] Arnaud Bannier and Eric Filiol. *Partition-based Trapdoor Ciphers*. InTech editions, 2017. ISBN:978-953-51-3386-5 (Print), ISBN:978-953-51-3385-8 (Online).
 - [13] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.
 - [14] Eli Biham and Adi Shamir. *Differential cryptanalysis of the data encryption standard*, volume 28. Springer, 1993.
 - [15] Alex Biryukov and Léo Paul Perrin. State of the art in lightweight symmetric cryptography. 2017.
 - [16] Céline Blondeau, Roberto Civino, and Massimiliano Sala. Differential attacks: Using alternative operations. Cryptology ePrint Archive, Report 2017/610, 2017. <http://eprint.iacr.org/2017/610>.
 - [17] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsøe. PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems-CHES 2007*, pages 450–466. Springer, 2007.
 - [18] KA Browning, JF Dillon, MT McQuistan, and AJ Wolfe. An apn permutation in dimension six. *Finite Fields: theory and applications*, 518:33–42, 2010.
 - [19] Carlo Brunetta, Marco Calderini, and Massimiliano Sala. Algorithms and bounds for hidden sums in cryptographic trapdoors. *arXiv preprint arXiv:1702.08384*, 2017.
 - [20] Lilya Budaghyan, Claude Carlet, and Alexander Pott. New classes of almost bent and almost perfect nonlinear polynomials. *IEEE Transactions on Information Theory*, 52(3):1141–1152, 2006.
 - [21] 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.
 - [22] Marco Calderini. *On Boolean functions, symmetric cryptography and algebraic coding theory*. PhD thesis, University of Trento, 2015.
 - [23] Marco Calderini. A note on some algebraic trapdoors for block ciphers. *arXiv preprint arXiv:1705.08151*, 2017.

- [24] Marco Calderini and Massimiliano Sala. On differential uniformity of maps that may hide an algebraic trapdoor. In *International Conference on Algebraic Informatics*, pages 70–78. Springer, 2015.
- [25] Marco Calderini and Massimiliano Sala. Elementary abelian regular subgroups as hidden sums for cryptographic trapdoors. *arXiv preprint arXiv:1702.00581*, 2017.
- [26] Peter J Cameron. *Permutation groups*, volume 45. Cambridge University Press, 1999.
- [27] Anne Canteaut. *Lecture Notes on Cryptographic Boolean Functions*. Inria, 2016. <https://www.rocq.inria.fr/secret/Anne.Canteaut/poly.pdf>.
- [28] Anne Canteaut, Pascale Charpin, and Hans Dobbertin. A new characterization of almost bent functions. In *Fast Software Encryption*, volume 99, pages 186–200. Springer-Verlag, 1999.
- [29] Anne Canteaut, Pascale Charpin, and Hans Dobbertin. Binary m-sequences with three-valued crosscorrelation: a proof of welch’s conjecture. *IEEE Transactions on Information Theory*, 46(1):4–8, 2000.
- [30] A Caranti, Francesca Dalla Volta, and Massimiliano Sala. On some block ciphers and imprimitive groups. *Applicable algebra in engineering, communication and computing*, 20(5-6):339–350, 2009.
- [31] A Caranti, F Dalla Volta, Massimiliano Sala, and Francesca Villani. Imprimitive permutations groups generated by the round functions of key-alternating block ciphers and truncated differential cryptanalysis. *arXiv preprint math/0606022*, 2006.
- [32] Andrea Caranti, Francesca Dalla Volta, and Massimiliano Sala. An application of the o’nan-scott theorem to the group generated by the round functions of an aes-like cipher. *Designs, Codes and Cryptography*, 52(3):293–301, 2009.
- [33] Claude Carlet. Vectorial Boolean functions for cryptography. *Boolean models and methods in mathematics, computer science, and engineering*, 134:398–469, 2010.
- [34] Claude Carlet, Pascale Charpin, and Victor Zinoviev. Codes, bent functions and permutations suitable for des-like cryptosystems. *Designs, Codes and Cryptography*, 15(2):125–156, 1998.
- [35] Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology—EUROCRYPT’94*, pages 356–365. Springer, 1995.
- [36] Huiju Cheng, Howard M Heys, and Cheng Wang. Puffin: A novel compact block cipher targeted to embedded digital systems. In *Digital System Design Architectures, Methods and Tools, 2008. DSD’08. 11th EUROMICRO Conference on*, pages 383–390. IEEE, 2008.

BIBLIOGRAPHY

- [37] Baudoin Collard, F-X Standaert, and J-J Quisquater. Improved and multiple linear cryptanalysis of reduced round Serpent. In *Information Security and Cryptology*, pages 51–65. Springer, 2008.
- [38] Don Coppersmith and Edna Grossman. Generators for certain alternating groups with applications to cryptography. *SIAM Journal on Applied Mathematics*, 29(4):624–627, 1975.
- [39] Joan Daemen and Vincent Rijmen. *The design of Rijndael*. Springer Verlag, 2002.
- [40] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *Journal of Mathematical Cryptology JMC*, 1(3):221–242, 2007.
- [41] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.
- [42] John D Dixon and Brian Mortimer. *Permutation groups*, volume 163. Springer Science & Business Media, 1996.
- [43] Hans Dobbertin. Almost perfect nonlinear power functions on $\text{gf}(2^n)$: the niho case. *Information and Computation*, 151(1-2):57–72, 1999.
- [44] Hans Dobbertin. Almost perfect nonlinear power functions on $\text{gf}(2^{\sup n})$: the welch case. *IEEE Transactions on Information Theory*, 45(4):1271–1275, 1999.
- [45] Robert Gold. Maximal recursive sequences with 3-valued recursive cross-correlation functions. *IEEE transactions on Information Theory*, 14(1):154–156, 1968.
- [46] Jay Goldman and Gian-Carlo Rota. The number of subspaces of a vector space. Technical report, DTIC Document, 1969.
- [47] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace trail cryptanalysis and its applications to aes - extended version. <http://eprint.iacr.org/2016/592>.
- [48] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace trail cryptanalysis and its applications to aes. *IACR Transactions on Symmetric Cryptology*, 2016(2):192–225, 2017.
- [49] Jian Guo, Jérémy Jean, Ivica Nikolic, Kexin Qiao, Yu Sasaki, and Siang Meng Sim. Invariant subspace attack against midori64 and the resistance criteria for s-box designs. *IACR Transactions on Symmetric Cryptology*, 2016(1):33–56, 2016.
- [50] Carlo Harpes. *Cryptanalysis of iterated block ciphers*. PhD thesis, Diss. Techn. Wiss. ETH Zürich, Nr. 11625, 1996. Ref.: JL Massey; Korref.: U. Maurer, 1996.

- [51] Carlo Harpes, Gerhard G Kramer, and James L Massey. A generalization of linear cryptanalysis and the applicability of matsui's piling-up lemma. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–38. Springer, 1995.
- [52] Carlo Harpes and James L Massey. Partitioning cryptanalysis. In *International Workshop on Fast Software Encryption*, pages 13–27. Springer, 1997.
- [53] Henk DL Hollmann and Qing Xiang. A proof of the welch and niho conjectures on cross-correlations of binary m-sequences. *Finite Fields and Their Applications*, 7(2):253–286, 2001.
- [54] G Hornauer, W Stephan, and Ralph Wernsdorf. Markov ciphers and alternating groups. *Lecture Notes in Computer Science*, 765:453–460, 1994.
- [55] Alexander Hulpke. Notes on computational group theory. *Department of Mathematics. Colorado State University*, 2010.
- [56] Burton S Kaliski, Ronald L Rivest, and Alan T Sherman. Is the data encryption standard a group? (results of cycling experiments on des). *Journal of Cryptology*, 1(1):3–36, 1988.
- [57] Masayuki Kanda, Youichi Takashima, Tsutomu Matsumoto, Kazumaro Aoki, and Kazuo Ohta. A strategy for constructing fast round functions with practical security against differential and linear cryptanalysis. In *Selected Areas in Cryptography*, pages 264–279. Springer, 1999.
- [58] Tadao Kasami. The weight enumerators for several classes of subcodes of the 2nd order binary reed-muller codes. *Information and Control*, 18(4):369–394, 1971.
- [59] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, pages 5–83, 1883.
- [60] Lars Knudsen. Truncated and higher order differentials. In *Fast Software Encryption*, pages 196–211. Springer, 1995.
- [61] Lars Knudsen, Gregor Leander, Axel Poschmann, and Matthew JB Robshaw. Printcipher: a block cipher for ic-printing. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 16–32. Springer, 2010.
- [62] Lars R Knudsen. *Block Ciphers – Analysis, Design and Applications*. PhD thesis, Aarhus University, Denmark, 1994.
- [63] Lars R Knudsen. Block ciphers-a survey. *Lecture notes in computer science*, 1528:18–48, 1998.
- [64] Lars R Knudsen and Matthew JB Robshaw. *The block cipher companion*. Springer, 2011.

BIBLIOGRAPHY

- [65] Marc Krasner and Léo Kaloujnine. Produit complet des groupes de permutations et problème d’extension de groupes. III. *Acta Sci. Math.(Szeged)*, 14:69–82, 1951.
- [66] Gilles Lachaud and Jacques Wolfmann. The weights of the orthogonals of the extended quadratic binary goppa codes. *IEEE transactions on information theory*, 36(3):686–692, 1990.
- [67] Xuejia Lai, James L Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology—EUROCRYPT’91*, pages 17–38. Springer, 1991.
- [68] Serge Lang. Linear algebra. 3rd corr. printing 1993 (undergraduate texts in mathematics), 1987.
- [69] Gregor Leander. Small Scale Variants Of The Block Cipher PRESENT. *IACR Cryptology ePrint Archive*, 2010:143, 2010.
- [70] Gregor Leander, Mohamed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A cryptanalysis of printcipher: the invariant subspace attack. *Advances in Cryptology—CRYPTO 2011*, pages 206–221, 2011.
- [71] Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of robin, iscream and zorro. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 254–283. Springer, 2015.
- [72] Gregor Leander and Axel Poschmann. On the classification of 4 bit s-boxes. In *Arithmetic of Finite Fields*, pages 159–176. Springer, 2007.
- [73] James L Massey. Cryptography: Fundamentals and applications. In *Copies of transparencies, Advanced Technology Seminars*, volume 109, page 119, 1993.
- [74] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology—EUROCRYPT’93*, pages 386–397. Springer, 1994.
- [75] Mitsuru Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology—Crypto’94*, pages 1–11. Springer, 1994.
- [76] Mitsuru Matsui. On correlation between the order of S-boxes and the strength of DES. In *Advances in Cryptology—EUROCRYPT’94*, pages 366–375. Springer, 1995.
- [77] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.
- [78] Sean Murphy, Kenneth Paterson, and Peter Wild. A weak cipher that generates the symmetric group. *Journal of Cryptology*, 7(1):61–65, 1994.
- [79] Kaisa Nyberg. Perfect nonlinear s-boxes. In *Advances in Cryptology—EUROCRYPT’91*, pages 378–386. Springer, 1991.

- [80] Kaisa Nyberg. Differentially uniform mappings for cryptography. In *Advances in cryptology—Eurocrypt’93*, pages 55–64. Springer, 1993.
- [81] Kaisa Nyberg. On the construction of highly nonlinear permutations. In *Advances in Cryptology—EUROCRYPT’92*, pages 92–98. Springer, 1993.
- [82] Kaisa Nyberg and Lars R Knudsen. Provable security against differential cryptanalysis. In *Crypto*, volume 92, pages 566–574. Springer, 1992.
- [83] National Institute of Standards and Technology. *Data encryption standard*. Federal Information Processing Standard (FIPS), Publication 46, 1977.
- [84] National Institute of Standards and Technology. *DES modes of operation*. Federal Information Processing Standard (FIPS), Publication 81, 1980.
- [85] National Institute of Standards and Technology. *Advanced encryption standard*. Federal Information Processing Standard (FIPS), Publication 197, 2001.
- [86] National Institute of Standards and Technology. *Recommendation for block cipher modes of operation*. Special Publication 800-38A, 2001.
- [87] Kazuo Ohta, Shiho Moriai, and Kazumaro Aoki. Improving the search algorithm for the best linear expression. In *Advances in Cryptology—CRYPTO’95*, pages 157–170. Springer, 1995.
- [88] Kenneth G Paterson. Imprimitive permutation groups and trapdoors in iterated block ciphers. In *Fast Software Encryption*, pages 201–214. Springer, 1999.
- [89] Vincent Rijmen and Bart Preneel. A family of trapdoor ciphers. In *Fast Software Encryption*, pages 139–148. Springer, 1997.
- [90] Joseph Rotman. *An introduction to the theory of groups*, volume 148 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, fourth edition, 1995.
- [91] Claude E Shannon. Communication theory of secrecy systems. *Bell Labs Technical Journal*, 28(4):656–715, 1949.
- [92] Vladimir Michilovich Sidelnikov. On the mutual correlation of sequences. In *Soviet Math. Dokl.*, volume 12, pages 197–201, 1971.
- [93] Neil JA Sloane et al. The On-line Encyclopedia of Integer Sequences, 2017. <http://oeis.org>.
- [94] Rüdiger Sparr and Ralph Wernsdorf. Group theoretic properties of rijndael-like ciphers. *Discrete Applied Mathematics*, 156(16):3139–3149, 2008.
- [95] Francois-Xavier Standaert, Gilles Piret, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. ICEBERG: An involutinal cipher efficient for block encryption in reconfigurable hardware. In *Fast Software Encryption*, pages 279–298. Springer, 2004.

BIBLIOGRAPHY

- [96] Anne Tardy-Corffdir and Henri Gilbert. A known plaintext attack of feal-4 and feal-6. In *Advances in Cryptology—CRYPTO'91*, pages 172–182. Springer, 1992.
- [97] Ralph Wernsdorf. The one-round functions of the des generate the alternating group. In *Eurocrypt*, pages 99–112. Springer, 1992.
- [98] Ralph Wernsdorf. The round functions of rijndael generate the alternating group. In *FSE*, pages 143–148. Springer, 2002.
- [99] Helmut Wielandt. *Finite Permutation Groups*. Academic Press, 1964.
- [100] Hongjun Wu, Feng Bao, Robert H Deng, and Qin-Zhong Ye. Cryptanalysis of rijmen-preneel trapdoor ciphers. In *Advances in Cryptology—Asiacrypt'98*, pages 126–132. Springer, 1998.

Analyse combinatoire des chiffrements par blocs avec trappes

Résumé. Les trappes jouent un double rôle dans la cryptographie moderne. Même si elles sont essentielles en cryptographie asymétrique, leur rôle est tout autre lorsque l'on considère la cryptographie symétrique. Dans ce cas, une trappe désigne une faiblesse mathématique insérée volontairement au cœur du chiffrement, permettant à son concepteur de le casser efficacement. Une telle propriété est alors fortement indésirable. Pour qu'un chiffrement à trappe puisse inspirer confiance, il doit fournir les mêmes preuves de sécurité que tout autre chiffrement. La première partie de cette thèse se concentre sur les analyses de sécurité par rapport aux deux principales cryptanalyses des chiffrements par blocs, à savoir les attaques différentielles et linéaires.

La seconde partie est quant à elle dédiée à l'étude d'une famille de chiffrements à trappes introduite par Paterson et Harpes. Ces chiffrements envoient une partition des messages clairs sur une partition des messages chiffrés indépendamment des clés utilisées. Tout d'abord, nous étudions la structure de tels chiffrements puis obtenons des bornes sur leur sécurité. Nous expliquons ensuite comment les primitives du chiffrement doivent être conçues pour atteindre ces bornes. Enfin, nous présentons BEA-1, un chiffrement à trappe grandeur nature développé à partir de cette théorie. Bien qu'il soit résistant aux cryptanalyses différentielle et linéaire, la connaissance de la trappe permet de retrouver la clé de 120 bits en seulement quelques secondes sur un portable.

Mots clés : cryptographie, trappes, partitions.

Combinatorial Analysis of Block Ciphers With Trapdoors

Abstract. Trapdoors are a two-face key concept in modern cryptography. Even if they are essential in asymmetric cryptography, their role is reversed in symmetric cryptography. In this case, the aim is to insert hidden mathematical weaknesses which enable one who knows them to break the cipher, making the existence of a trapdoor a strongly undesirable property. For a backdoor cipher to be trusted, it must provide the same security proofs than any other cipher. The first part of this thesis focuses on a security analysis with respect to the two main attacks on block ciphers, namely differential and linear cryptanalysis.

The second part is devoted to the study of a family of backdoor ciphers introduced by Paterson and Harpes. These ciphers maps a partition of the plaintexts to a partition of the ciphertext independently of the keys used. First the structure of such ciphers is investigated and bounds of their security are obtained. We then explain how the basic components of a backdoor cipher can be designed to achieve these bounds. Finally we introduce BEA-1, a real-size backdoor cipher based on this theory. This cipher resists differential and linear cryptanalysis whereas the knowledge of the trapdoor enables recovery of the full 120-bit cipher key in just a few second on a laptop computer.

Keywords: Cryptography, Trapdoors, Partitions.