Tools and Algorithms for Cryptanalysis

Patrick Derbez

UNIVERSITÉ DE RENNES 1

HABILITATION À DIRIGER DES RECHERCHES

Tools and Algorithms for Cryptanalysis

Patrick DERBEZ

HDR présentée et soutenue à l'IRISA, le 9 Juin 2022
Diplôme de l'Université Rennes 1
Spécialité: Informatique

Composition du Jury :

Rapporteurs :	Anne CANTEAUT	Directrice de Recherche, INRIA Paris
	Henri GILBERT	Responsable du laboratoire de cryptographie de l'ANSSI
	Willi MEIER	Professeur d'Université, FHNW (Suisse)
Examinateurs :	Sylvain DUQUESNE	Professeur d'Université, Rennes 1
	María NAYA-PLASENCIA	Directrice de Recherche, INRIA Paris

Remerciements

Tout comme pour ma thèse, je tiens à remercier en premier lieu Pierre-Alain Fouque. Après avoir été mon directeur de thèse pendant 3 ans à l'ENS où j'ai pu bénéficier de son appui scientifique et de ses excellents conseils, je suis heureux qu'il soit aujourd'hui le responsable de mon équipe de recherche, mon plus proche collègue, mon co-bureau et aujourd'hui un ami. Je suis vraiment reconnaissant pour tout ce qu'il m'a apporté, pour toutes nos discussions scientifiques ou non, et j'espère sincèrement que notre collaboration ne s'arrêtera pas là.

Je suis très reconnaissant envers les trois rapporteurs de ce manuscrit, Anne Canteaut, Henri Gilbert et Willi Meier, pour avoir examiné mon travail en le peu de temps qui leur était imparti, et ce, malgré des emplois du temps parfois chargés. Notamment je remercie Anne qui en acceptant de participer à ce jury était sûre d'être rapporteur, Henri qui était déjà membre de mon jury de thèse et Willi qui m'a soutenu scientifiquement. J'associe également à ces remerciements Sylvain Duquesne et Maria Naya-Plasencia qui ont tout deux accepté de participer à ce jury.

Je remercie chaleureusement toute l'équipe CryptoLUX de l'Université du Luxembourg pour les 2 années de postdoc que j'ai passé avec eux. Je remercie également tous les membres des équipes EMSEC, SPICY et CAPSULE de l'IRISA. Une mention spéciale pour Gildas Avoine et Mohamed Sabt avec qui j'ai toujours pris plaisir à échanger, et une autre à Stéphanie Delaune avec qui j'ai en plus encadré des étudiants et participé aux évènements de l'école Bourgchevreuil.

Je n'oublie pas mes co-auteurs que je remercie également, ainsi que mes 4 doctorants: Baptiste, Victor, Arthur et Hoa. Je tiens aussi à remercier tous les collègues qui font que les conférences et les workshops sont toujours des moments agréables. Une dédicace spéciale à Pierre-Alain, Jérémy, Christina, Maria, Thomas $\times 2$, Virginie, Gregor, Gaëtan, Medhi, Yu et Siwei pour tous ces bons souvenirs!

Enfin je souhaite remercier toute ma famille et tous mes amis pour tous ces bons moments passés ensemble, et en particulier ma femme Natacha et mes deux enfants Loup et Alice qui doivent régulièrement composer avec les deadlines inhérentes à la recherche académique et ma fâcheuse manie de toujours tout faire au dernier moment.

Table of Contents

Chapter 1

General Introduction

The results presented in this manuscript are a logical continuation of the research initiated during my PhD. I have continued to improve cryptanalysis techniques and to develop tools and algorithms dedicated to cryptography, leading to several new attacks against block ciphers.

One of my major interests is to develop tools and algorithms for cryptanalysis. This implies to well understand cryptanalysis techniques and most often to improve and generalize them as well. Because the search space is typically huge, a large part of my work is about exploiting the structure of primitives and/or techniques to reduce it, in order to make the search practical. So far I contributed to many tools including a very versatile one about meet-in-the-middle and impossible differential attacks, an ad-hoc tool to search for integral distinguishers based on division property (handling for the first time division tables of Super-Sboxes), a new MILP/CP/ad-hoc approach to find the best boomerang distinguishers on `SKINNY` (Best Paper Award) and a dynamic programming-based algorithm to exhaust truncated differential characteristics on `SKINNY` as well. They all led to interesting new results as for instance longer integral distinguishers on `Midori-64`, `SKINNY-64` and `HIGHT`, or boomerang distinguishers on `SKINNY` holding with much higher probability than previously known ones (up to 2^{30} times higher).

I also studied algorithms dedicated to the conception of symmetric primitives, aiming at generating optimal components regarding various parameters. For instance in [Der+18a], our goal was to find the best permutation which could be used as key schedule for `AES` in order to obtain an optimal resistance against differential attacks. But my main result in this area is about optimal permutations for Generalized Feistel Networks (GFN). Indeed, in [Der+19] and together with Pierre-Alain Fouque and my two PhD students, Baptiste Lambin and Victor Mollimard, we solved a 10-year open problem regarding the optimal diffusion rounds for 16-block GFN with the help of an original algorithm.

Finally, I am also particularly interested by *practical* attacks against cryptographic

primitives. I participated to several challenges organized by designers of the block ciphers PRINCE and SKINNY and won some of them, especially the ones related to breaking as many rounds as possible using only a limited amount of plaintext/ciphertext pairs. I also studied some of the proposals of white-box implementation of AES and proposed a generic (and practical) attack against Baek et al. scheme. Very recently I also participated to the first publicly available cryptanalysis of both GEA-1 and GEA-2 stream ciphers, used to encrypt GPRS traffic in 2G technology and still present on modern phones. Following those results, the organisation responsible for telecommunications standards (ETSI) stated that new smartphones should not support those stream ciphers anymore.

Chapter 2

Computer-aided Design of Optimal Components

Contents

When designing block ciphers, we need to make decisions on which specific components to use (e.g. S-boxes, linear layer etc.). These decisions are made by taking into account the security of the resulting block cipher, but also the underlying cost in term of performances. In this chapter, I present two of my works [Der+18a; Der+19] aiming at finding optimal components with respect to a given criterion.

2.1 Variants of the AES Key Schedule for Better Truncated Differential Bounds

An interesting problem I worked on was the design of an alternative to the original key schedule of AES leading to a better resistance against differential cryptanalysis. First introduced in 1990 by Biham and Shamir [BS90], differential cryptanalysis is one of the main tools to analyze and attack symmetric primitives. The main idea is to introduce some differences in the plaintext, and see how these differences propagate through the different steps of the algorithm, independently from the key. For example, given an encryption

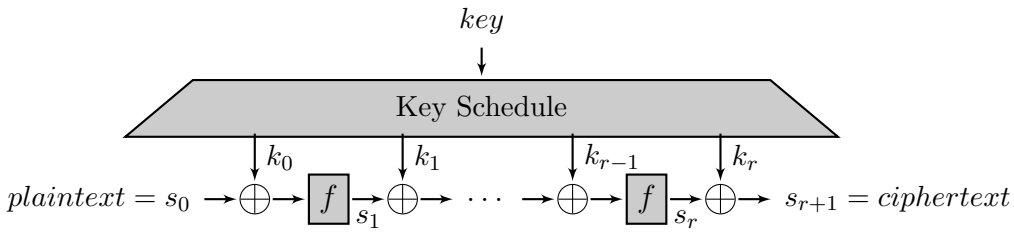

Figure 2.1 – Generic iterated cipher construction

function $\mathcal{E}(p, k)$ encrypting the plaintext $p \in \mathbb{F}_2^{n_b}$ using a key $k \in \mathbb{F}_2^{n_k}$, if one is able to prove that there exists a pair of differences $\Delta_{in}, \Delta_{out} \in \mathbb{F}_2^{n_b}$ such that $\mathcal{E}(p \oplus \Delta_{in}, k) = \mathcal{E}(p, k) \oplus \Delta_{out}$ for all keys, then it gives a strong distinguisher for the encryption function $\mathcal{E}$. Because of the non-linearity of $\mathcal{E}$, such a differential relation could only hold with a certain probability and a lot of work has been put into designing algorithms that search for the best possible differential distinguishers of a given cipher. For instance, Matsui designed two such algorithms in [Mat94]. Most of modern ciphers are now built as *iterated ciphers*, where a round function f is built and repeated several times, XOR-ing a round key between each application of f, see Figure 2.1. Thus, to search for such a pair $(\Delta_{in}, \Delta_{out})$, one often studies the propagation of the input difference through each round of the cipher, leading to a *differential characteristic* consisting of all differences in each state s_i.

One can also choose to consider only *truncated differences*, that is, only look at whether or not the difference in one byte is zero. While this can also directly lead to various attacks, as impossible differential attacks [BBS99; Knu98], it can also be used to get some results in differential cryptanalysis. Indeed, in most cipher designs, the non-linear component consists of an S-box, a small non-linear function applied several times over all iterations. This S-box is the reason that differential characteristic only holds with a certain probability. Given an S-box S acting on a small number of s bits, and for each pair $(\Delta_{in}, \Delta_{out}) \in \mathbb{F}_2^{2s}$, one can easily compute how many $x \in \mathbb{F}_2^s$ verifies the relation $S(x \oplus \Delta_{in}) = S(x) \oplus \Delta_{out}$. This allows to compute the Difference Distribution Table (DDT) of the S-box, which gives the probability that the above relation holds for each $(\Delta_{in}, \Delta_{out})$. Thus, given a differential characteristic, one can compute the probability that it holds, simply by multiplying the differential probabilities of all S-boxes together [1]. Hence, given a *truncated* differential characteristic, while we cannot determine the exact

1. Using the fair assumption that each round is independent, which while obviously not true, is admitted as a reasonable assumption.

probability that this characteristic holds, we can deduce its maximal probability. Indeed, if the S-box has a maximal differential probability of p, and there are n S-boxes with a non-zero difference (called *active S-boxes*), then the truncated differential characteristic holds with a probability at most p^n. Thus, given the maximal differential probability of the S-box used and the bit-length n_k of the key, one can easily deduce the minimal number of active S-boxes n_{min} that leads to $p^{n_{min}} < 2^{-n_k}$. So, if for a given number of rounds, we can prove that there are at least n_{min} active S-boxes, we know that there would be no differential characteristic with a probability better than 2^{-n_k}, which would mean that finding a pair of plaintexts satisfying this characteristic would *a priori* cost more than an exhaustive search for the key.

Such differentials and truncated differentials can also be considered in the *related-key model*. First introduced in 2009 to attack AES-192 and AES-256 [BK09; BKN09], this model allows the attacker to inject differences in the plaintext, but also in the key. Another worth-mentionning model is the more recent *related-tweak model* for tweakable block ciphers, where the attacker fully controls an additional input for the block cipher called a *tweak* [LGS17; ZD19]. While this model is closer to chosen-plaintext attacks, the tweak is often (but not necessarily) used alongside the key and thus involved in the key schedule, such as in the TWEAKEY framework [JNP14]. Since the attacker can now inject some differences in both the plaintext and the key, this causes a large increase in the complexity to search differential and truncated differential characteristics. Nonetheless, several tools have been designed to tackle this problem [BN10; FJP13; Gér+18].

2.1.1 Designing an optimal key schedule for AES

A few proposals were made to give another, more secure, key schedule for some primitives, such as [Nik10; Cho+11] for AES and [Nik17] for SKINNY and AES-based constructions from FSE 2016 [JN16]. However, their main concern was mostly to design a more secure key schedule, without considering the possible loss in efficiency. To that regard, Khoo *et al.* [Kho+17] proposed a new key schedule for AES which consists of only a permutation at the byte level, based on their proof on the number of active S-boxes in the related-key model for AES. Using a permutation thus leads to a very efficient key schedule, both in software and hardware, and can also make the analysis easier. However, they did not provide any proof of optimality for this permutation but showed that it increases the minimal number of active Sboxes compared to the original key schedule. Thus our main objective was to prove the optimality of their permutation or to find a better one.

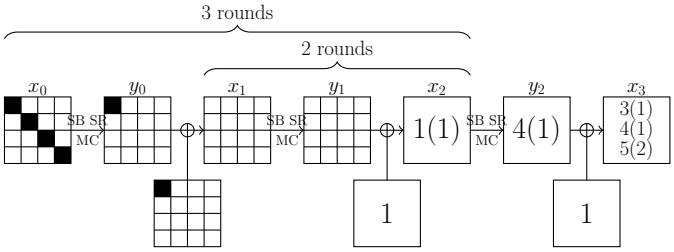

Figure 2.2 – Truncated differential characteristic always valid for 2, 3 and 4 rounds. $x(y)$ means that there are x active S-boxes somewhere in the state, with y columns containing at least one active bytes. Multiple $x(y)$ in a state means that one of them must be true

Generic Bounds

Before trying to find a permutation that reaches a certain number of active S-boxes, we need to study which number of S-boxes we can reach. From the fact that using a permutation as the key schedule implies that the number of active bytes in the round keys is constant, we can deduce several bounds on the number of active S-boxes. To demonstrate these bounds, we show that there is always a differential characteristic of a certain number of active S-boxes, independently from the permutation used in the key schedule.

Proposition 1. *Using a permutation as the key schedule, there is always a truncated differential characteristic of with 1 (resp. 5) active S-box(es) for 2 (resp. 3) rounds. For 4 rounds, there is always a truncated differential characteristic of with either 8, 9 or 10 active S-boxes.*

Such characteristics are depicted in Figure 2.2.

If we try to extend the previous characteristic with one more round, we obtain that there is always a characteristic with either 19, 20, 21, 24 or 25 active S-boxes in the truncated differential setting. However, by considering a totally different truncated characteristic we have the following proposition.

Proposition 2. *For 5, 6 and 7 rounds, there is always a characteristic with respectively 14, 18 and 21 active S-boxes in the truncated differential setting.*

Such truncated characteristics are depicted in Figure 2.3.

Now the first question that we may ask is whether or not there exists a permutation which reaches all those bounds. Fortunately, such a permutation was already found by

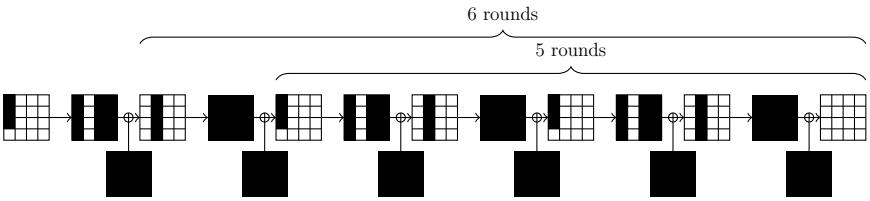

Figure 2.3 – Truncated differential characteristic always valid for 5, 6 and 7 rounds.

Khoo *et al.* in [Kho+17], which is

$$P_{KLPS} = \Big(\begin{array}{cccccccccccccccc} 5 & 2 & 3 & 8 & 9 & 6 & 7 & 12 & 13 & 10 & 11 & 0 & 1 & 14 & 15 & 4 \end{array} \Big).$$

Better Model

The main issue regarding truncated differential characteristics is that some of them may be false, in the sense that it is impossible to find actual values satisfying the characteristic. In order to remove some of these impossible truncated characteristics we used the model proposed by Gérault *et al.* in [Gér+18]. They noticed that because the **MixColumns** operation is linear, the MDS property of the matrix also applies to the sum of two columns. More precisely we have the following property:

Proposition 3. *Let z and z' be two state columns, $w(z)$ and $w(z')$ the number of active bytes and MC the **MixColumns** matrix. Let $y = MC(z)$ and $y' = MC(z')$. Since the matrix MC is MDS we have the three constraints:*

- $w(z) + w(y) = 0$ *or* ≥ 5
- $w(z') + w(y') = 0$ *or* ≥ 5
- $w(z \oplus z') + w(y \oplus y') = 0$ *or* ≥ 5

This proposition is highly effective when the key schedule of AES is replaced by a permutation as it may forbid differences on particular bytes to be the same or to be distinct. Actually we can go a bit further with the next proposition.

Proposition 4. *Let k, x, y, z be four state columns such that $MC(z) = y$, z contains at least one active byte and $x = y \oplus k$. Denote by $i_{y,z}$ the number of inactive bytes in y and z (i.e., $i_{y,z} = 8 - w(y) - w(z)$) and $c_{z,k,x}$ the number of bytes from z that are cancelled by k in x . If $i_{y,z} + c_{y,z,k} \geq 5$, then there is at least one linear equation on some bytes of k . Moreover, this can only happens if $c_{y,z,k} \geq 2$.*

2.1.2 Algorithms and Results

We used the two previous propositions to refine the definition of truncated differential characteristics. More precisely, they allow us to define a more sophisticated model in which extra linear constraints are added and have to be satisfied to expect a valid differential characteristic.

Model. It takes as input a permutation P_k to use as the key schedule and a number of rounds, and output the minimal number of active S-boxes with these parameters in the truncated differential setting. We do take into account the equations coming from the **MixColumns** operation, resulting in a more reliable result, albeit being slower.

Bound on 5 Rounds

Using our new model we were able to refine the bound for 5 AES rounds. We showed there is no permutation that, when used as key schedule, can reach a minimal number of active S-boxes of 18 or higher over 5 rounds. To get this result we mainly used the decomposition of permutations into cycles, identifying the cycles which could belong to a permutation with a number of minimal active Sboxes of 18, i.e. removing cycles leading to a truncated characteristic with less than 18 active Sboxes. Once all such cycles were obtained we tried to compose them and searched for the minimal number of active Sboxes for each resulting permutation.

Unfortunately, our algorithm was too slow to exhaust the case with 17 active Sboxes. However, we were able to perform it partially with 16 active Sboxes and found one permutation which has a minimal number of active S-boxes of 16 over 5 rounds, namely:

$$(15\ 0\ 2\ 3\ 4\ 11\ 5\ 7\ 6\ 12\ 8\ 10\ 9\ 1\ 13\ 14)\,.$$

Bound on 6 Rounds

Due to the huge space search, we used a totally different approach for 6 rounds. Inspired by the work of Nikolic [Nik17], we used a meta-heuristic called simulated annealing. Meta-heuristics are a class of search algorithms which aim to find an (almost) optimal solution to an optimization problem, often inspired by some real-life phenomenon. To be more precise, unlike Constraint Programming or Integer Linear Programming which aims at recovering an optimal solution, meta-heuristics only look for a good enough solution: it may not be optimal, but it should be rather close to an optimal solution.

Number of rounds	**2**	**3**	**4**	**5**	**6**	**7**
Original key schedule	1	3	9	11	$13^{\dagger}$	15
P_{KLPS}	1	5	10	14	$18^{\dagger}$	22
P_k	1	5	10	15	$20^{\dagger}$	23

Table 2.1 – Minimal number of S-boxes that our permutation P_k reaches on a given number of rounds compared to the one from [Kho+17]. †No instantiation with a better probability than 2^{-128}.

We first launched our algorithm for 20 active S-boxes, and were able to find the permutation P_k (given below) reaching this minimal number of S-boxes in about 2^{16} tries:

$$P_k := (8\ 1\ 7\ 15\ 10\ 4\ 2\ 3\ 6\ 9\ 11\ 0\ 5\ 12\ 14\ 13)$$

Reaching 21 S-boxes is still an open question and for reference, we were able to test about 2^{24} permutations in several days.

Tweaking Both ShiftRows and the Key Schedule

Finally, we tried to see if by changing the **ShiftRows** operation in the AES-128, we could reach a better number of active Sboxes, namely 21 or 22. Obviously, we cannot try all possible permutations for **ShiftRows** as there are 2^{44} permutations over 16 elements, and trying 1 permutation takes a non-marginal time.

Relying on some equivalence relations and restricting ourself to permutations achieving full diffusion in at most 3 rounds, we got 3288 possible candidates for the permutation P_s.

We used our meta-heuristic algorithm on several of them and found a pair of permutation (P_s, P_k) reaching 21 active Sboxes after an hundred of trials, trying 2^{25} permutations P_k for each of them.

$$P_s = (0\ 1\ 2\ 4\ 3\ 8\ 9\ 12\ 5\ 13\ 14\ 15\ 6\ 7\ 10\ 11)$$
$$P_k = (10\ 4\ 12\ 11\ 6\ 2\ 5\ 1\ 8\ 0\ 9\ 7\ 13\ 14\ 15\ 3)$$

We also searched for a pair of permutations reaching 22 active Sboxes but were not able to find one after trying a thousand of permutations P_s.

The fact that we were able to build a more resistant cipher from a non-optimal **ShiftRows** operation (achieving full diffusion in 3 rounds instead of 2) is quite interesting as it shows that combining optimal cipher components is not the necessarily optimal.

2.2 Optimal Diffusion Layers of Generalized Feistel Networks

The Feistel network is one of the main generic designs for building modern block ciphers. It was initially proposed in the data encryption standard `DES` [DES77], and is still used in more recent ciphers such as `Twofish` [Sch+98], `Camellia` [Aok+00] or `SIMON` [Bea+13]. The idea behind this construction is to split the plaintext into two halves x_0, x_1, and build the round function which sends (x_0, x_1) to $(x_1, x_0 \oplus F_i(x_1))$, where F_i is a non-linear function for the i-th round. In 1989 at CRYPTO, Zheng *et al.* [ZMI89] proposed some generalizations of the Feistel construction. Especially, they defined the *Type-2 Feistel*[2] construction, which splits the message into $2k$ blocks and uses a round function of the form

$$(x_0, \ldots, x_{2k-1}) \mapsto (x_{2k-1}, x_0 \oplus F_{i,0}(x_1), x_1, x_2 \oplus F_{i,1}(x_3), x_3, \ldots, x_{2k-2} \oplus F_{i,k-1}(x_{2k-1})),$$

where each $F_{i,j}$ is a pseudorandom function for the i-th round (Figure 2.4). This is essentially a parallel application of k Feistels followed by a cyclic shift of the blocks. An interesting property is that when all $F_{i,j}$ are pseudorandom functions, then $2k+1$ rounds are enough to make the corresponding block cipher indistinguishable from a random permutation. At ASIACRYPT'96, Nyberg [Nyb96] studied a variant of the Type-2 Feistel construction using a different permutation than the cyclic shift, called Generalized Feistel Network (GFN).

Definition 1. *Let $2k$ be an even number, n, r be positive integers, and $\{F_{i,j}\}_{i\in\{1,\ldots,r\},j\in\{0,\ldots,k-1\}}$ be a set cryptographic keyed functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. Let π be a permutation over $2k$ elements. A Generalized Feistel Network is a block cipher built as $\mathcal{R}_r \circ \cdots \circ \mathcal{R}_1$, where $\mathcal{R}_i$ is the round function*

$$\mathcal{R}_i : \ (X_0, \ldots, X_{2k-1}) \to \pi(X_0 \oplus F_{i,0}(X_1),\ X_1, \ldots,\ X_{2k-2} \oplus F_{i,k-1}(X_{2k-1}), X_{2k-1})$$

2. Note that some papers use the term Type-2 Generalized Feistel to denote this construction

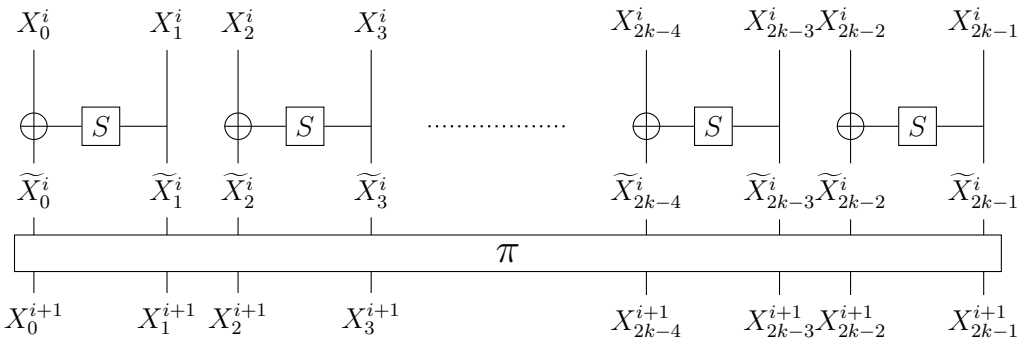

Figure 2.4 – Generalized Feistel Network

Such a construction was used to design more recent block ciphers as for instance `TWINE` [Suz+12] and `Piccolo` [Shi+11].

In the following neither the exact definition of the keyed functions $F_{i,j}$ nor their sizes are relevant. We thus consider all of them as an arbitrary S-box S, leading to the framework depicted in Figure 2.4. As the only variable parameters are thus k and π, we denote by GFN_π^k a GFN with $2k$ blocks that uses the permutation π.

2.2.1 The Problem

It is easy to see from Definition 1 that $X_{\pi(0)}^1$ depends on X_0^0 and X_1^0. More generally, any block X_j^r depends on several blocks from the round 0, i.e. computing X_j^r requires some blocks $\{X_{j_0}^0, \ldots, X_{j_l}^0\}$. We say in that case that any of these $X_{j_i}^0$ *diffuses* to X_j^r, and we focus our study on the number of rounds needed to reach *full diffusion*.

Definition 2. *Let π be a permutation over $2k$ elements. We say that a block X_j^0 fully diffuses after r rounds if for all $i \in \{0, \ldots, 2k-1\}$, X_j^0 diffuses to X_i^r. We say that π reaches full diffusion after r rounds if for all $j \in \{0, \ldots, 2k-1\}$, X_j^0 fully diffuses after r rounds. The smallest r that verifies this property for the block X_i^0 is called the diffusion round of the block X_i^0.*

Note that we need to study both the diffusion over the encryption *and* the decryption process. Indeed, there is no guarantee that an encryption function with good diffusion also keeps this property for its inverse. Since we have $(GFN_\pi^k)^{-1} = GFN_{\pi^{-1}}^k$, we need to study both the diffusion of π and π^{-1}. Naturally, we would like both π and π^{-1} to fully diffuse as quickly as possible, which leads to the following definition.

Definition 3. *Let π be a permutation over $2k$ elements. Denote by $DR_i(\pi)$ the minimum number of rounds r such that X_i^0 fully diffuses after r rounds in GFN_π^k.*

The diffusion round of a permutation π is:

$$DR_{max}(\pi) = \max_{0 \leq i \leq 2k-1} \left\{ DR_i(\pi), DR_i(\pi^{-1}) \right\} \tag{2.1}$$

This definition gives the same importance to the total diffusion of both π and π^{-1}. Definition 3 defines a natural partial order on the permutations: a permutation π_1 is better (at diffusing) than a permutation π_2 if $DR_{max}(\pi_1) \leq DR_{max}(\pi_2)$. A natural problem regarding GFN is thus to determine the *optimal* permutations, the ones leading to the most secure constructions.

Suzaki and Minematsu [SM10]

Searching for the best permutations (for the diffusion) directly can be difficult. A naive way to search for optimal permutations would be to simply go through all of them and check the diffusion one permutation by one. However, there are $(2k)!$ permutations, which quickly grows beyond practical means. For example with $2k = 32$, approximately 2^{117} permutations should be checked.

In [SM10], Suzaki and Minematsu did an exhaustive search for $1 \leq k \leq 8$, and made the observation that every optimal permutation (for such k) mapped even-number input blocks to odd-number output blocks and vice versa. We call such permutations even-odd.

An even-odd permutation π of size $2k$ is denoted by a pair of permutations (p, q) of size k verifying $\forall i \in [0, k-1],\ \pi(2i) = 2 \cdot p(i) + 1$ and $\pi(2i+1) = 2 \cdot q(i)$. The search space is now reduced to $(k!)^2$ permutations.

Cauchois *et al.* [CGT19]

To further reduce the size of the search space, Cauchois *et al.* observed that given two even-odd permutations $\pi = (p, q)$ and $\pi' = (p', q')$, if p and p' share the same cycle structure and if $q = q'$ then the corresponding block ciphers do share the same diffusion round. This directly comes from the fact that the diffusion round is invariant by block reordering.

As a consequence, there are only $\mathcal{N}_k.k!$ permutations to consider instead of $(k!)^2$, where $\mathcal{N}_k$ is the number of partitions of the integer k (which is equal to the number of

cycle structures for p). This is a significant improvement and it allowed Cauchois *et al.* to perform an exhaustive search up to $2k \leq 24$.

2.2.2 A New Algorithm

While Cauchois *et al.* significantly reduced the search space, they checked the remaining permutations one by one. In [Der+19], a joint work with Pierre-Alain Fouque and my two PhD students Baptiste Lambin and Victor Mollimard, we proposed a new algorithm to exhaust this restricted search space in a more clever way.

New representation

The first step of our approach is to give a better representation of the problem for even-odd permutations. Let $\pi = (p, q)$ be an even-odd permutation, r a positive integer and $\mathbb{J}^r$ the set of all permutations σ such that $\sigma = p^{\alpha_1} \circ (p \circ q)^{\beta_1} \circ \ldots \circ p^{\alpha_n} \circ (p \circ q)^{\beta_n}$ with $\alpha_1 + \ldots + \alpha_n + 2\beta_1 + \ldots 2\beta_n = r$. Let us also define $\mathbb{J}^r_i$ as the set $\{\sigma(i) \mid \sigma \in \mathbb{J}^r\}$. We show that π fully diffuses after R rounds if and only if $|\mathbb{J}^{R-3}_i| = k$ for all integers $i \in [0, k-1]$.

For example, we give in Table 2.2 the diffusion tables for the cyclic shift ($p = (7, 0, 1, 2, 3, 4, 5, 6)$ and $q = (0, 1, 2, 3, 4, 5, 6, 7)$) and one of the optimal permutations proposed by [CGT19] ($p = (6, 3, 7, 1, 0, 2, 4, 5)$ and $q = (3, 5, 1, 6, 4, 0, 2, 7)$) for $k = 8$ and $R = 8$.

New algorithm

Let us pick a permutation p and assume we want to find q such that $|\mathbb{J}^r_x| = k$ for all $x \in [0, k-1]$. We can first pick $x_0 \in [0, k-1]$, guess enough images of q to compute $\mathbb{J}^r_{x_0}$ and then check whether $|\mathbb{J}^r_{x_0}| = k$ or not before repeating the process. To minimize the number of guesses between each check we propose the following strategy:

— Pick x_0 on the smallest cycle of p;

— If not already processed, set $x_{i+1} = p(x_i)$.

To support this strategy, let us study the case $r = 5$ as an example. Computing $\mathbb{J}^5_x$ requires to guess the images of 11 elements by q:

$$\{x, p(x), p^2(x), p^3(x), p^4(x), pq(x), p^2q(x), p^3q(x), pqp(x), p^2qp(x), pqp^2(x)\}.$$

If x belongs to a small cycle of p then several of those elements will be the same, decreasing the number of guesses to perform. For instance, if x is a fixed point of p then we have to

i	0	1	2	3	4	5	6	7
p^5	3	4	5	6	7	0	1	2
p^4q	4	5	6	7	0	1	2	3
p^3qp	4	5	6	7	0	1	2	3
p^2qp^2	4	5	6	7	0	1	2	3
pqp^3	4	5	6	7	0	1	2	3
qp^4	4	5	6	7	0	1	2	3
p^2qpq	5	6	7	0	1	2	3	4
pqp^2q	5	6	7	0	1	2	3	4
qp^3q	5	6	7	0	1	2	3	4
$pqpqp$	5	6	7	0	1	2	3	4
qp^2qp	5	6	7	0	1	2	3	4
$qpqp^2$	5	6	7	0	1	2	3	4
$qpqpq$	6	7	0	1	2	3	4	5
$\lvert\mathbb{J}_i^5\rvert$	4	4	4	4	4	4	4	4

i	0	1	2	3	4	5	6	7
p^5	4	3	5	1	6	7	0	2
p^4q	3	2	1	4	0	6	7	5
p^3qp	2	6	7	5	1	3	4	0
p^2qp^2	6	7	4	0	5	2	3	1
pqp^3	1	4	3	2	0	6	7	5
qp^4	2	5	7	6	3	1	4	0
p^2qpq	7	1	0	6	3	5	2	4
pqp^2q	4	5	2	1	7	0	6	3
qp^3q	5	0	6	2	4	3	1	7
$pqpqp$	5	0	6	3	2	4	1	7
qp^2qp	0	3	1	7	6	5	2	4
$qpqp^2$	3	1	2	4	7	0	5	6
$qpqpq$	1	6	4	3	5	7	0	2
$\lvert\mathbb{J}_i^5\rvert$	8	8	8	8	8	8	8	8

Table 2.2 – Diffusion tables for the cyclic shift (left table) and one optimal permutation proposed by [CGT19] (right table).

guess the images of only 4 elements by q:

$$\{x, pq(x), p^2q(x), p^3q(x)\}.$$

Next, computing $\mathbb{J}^5_{p(x)}$ requires to guess the images of 11 elements by q:

$$\{p(x), p^2(x), p^3(x), p^4(x), p^5(x), pqp(x), p^2qp(x), p^3qp(x), pqp^2(x), p^2qp^2(x), pqp^3(x)\}.$$

We observe that several of them were already required to compute $\mathbb{J}^5_x$ and thus we have to guess at most 4 new images:

$$\{p^5(x), p^3qp(x), p^2qp^2(x), pqp^3(x)\}.$$

However, if p does not have a small enough cycle, the overall complexity is quite close to $k!$ since q has to be almost fully guessed to compute the first set $\mathbb{J}^r_{x_0}$.

To lower the number of guesses one have to perform, we can adopt a different strategy. Because of the structure of $\mathbb{J}^r$ we can define $\mathbb{P}^r_x$ and $\mathbb{Q}^r_x$ such that $\mathbb{J}^r_x = \mathbb{P}^r_x \cup \mathbb{Q}^r_x$ and $\mathbb{J}^{r+1}_x = q(\mathbb{P}^r_x) \cup p(\mathbb{J}^r_x)$ for any integers r and x. Now let us assume we guessed enough

images of q to compute $\mathbb{J}^{r-1}$. Instead of guessing images of each element from $\mathbb{P}_x^{r-1}$ by q we keep the constraint:

$$[0, k-1] \setminus (p(\mathbb{J}_x^{r-1}) \cup q(\mathbb{P}_x^{r-1} \setminus \mathbb{A})) \subset q(\mathbb{P}_x^{r-1} \cap \mathbb{A}),$$

where $\mathbb{A}$ is the set of unset elements by q. This is very effective to decrease the number of guesses made between each check. For instance, with $r = 5$, we now need to guess the images of only 7 elements:

$$\{x, p(x), p^2(x), p^3(x), pq(x), p^2q(x), pqp(x)\}.$$

Note that the constraint should be checked in two steps and updated each time a new guess is performed. First we verify that $|\,[0, k-1] \setminus (p(\mathbb{J}_x^{r-1}) \cup q(\mathbb{P}_x^{r-1} \setminus \mathbb{A}))| \leq |\mathbb{P}_x^{r-1} \cap \mathbb{A}|$ and then that $|p(\mathbb{J}_x^{r-1}) \cup q(\mathbb{P}_x^{r-1} \setminus \mathbb{A}) \cup \mathbb{I}| = k$ where $\mathbb{I}$ is the set of unset images of q.

To reduce further the number of guesses, we can write $\mathbb{J}_x^r = p^2(\mathbb{J}_x^{r-2}) \cup qp(\mathbb{J}_x^{r-2}) \cup pq(\mathbb{P}_x^{r-2})$ and keep the constraint:

$$[0, k-1] \setminus (p^2(\mathbb{J}_x^{r-2}) \cup pq(\mathbb{P}_x^{r-2} \setminus \mathbb{A}) \cup q(p(\mathbb{J}_x^{r-2}) \setminus \mathbb{A})) \subset pq(\mathbb{P}_x^{r-2} \cap \mathbb{A}) \cup q(p(\mathbb{J}_x^{r-2}) \cap \mathbb{A}).$$

But verifying this constraint is complicated and thus our idea is to only verify a weaker version. More precisely, we verify that there exist two sets $\mathbb{S}_1 \subset p(\mathbb{I})$ and $\mathbb{S}_2 \subset \mathbb{I}$ such that $|\mathbb{S}_1| = |\mathbb{P}_x^{r-2} \cap \mathbb{A}|$, $|\mathbb{S}_2| = |p(\mathbb{J}_x^{r-2}) \cap \mathbb{A}|$ and

$$[0, k-1] \setminus (p^2(\mathbb{J}_x^{r-2}) \cup pq(\mathbb{P}_x^{r-2} \setminus \mathbb{A}) \cup q(p(\mathbb{J}_x^{r-2}) \setminus \mathbb{A})) \subset \mathbb{S}_1 \cup \mathbb{S}_2.$$

In practice it is rare for the weaker constraint to be satisfied while the original one is not. Furthermore it is very fast to check it as it requires computing the size of only few intersections of sets.

It seems natural to try writing $\mathbb{J}_x^r$ using $\mathbb{J}_x^{r-3}$ and $\mathbb{P}_x^{r-3}$ but unfortunately the corresponding weaker constraint does not filter enough to reduce the overall complexity.

2.2.3 Results

Using our new approach, we were able to prove that with even-odd permutations:

— For $k = 14$, 15, 16 and 18, the optimal number of rounds for full diffusion is 9.

— For $k = 17$, the optimal number of rounds for full diffusion is 10.

— For $k = 19$, 20 and 21, the optimal number of rounds for full diffusion is at least 10 and at most 11.

In particular we solved the 10-year-old problem of finding optimal permutations for 32 blocks GFN.

Open problems

There are still many open problems regarding optimal permutations for GFN. In our opinion, the most interesting one would be to show that for any value of k there is at least one optimal permutation which is even-odd.

Chapter 3

Tools for Cryptanalysis

Contents

To evaluate the security of cryptographic primitives, cryptographers aim at finding the best possible attacks and distinguishers, which typically means the ones covering as many rounds as possible with the smallest complexity. Doing so most often requires to explore a large search space in order to find the best parameters for the technique. The help of a computer is thus becoming mandatory in cryptanalysis works to support researchers. I dedicated a large part of my research in developing tools for this purpose and I will present three of them ([DF16; DF20; DDV20]) in this chapter.

3.1 Demirci-Selçuk Meet-in-the-Middle Attacks

During my PhD I mainly worked on improving the Demirci-Selçuk attacks [DS08], a type of advanced meet-in-the-middle attacks, and obtained some of the best known

attacks against round-reduced AES [DF13; DFJ13]. Since then I continued to work on this cryptanalysis technique and published several works related to it [DP15; BDP15; DF16; Shi+18; Der+18c]. At CRYPTO'16, in a joint work with Pierre-Alain Fouque, we proposed a new tool to automatically search for the best Demirci-Selçuk attacks against a large class of block ciphers.

3.1.1 Generalized Demirci-Selçuk (GDS) Attack

To design the tool we needed to generalize the original attack of Demirci and Selçuk against AES. We thus proposed a generic view of this cryptanalysis technique applicable to any block ciphers. Let $E = E_3 \circ E_2 \circ E_1$ be an encryption function split into three parts. For the first step we pick a truncated difference Δ_X with b_i active bits, propagate it through E_1^{-1} (resp. $E_3 \circ E_2$) with probability 1 and denote the set of active bits by I_P (resp. I_C). Then, for the second step, we mount a basic meet-in-the-middle attack against $E = E_3 \circ (E_2 \circ E_1)$: let Y be the output state of $E_2 \circ E_1$, we pick b_o bits of Y and denote by O_P (resp. O_C) the bits required to compute their difference in Y from the difference in the plaintexts (resp. ciphertexts).

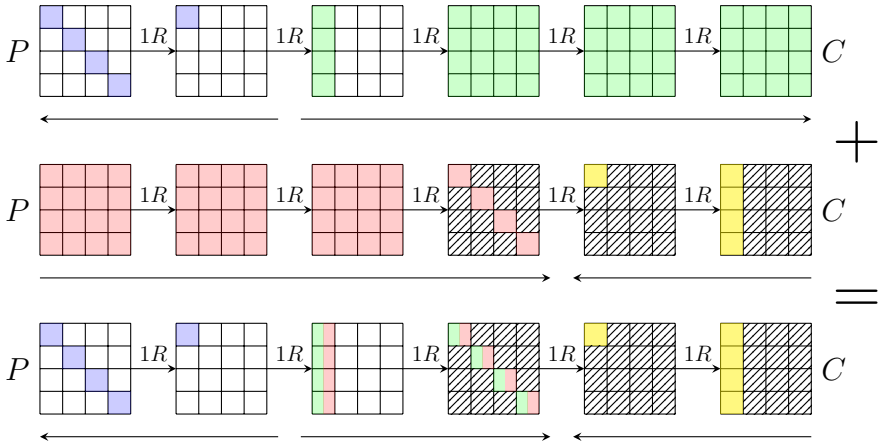

Figure 3.1 – Example of GDS attack (on 6-round AES). I_P is in blue, I_C in green, O_P in red and O_C in yellow. Hatched bytes play no roles and white bytes are constant.

To explain further the GDS attack we introduce the definition of a b-δ-set:

Definition 4 (b-δ-set)**.** *A b-δ-set is a set of 2^b states such that b bits are active and take all the possible values while the others bits are constant. We also assume that the states of a b-δ-set are sorted according to differences.*

The structure of the Generalized Demirci-Selçuk attack is then as follows:

— **Offline phase:**

1. Consider the encryption of a b_i-δ-set $\{x^0, x^1, \ldots\}$ corresponding to the truncated difference Δ_X through E_2.
2. Guess the value of internal bits from $I_C \cap O_P$ for message x^0.
3. Deduce the differences in the b_o chosen bits of Y for the b_i-δ-set.
4. Store them as a sequence of $2^{b_i} - 1$ b_o-bit values in a hash table.

— **Online phase:**

1. Pick a plaintext P.
2. Guess the value of I_P for P and identify a set $\{P, P^1, P^2, \ldots\}$ leading to a b_i-δ-set associated to Δ_X.
3. Ask for the corresponding ciphertexts.
4. Guess the value of internal bits in O_C and partially decrypt the ciphertexts to compute the differences in the b_o chosen bits of Y.
5. Check whether the sequence belongs to the hash table. If not, discard the guess.

The complexity of this procedure depends directly on how many values the sets I_P and $I_C \cap O_P$ can assume (denoted by $\mathcal{S}$), and on how fast all the possible values of sets $I_P \cup O_C$ and $I_C \cap O_P$ can be enumerated (denoted by $\mathcal{T}$):

— **Data:** $(2^{b_i} - 1) \cdot \mathcal{S}(I_P)$ adaptively chosen plaintexts,
— **Time (online):** $2^{b_i} \cdot \mathcal{T}(I_P \cup O_C)$ partial encryptions,
— **Memory:** $b_o \cdot (2^{b_i} - 1) \cdot \mathcal{S}(I_C \cap O_P)$ bits,
— **Time (offline):** $2^{b_i} \cdot \mathcal{T}(I_C \cap O_P)$ partial encryptions.

At the end of this attack we expect $\min(1, \mathcal{S}(I_C \cap O_P) \cdot 2^{-b_o(2^{b_i}-1)}) \cdot \mathcal{S}(I_P \cup O_C)$ candidates to remain for $I_P \cup O_C$. Thus b_i and b_o have to be chosen such that they provide enough filtration, but expanding them also increases the size of the sets I_P, I_C, O_P and O_C which then may rise the complexity of the resulting attack.

Remarks:

— In the case where the truncated difference Δ_X does not make Δ_P fully active, *i.e.* differences in some plaintext bits are null, the attack can be turned into a chosen-plaintext attack by asking either for a structure of plaintexts. Actually this is (almost) always better to do so since, in general, $(2^{b_i} - 1) \cdot \mathcal{S}(I_P)$ is greater than $2^{|\Delta_P|}$.

— Some extra memory can be used to map each sequence to its corresponding value of $I_C \cap O_P$.
— Given two invertible matrices M_1 and M_2, we can rewrite the encryption function $E = (E_3 \circ M_2^{-1}) \circ (M_2 \circ E_2 \circ M_1^{-1}) \circ (M_1 \circ E_1)$. Hence the sentences "with b_i active bits" or "pick b_o bits of Y" should be understood as "with b_i active *linear combinations of* bits" or "pick b_o *linear combinations of* bits of Y".

3.1.2 A New Ad-hoc Tool

To handle as many block ciphers as possible we used a generic representation of a block cipher. Namely, it has to be represented using equations as:

$$\sum \alpha_i S_{i,j}(x_{\sigma(0)}, \ldots, x_{\sigma(j)}) + \sum \beta_j x_j + c = 0,$$

where the α_i's, β_j's and c belong to the same finite field and where the $S_{i,j}$ are seen as black box Sboxes. We then faced several problems to design an efficient tool.

Linear combinations. A priori, to fully explore the search space, we have to try all pairs of invertible matrices M_1 and M_2 and write $E = (E_3 \circ M_2^{-1}) \circ (M_2 \circ E_2 \circ M_1^{-1}) \circ (M_1 \circ E_1)$. But the size of internal states of typical block ciphers forbids such a naive approach. Instead we developed a branch-and-cut algorithm in which minimal equations (i.e. equations involving a minimal set of variables regarding the inclusion) are exhausted using an early abort strategy, removing the ones that would not lead to an *optimal* attack. We refer the interested readers to [DF16] for more details on the algorithm.

Evaluating the complexities. As explained above, determining the data, time and memory complexities requires an algorithm computing $\mathcal{S}(X)$ and $\mathcal{T}(X)$ under the constraints of the block ciphers equations for any set of variables X. While the generic problem is complicated, in our case it is only about finding relations between the (linear combinations of) round key bits involved in the attack. To solve the problem we used an adapted version of the tool we developed with Charles Bouillaguet during my PhD [BDF11].

AND and OR. Handling multi-variables S-boxes naturally leads to the particular case of AND and OR. While until now S-boxes were considered as black boxes, both those functions have a special property that can be properly handled. Indeed, the following

equation holds for any variables x and y:

$$\mathrm{AND}(x, y) \oplus \mathrm{AND}(x \oplus \Delta x, y \oplus \Delta y) = \mathrm{AND}(x, \Delta y) \oplus \mathrm{AND}(\Delta x, y) \oplus \mathrm{AND}(\Delta x, \Delta y).$$

In particular, if $\Delta y = 0$ then $\mathrm{AND}(x, y) \oplus \mathrm{AND}(x \oplus \Delta x, y) = \mathrm{AND}(\Delta x, y)$, meaning that computing the difference after the AND requires Δx and y but not the actual value of x. This is also true for the OR operator since $\mathrm{OR}(x, y) = \mathrm{AND}(x, y) \oplus x \oplus y$. As a consequence, in the previous algorithms, we have to define new sets I'_P, I'_C, O'_P and O'_C containing the variables required to compute the differences in each variable of I_P, I_C, O_P and O_C respectively, and use them instead for the complexity computations.

3.1.3 Applications

Our tool handles a large class of block ciphers and allowed us to find several new attacks. For instance it found the best known attacks against the block cipher `mCrypton`, breaking one more round for all the three key sizes.

Interestingly, the building blocks of our tool can be used in a straightforward way to search for basic meet-in-the-middle attacks and impossible differential attacks. For instance we automatically recovered the 6-round meet-in-the-middle attack described by Biham *et al.* in [Bih+15] against `IDEA` and found better impossible differential attacks against `SIMON` than Boura *et al.* in [BNS14].

3.2 Algorithms for Division Property

Integral cryptanalysis exploits distinguishers computing the sum of ciphertexts corresponding to a set of plaintexts spanning a linear subspace. This technique was originally introduced by Knudsen in [DKR97] as a specific attack against the byte-oriented structure of the block cipher `SQUARE` and unified by Knudsen and Wagner in [KW02]. In 2000, Ferguson *et al.* [Fer+00] presented at FSE powerful attacks based on integral distinguishers against round-reduced versions of `AES`, named *Partial Sum attacks*. In particular they described a practical attack against 6 rounds which is still one of the best known attacks against `AES`. Integral distinguishers [BS01] were found by propagating through the round functions simple properties on words composing the internal states: ALL (the word takes all the possible values once), BALANCED (the word sums to zero), CONSTANT (the value of the word is constant).

The so-called division property, introduced by Todo at Eurocrypt'15 [Tod15b], is a method to find more sophisticated integral distinguishers. The idea behind the division property technique is actually quite simple. Let f and g be two n-bit functions and assume the goal is to find an integral distinguisher on $g \circ f$ without computing it explicitly. Let $y_i = f_i(x_0, \ldots, x_{n-1})$ and $z_i = g_i(y_0, \ldots, y_{n-1})$ be the intermediate and final expressions of the coordinate functions of f and of g, and let m_z be a monomial in the z_i's, and so m_z is a polynomial consisting of some monomials m_y. Division property actually captures that if for a subset $\mathcal{X}$ of $\mathbb{F}_2^n$ each monomial m_y appearing in m_z satisfies $\bigoplus_{x \in \mathcal{X}} m_y(x) = 0$ then $\bigoplus_{x \in \mathcal{X}} m_z(x) = 0$. Several variants of this property were used to find integral distinguishers. For instance, in [TM16], Todo and Morii used that if all monomials m_y but one sum to zero then $\bigoplus_{x \in \mathcal{X}} m_z(x) = 1$. And more recently, in both [Hao+20] and [Heb+20], the exact relation was used: $\bigoplus_{x \in \mathcal{X}} m_z(x) = 0$ if and only if the number of monomials m_y for which $\bigoplus_{x \in \mathcal{X}} m_y(x) = 1$ is even.

In practice we cannot try all possible sets $\mathcal{X}$ nor compute the corresponding sums for all monomials involved in the description of a cryptographic primitive. Furthermore we typically want integral distinguishers independent from the key, adding an extra complexity to the problem. However it is easy to show that if P is a polynomial in variables $(x_1, \ldots, x_n)$ then $\bigoplus_{(x_1,\ldots,x_i) \in \mathbb{F}_2^i} P(x_1, \ldots, x_n) = 0$ for each value of $(x_{i+1}, \ldots, x_n)$ if and only if P does not involve a monomial containing all the variables $x_1, \ldots, x_i$. This property can be understood more easily using higher-order differential and means that if we derive i times w.r.t. to the first i variables, a multivariate polynomial P that does not contain a monomial involving the $x_1 x_2 \ldots x_i$ monomial, then we get the 0 polynomial. Thus integral distinguishers are highly related to the maximal monomials involved in a polynomial and division property can be seen as a method to track them through an iterated function.

3.2.1 Searching for integral distinguishers

The main difficulty is to efficiently modelize the propagation of division property through the round functions of a cipher. Except in [TM16] where Todo and Morii used an ad-hoc tool to exhaust division trails on `SIMON`-32, searching for integral distinguishers based on division property usually relies on generic solvers for MILP, SAT or SMT models. In [Xia+16] Xiang *et al.* show that it is possible to describe transitions through small Sboxes with inequalities by computing the convex hull of points. This work has been extended by Zhang and Rijmen [ZR19] to binary linear mapping. Eskandari *et al.* in [Esk+18] have built a tool called Solvatore to find such division property trails using

a SAT solver and found many new integral distinguishers. The difficulty of the search procedure depends on the cipher and on the variant of division property implemented. The original variant is the simplest to search for but is also the less accurate as it may miss some cancellations of monomials and thus miss distinguishers. In [Heb+20], Hebborn *et al.* worked with the exact variant and described a new method dedicated to (small) block ciphers aiming at proving that for each linear combination of the ciphertext bits and for each monomial of degree $n-1$ in the plaintexts bits, there is at least one key (considering independent round keys) for which the monomial appears in the ANF of the linear combination. They used a heuristic approach to find round keys for which evaluating the parity of division trails is the cheapest. As a result they found that 13-round `SKINNY-64`, 11-round `Gift` and 11-round `PRESENT` are all immune to integral distinguishers if considering independent round keys.

3.2.2 Several Improvements

I worked on division property with Pierre-Alain Fouque and my PhD student Baptiste Lambin and we published two papers [LDF20; DF20] related to this topic. We proposed several refinements of the cryptanalysis technique which allowed us to find new distinguishers requiring either less data or covering more rounds. We present two of them in this section and refer the interested readers to the original publications for more details.

Linear combinations. We observed that for a given block cipher E, one should consider $L_{out} \circ E \circ L_{in}$, where both L_{out} and L_{in} are linear mappings, since division property is not linearly invariant contrary to differential nor linear cryptanalysis. For instance, let f_k be the encryption function

$$f_k(x, y) = (p_0(k)x \oplus p_1(k)y, p_2(k)x \oplus p_3(k)y)$$

where $p_0, \ldots, p_3$ are non-zero polynomials and x, $y \in \mathbb{F}_2$. In that case classical application of division property would conclude that no output bit is balanced. But if either $p_0 = p_2$ or $p_1 = p_3$ then the xor of both output bits is balanced. This may lead to new distinguishers but the drawback is that the search space is greatly increased. However we showed that not all linear mappings have to be considered. Regarding the output and since we are looking for integral distinguishers, we are only interested in knowing whether the i-th bit is balanced or not. Hence there is no reason to consider invertible matrices, linear combinations are enough, reducing the number of mappings to try for an n-bit cipher from

$\mathcal{O}(2^{n^2})$ to $\mathcal{O}(2^n)$. As for the output, it is not required to try all invertible matrices at the input to cover the whole search space. Actually, what matters for integral distinguishers is the vector space spawn by constant (linear combinations of) bits (more precisely, bits that will be constant in the integral distinguisher). Indeed, let $P(x_1, \dots x_n)$ be a polynomial and let $\mathcal{H}(i, j)$ be the property that a polynomial does not contain any monomial greater than or equal to (i.e. multiple of) $x_i \dots x_j$. We know there exist two polynomials P_1 and Q_1 such that $P(x_1, \dots, x_n) = x_1 P_1(x_2, \dots, x_n) \oplus Q_1(x_2, \dots, x_n)$. In particular, for any $k \in \{1, \dots, n\}$, P satisfies $\mathcal{H}(1, k)$ if and only if P_1 satisfies $\mathcal{H}(2, k)$. Now let be $j \in \{2, \dots, n\}$ and consider polynomial $P'(x_1, \dots, x_n) = P(x_1 \oplus x_j, x_2, \dots, x_n)$. We have the following equalities:

$$\begin{aligned} P'(x_1, \dots, x_n) &= P(x_1 \oplus x_j, x_2, \dots, x_n) \\ &= (x_1 \oplus x_j) P_1(x_2, \dots, x_n) \oplus Q_1(x_2, \dots, x_n) \\ &= x_1 P_1(x_2, \dots, x_n) \oplus (x_j P_1(x_2, \dots, x_n) \oplus Q_1(x_2, \dots, x_n)) \\ &= x_1 P_1(x_2, \dots, x_n) \oplus Q'_1(x_2, \dots, x_n) \end{aligned}$$

As a consequence, P' satisfies $\mathcal{H}(1, k)$ if and only if P_1 satisfies $\mathcal{H}(2, k)$ and thus P' satisfies $\mathcal{H}(1, k)$ if and only if P satisfies $\mathcal{H}(1, k)$. Hence, any invertible matrix that does not modify the vector space of constant bits does not modify the integral distinguisher. In particular, when looking only for the existence of an integral distinguisher *i.e.* without optimizing the data complexity, it is enough to exhaust the only linear combinations of bits that will be constant, reducing the number of mappings to test from $\mathcal{O}(2^{n^2})$ to $\mathcal{O}(2^n)$.

Propagation table of Super-Sboxes. At ASIACRYPT'16, Xiang *et al.* [Xia+16] proposed an algorithm to compute the *propagation table* of an n-bit to n-bit function f. The propagation table of f is a table T such that for any $\mathbf{m} \in \mathbb{F}_2^n$, $T[\mathbf{m}]$ contains all possible monomials $\mathbf{m'}$ such that the transition $\mathbf{m} \xrightarrow{f} \mathbf{m'}$ is valid, fully describing the propagation rules through f. The algorithm produces the propagation table in roughly $\mathcal{O}(2^{3n})$ operations which is practical up to $n \approx 16$. To improve the precision of division property our goal was to remove false trails, which correspond to valid trails $\mathbf{m_0} \xrightarrow{f_0} \mathbf{m_1} \xrightarrow{f_1} \mathbf{m_2}$ for which the transition $\mathbf{m_0} \xrightarrow{f_1 \circ f_0} \mathbf{m_2}$ is actually invalid because of monomial cancellations. Thus, our idea was to build the propagation table of Super-Sboxes. Introduced in [GP10] by Gilbert and Peyrin, Super-Sboxes are Sboxes operating on columns and equivalent to a first application of the simple Sbox on each word of the column, an application of the **MixColumns** operation, a XOR with a key and a second application of the simple Sbox.

Because of the key addition between the two layers of Sboxes, a naive approach would require to run the previous algorithm for all possible values of the (part of) round key used in the Super-Sbox and then merge the propagation tables. This would quickly make the computation untractable. Instead we proposed a new algorithm, taking as input a collection of k n-bit functions and outputting the propagation table containing all the valid transitions for at least one of the function. We mainly reorganized the computations to avoid redundant ones and its complexity is in $\mathcal{O}(kn2^{2n} + 2^{3n})$. Note that typical value for k is 2^n and so our algorithm has complexity $\mathcal{O}(n2^{3n})$, to be compared to $\mathcal{O}(2^{4n})$, the cost of calling 2^n times the original algorithm.

3.2.3 A New Tool for Division Property

In [TM16], Todo and Morii proposed a way to search for integral distinguishers based on the division property, with a complexity upper bounded by 2^n, where n is the block size of the block cipher. In practice, they said that their algorithm is not suitable for block ciphers with block size beyond 32 bits, and thus the number of possible targets is very limited. However, a lot of work has been done towards efficiently searching such distinguishers, based on either MILP or SAT/SMT solvers.

Regarding MILP-based search algorithms, the main point is to generate sets of inequalities describing all the propagation tables involved in the decomposition of the cipher. But the number of inequalities required to describe a 16-bit propagation table seems too large to be handled efficiently by any MILP solver. For instance, the propagation table of the Super-Sbox of `Midori-64` contains approximately 2^{23} elements. Hence we developed a dedicated algorithm to search for integral distinguishers based on a branch-and-bound approach. This was the first time one showed a practical algorithm to search for division trails on 64-bit block ciphers not relying on generic solvers for MILP, SAT or SMT models.

3.2.4 Results

Using our tool we found new integral distinguishers against the three well-studied block ciphers `SKINNY-64` [Bei+16], `Midori-64` [Ban+15] and `HIGHT` [Hon+06], increasing the number of rounds covered compared to previously best known integral distinguishers. We also experimentally verified some distinguishers found on smaller instances in order to validate our tool. For instance, we searched for low data distinguishers by fixing some input bits of the Super-Sboxes to constant and we found integral distinguishers requiring

only 2^{15} chosen plaintexts against both 8-round SKINNY-64 and 6-round Midori-64.

3.3 Boomerang Characteristics

Nowadays we know how to design ciphers resistant to differential cryptanalysis, ciphers for which we can give upper bounds on the probability of the best differential characteristics. To go further, Wagner proposed the *boomerang attack* in [Wag99]. The main idea introduced by Wagner is that combining two short differentials may lead to a higher probability than one long differential. In boomerang attacks, a cipher E is regarded as the composition of two sub-ciphers E_0 and E_1 so that $E = E_1 \circ E_0$. Suppose there exist both a differential $\alpha \to \beta$ for E_0 and a differential $\gamma \to \delta$ for E_1 with probabilities p and q respectively. If we assume the two differentials are independent then we obtain a boomerang distinguisher of probability:

$$\mathbb{P}\left(E^{-1}(E(P) \oplus \delta) \oplus E^{-1}(E(P \oplus \alpha) \oplus \delta) = \alpha\right) = p^2q^2.$$

However, in practice the independence assumption usually does not hold, especially at the junction of both the lower and upper differentials. At SAC'07, Wang *et al.* [WKD07] first gave some evidences for non-returning boomerangs (i.e. $\mathbb{P} = 0$ instead of p^2q^2). In 2011, Murphy [Mur11] provided several examples for both AES and DES of boomerangs never coming back. Similar results were obtained by Kircanski in [Kir15]: a SAT solver is used to show that previous rectangle/boomerang attacks on XTEA [Lu09], SM3 [WKD07] and SHACAL-1 [DKK06] primitives were based on incompatible characteristics.

Recently, in [Cid+18], Cid *et al.* proposed a new tool named *boomerang connectivity table* (BCT) to overcome the dependency issues. The BCT is actually a precomputation of all boomerangs through one single Sbox. Its main advantage is to provide a unified view of the *switches* previously introduced to refine the computation of the probability [BK09; DKS14]. In [SQH19], Song *et al.* give a generalized framework for the BCT and propose a method to precisely evaluate the probability of a boomerang. They reevaluated the probability of several boomerang distinguishers from [LGS17] against both SKINNY and AES, showing their exact probability was much higher than expected.

3.3.1 Searching Boomerangs

One natural question when facing a new cryptanalysis technique is how to find the best distinguishers. For boomerang distinguishers, the classical approach is to first search for two short characteristics with high probability and to combine them. But we believe this approach should now be deprecated since the dependency in the middle rounds may hugely affect the probability of the distinguisher and thus it seems sub-optimal to search for both the lower and upper differentials independently.

In [Cid+17], Cid *et al.* used a MILP model to study the ladder switch for a boomerang attack on Deoxys. A more generic approach was proposed in [LS19], where Liu *et al.* describe a MILP model to directly search for the best boomerang distinguisher against the block cipher `GIFT`. The cipher is decomposed into three parts E_0, E_m and E_1 where E_m is restricted to one single round, the junction of both differentials which handles the BCTs. With this model they found a new boomerang distinguisher on 19-round `GIFT`, achieving a better probability than when merging two optimal short trails.

3.3.2 A New Tool

In [DDV20], a joint work with Stephanie Delaune and the Master student Mathieu Vavrille, we proposed to go further than both [SQH19] and [LS19] by providing a new tool to search for boomerang distinguishers. One limitation of the MILP model of Liu *et al.* is that it handles only one round for the middle part while Song *et al.* have shown that dependencies could affect much more rounds, for instance up to 6 rounds for `SKINNY`. First, we proposed a new approach to turn a MILP model to search for truncated characteristics into a MILP model to search for truncated boomerang characteristics. The main novelty was that this model handles the dependencies in the middle rounds automatically. Furthermore, there is no need to specify which rounds are the middle ones, this is also directly handled by the model. Second, we proposed a new Constraint Programming (CP) model to search for the best instantiation of a truncated boomerang characteristic. This model even goes further by clustering instantiations to improve the probabilities. Finally, we systematized the method from [SQH19] to precisely compute the probability of a boomerang.

From truncated differentials to truncated boomerangs. The most interesting technique described in this paper is certainly the process to turn a MILP model to search for truncated characteristics into a MILP model to search for truncated boomerang character-

istics. Let E be a classical SPN cipher with R rounds operating on an n-cell internal state and such that the round function is composed of a **SubCell** operation, a key addition and a linear layer which multiplies the internal state by a matrix M (at the cell level). We also assume the key schedule is fully linear. The first part of the model consists in writing twice the MILP model for truncated differential, once for the upper characteristic and once for the lower one. Such models are somehow easy to write and are already available for several block ciphers [ZDY19; Bei+16]. We consider for each cell of each internal state of the upper (resp. lower) characteristic a binary variable $\mathsf{isActiveUp}$ (resp. $\mathsf{isActiveLo}$) indicating whether the cell is active or not. To represent the fact that some differences will take any value uniformly, we introduce *free* variables (non free variables will be called *controlled* variables). Controlled variables are the differences that will be set to a fixed value in the characteristic.

We introduce two sets of binary variables for each characteristic: $\mathsf{isFreeXup}$ and $\mathsf{isFreeSBup}$ (resp. $\mathsf{isFreeXlo}$ and $\mathsf{isFreeSBlo}$) to indicate whether a difference will be free before and after the Sbox respectively. For the upper characteristic if a difference is free before an Sbox (i.e. $\mathsf{isFreeXup} = 1$), then it is free after the Sbox ($\mathsf{isFreeSBup} = 1$). For the lower characteristic, if a difference is free after an Sbox, then it is free before the Sbox (because the propagation is done in the opposite direction). This leads to the constraints:

$$\forall 0 \leq r < R, 0 \leq i < n, \quad \begin{aligned} \mathsf{isFreeSBup}[r][i] &\geq \mathsf{isFreeXup}[r][i] \\ \mathsf{isFreeXlo}[r][i] &\geq \mathsf{isFreeSBlo}[r][i] \end{aligned}$$

Those variables are also related to both $\mathsf{isActiveUp}$ and $\mathsf{isActiveLo}$ because a difference can be set to 0 only if the difference is controlled. Thus we have the constraints:

$$\forall 0 \leq r < R, 0 \leq i < n, \quad \begin{aligned} \mathsf{isActiveUp}[r][i] &\geq \mathsf{isFreeSBup}[r][i] \\ \mathsf{isActiveLo}[r][i] &\geq \mathsf{isFreeXlo}[r][i] \end{aligned}$$

Another important constraint is the one stating that a free difference propagates with probability 1 (i.e. no cancellations occur). For the upper characteristic we define $\mathsf{depsU}(i)$ as the set of all the indexes j such that the coefficient $m_{i,j}$ of the matrix M (of the linear layer) is non-zero. For the lower one, we define depsL in a similar way but for the matrix

M^{-1}. Then the constraints of propagation of free variables are simply:

$$\forall 0 < r < R, 0 \leq i < n, \quad \begin{aligned} \mathsf{isFreeXup}[r][i] &= \bigvee_{j \in \mathsf{depsU}(i)} \mathsf{isFreeSBup}[r-1][j] \\ \mathsf{isFreeSBlo}[r-1][i] &= \bigvee_{j \in \mathsf{depsL}(i)} \mathsf{isFreeXlo}[r][j] \end{aligned}$$

In order to apply the differential tables as the DDT or the BCT, we need an extra constraint to ensure that the probability of each Sbox can be computed. More precisely, we require that at most 2 variables can be free for each Sbox (considering upper and lower characteristic, before and after the Sbox). This leads to the constraints:

$$\forall 0 \leq r < R, 0 \leq i < n, \quad \begin{aligned} \mathsf{isFreeSBup}[r][i] + \mathsf{isFreeSBlo}[r][i] &\leq 1 \\ \mathsf{isFreeXup}[r][i] + \mathsf{isFreeXlo}[r][i] &\leq 1 \end{aligned}$$

With all these constraints, the solutions generated will lead to truncated boomerang characteristics. We emphasize with our new set of constraints there are no *middle rounds* defined for our truncated boomerang characteristics. In particular, the BCTs are not necessarily all on the same round but may be spread over several rounds. Thus our modelization is more generic than the previous ones, in particular than the modelization proposed by Liu *et al.* in [LS19].

3.3.3 Results and Open Problems

We applied our tool to the block cipher `SKINNY` [Bei+16] and found many new distinguishers on all versions of the ciphers. Our results are given in Table 3.1. All previous results from [SQH19] were improved, in particular we found a new boomerang distinguisher on 18-round `SKINNY`-128/256 (i.e. on the TK2 model) with probability $2^{-47.37}$ while the previous best distinguisher had probability $2^{-77.83}$. We experimentally verified some of the distinguishers to confirm the probabilities.

Open problems. We found two main limitations regarding our MILP model. First it is possible to have truncated boomerangs that differ only on some $free$ variables. These truncated boomerangs are duplicates in the point of view of distinguishers, and thus instantiations will be almost the same. Moreover, when applying the procedure to compute the probability of the boomerang, they will have exactly the same probability because the input and output will be the same. This also makes the number of solutions growing exponentially. The tool was configured to find the N best solutions and thus would find

version	block size	nb of rounds	proba
SK	64	11	$2^{-59.23}$
	128	13	$2^{-112.53}$
		14	$2^{-128.52}$
TK1	64	14	$2^{-40.34*}$
		15	$2^{-53.16}$
	128	14	$2^{-42.27}$
		15	$2^{-69.14}$
		16	$2^{-87.15}$
		17	$2^{-107.84}$

version	block size	nb of rounds	proba
TK2	64	17	$2^{-27.65*}$ $(2^{-29.78})$
		18	$2^{-38.20*}$
		19	$2^{-54.36}$
	128	18	$2^{-47.37}$ $(2^{-77.83})$
		19	$2^{-61.83}$
		20	$2^{-85.77}$
TK3	64	22	$2^{-39.44*}$ $(2^{-42.98})$
		23	$2^{-57.93}$
	128	22	$2^{-47.34}$ $(2^{-48.30})$
		23	$2^{-61.80}$
		24	$2^{-86.09}$

Table 3.1 – Results for different versions and number of rounds on SKINNY. Probabilities marked with asterisks have been validated experimentally. The four previous results from [SQH19] are also given in parenthesis.

non optimal solutions. But as there were too many of them, it was not able to go much further than optimal objective (in a reasonable time limit). The second limitation is related to the key schedule. We force it to be linear to be able to implicitly set all differences in the round keys as controlled. Thus a natural question is how to modify the model to handle more complex key schedule as for instance the one of AES.

Chapter 4

Real-life and Practical Cryptography

Contents

Many attacks and distinguishers proposed against cryptographic primitives do not threaten their security in real-life situation and are mainly theoretical weaknesses, highlighting unexpected behavior and meaning it should be possible to reach better security level. For instance, the block cipher MYSTI1 [Mat97] is considered broken since Todo showed an attack against it [Tod15a]. However this attack requires the knowledge of almost the full codebook ($2^{63.994}$ chosen plaintexts among a codebook of size 2^{64}) and to run an algorithm performing around 2^{107} non-trivial operations. The requirement is so high that in practice MYSTI1 can still be used safely but we would recommend to use a block cipher for which knowing the full codebook does not decrease the time complexity of retrieving the key. Thus it is interesting to study the resistance of a cipher against

practical attacks.

During my PhD I studied low-data-complexity attacks against round-reduced AES and found some of the best attacks in this setting. During my PostDoc I participated to the *PRINCE Challenge* in which the goal was to mount the fastest attacks against round-reduced versions of the blockcipher PRINCE using at most 2^{20} chosen plaintexts or 2^{30} known plaintexts. I won some of those challenges and the results were published in [DP15] and [DP20].

In this chapter I describe the results obtained in 3 recent papers ([DLU19; Der+18b; Bei+21]) regarding practical attacks and real-life cryptography.

4.1 Cryptanalysis of SKINNY in the Framework of the SKINNY 2018-2019 Cryptanalysis Competition

In order to motivate external cryptanalysis of their family of ciphers, SKINNY designers launched several one-year competitions. The first one started in 2016 and called for cryptanalysis of small-scaled variants of 18 up to 26 rounds of SKINNY-64-128, and of 22 up to 30 rounds of SKINNY-128-128. The two papers that won the competition are [Ank+17] for being the first submission that attacks up to 20 rounds of SKINNY-64-128 and [LGS17] for being the first submitted work to successfully attack up to 23 rounds of SKINNY-64-128.

The challenges launched in 2017 were similar, except that the number of rounds one has to break was higher. Nobody won these contests.

The last competition started on the 1st of April 2018 and ended on February 28, 2019. This time, the goal was to mount a practical key-recovery attack of small-scaled versions of SKINNY for which sets of only 2^{20} pairs (plaintext, ciphertext) were provided. The designers offered rewards for the teams that would break the maximum number of rounds for SKINNY-64-128 or SKINNY-128-128 [1].

4.1.1 Remark on the Provided Messages

While looking for messages with specific patterns, we realized that the plaintexts provided for the challenges were not uniformly distributed.

1. All the information on the competitions and on the cipher in general can be found on https://sites.google.com/site/skinnycipher/home.

To illustrate this, we provide in Figure 4.1 the distribution of the value of nibble 0 (top left corner in the Skinny internal state) and of nibble 15 (bottom right) in the set provided for the 12-round attack on SKINNY-64-128.

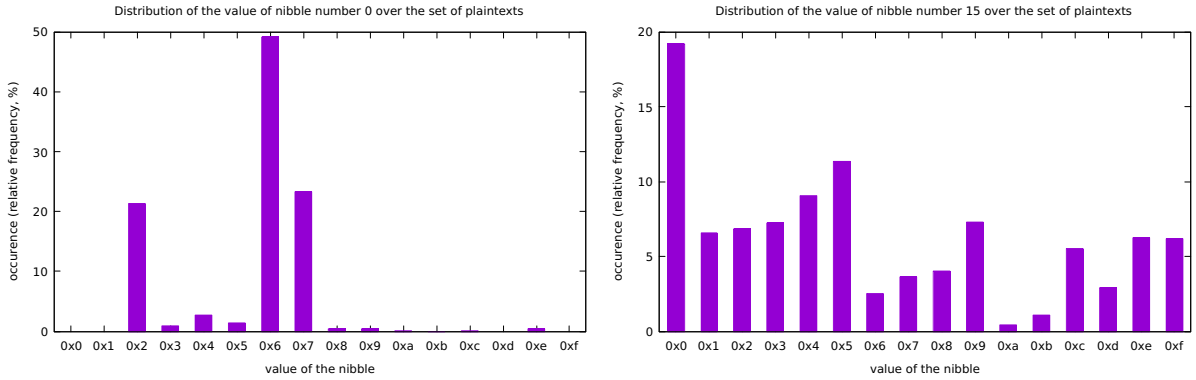

Figure 4.1 – Distribution of the value of nibble 0 (left) and of nibble 15 (right) of the 2^{20} plaintexts provided for the 12-round attack on SKINNY-64-128.

In fact, the bias observed in Figure 4.1 is present on the other nibble positions too, and we made the following observations:

— All the nibbles positioned at even indices have a distribution similar to the one of the left bar chart of Figure 4.1: the most frequent value is 0x6 (occurring roughly half of the time), followed by 0x7 and 0x2. The other values are very rare.

— The nibbles at odd positions don't have such a strong bias. Still, some values are more frequent than the others, like 0x0 that appears in one case out of 5.

It is rather direct to make the link between this distribution and the one of a text in UTF-8 code: indeed, the first hint comes from the fact that the UTF-8 code of the lower-case letters goes from 0x61 to 0x7a, which explains the overwhelming occurrence of the nibble 0x6 (followed by the nibble 0x7) in the distribution of nibbles at even positions. Also, a character that is ought to appear frequently is the space, encoded by 0x20. This one explains the third dominant higher nibble value (0x2) and the high number of occurrences of 0x0 in the lower nibbles.

This guess was confirmed once we printed the plaintexts. For instance, looking at the messages given for the challenge on 4 rounds SKINNY-64-128, we read:

> Project Gutenberg's Alice's Adventures in Wonderland, by Lewis Carroll This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever.

And few lines later, confirming that this is the book, one can read:

> [...] when suddenly a White Rabbit with pink eyes ran close by her. There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, 'Oh dear! Oh dear! I shall be late!'

Other data sets correspond to other books (for instance Metamorphosis, by Franz Kafka or The Prince, by Nicolo Machiavelli).

This high bias in the messages implies that we have some collisions on the plaintext values. In the file provided for 12 rounds of SKINNY-64-128 we counted 925615 different pairs ($2^{19.82}$, so $2^{16.90}$ collisions) out of the 2^{20} provided. The plaintext that appears the most corresponds to ." " (dot (0x2e), closed quote (0xe2809d), space (0x20), open quote (0xe2809c)) with 289 occurrences.

4.1.2 Results

Any human-readable text has a low entropy and so does the book whose encryption was provided in the SKINNY 2018-2019 cryptanalysis competition. This fact can be exploited by a cryptanalyst, who can devise attacks that would be ineffective in the classic known-plaintext scenario with uniformly distributed plaintexts. In this case, we showed that many pairs or even quadruples of plaintext blocks can be found in the provided datasets for which particular (differential) zero-sum properties hold with probability 1 after 6 rounds of SKINNY-64-128 and 7 rounds of SKINNY-128-128.

As a result, we were able to mount practical key recovery attacks up to 12-round SKINNY-64-128 and 10-round SKINNY-128-128 from given sets of 2^{20} messages. Our attacks consist in leveraging distinguishers based on a probability-1 truncated first-order and second-order differential paths. The attacks are possible because the provided sets of messages give much more exploitable pairs than what one could have expected from a random set. This also highlights the importance of the mode of operation used with a cipher.

Table 4.1 – Complexities of our attacks: the data complexity corresponds to the number of messages that are actually exploited in the attack, while time complexity is expressed in number of basic operations.

Version	Rounds	Technique	Data	Time	Memory
SKINNY-64-128	12	Truncated diff.	64	$2^{51.95}$	256 GB
SKINNY-128-128	10	2nd-order truncated diff.	24	2^{52}	0.5 GB

4.2 On Recovering Affine Encodings in White-Box Implementations

Historically, cryptanalysis is performed within the black-box model: the cryptographic algorithm under attack is executed in a trusted environment, and the view of the attacker is limited to the input-output behavior of the algorithm. Depending on the type of attack under consideration, the attacker may be able to observe the inputs and outputs of encryption or decryption queries, and perhaps choose the corresponding inputs, but nothing more. Such attack models are particularly relevant in scenarios where the attacker does not have direct access to an implementation of the scheme, either because it is executed remotely, or within a protected hardware environment such as a secure enclave.

Since the advent of side-channel attacks however, new attack models have come into the light, wherein the attacker has access to some auxiliary information leaked by the implementation. These models are sometimes called *gray-box* models, in contrast with the *black-box* model outlined in the previous paragraph. Attacks in the gray-box model may exploit physical leakage such as computation time, power consumption, or electromagnetic leakage, among many others. Such attacks can result in practical breaks against schemes that would otherwise appear secure in the standard black-box model.

4.2.1 White-box Cryptography

Going one step further, in 2002, Chow *et al.* introduced the white-box model [Cho+02a; Cho+02b]. In this model, the attacker has full access to an implementation of the target cryptographic algorithm, including the ability to control its execution environment. Therefore he can observe memory content, set breakpoints in the execution flow, change arbitrary values in the code or the memory, *etc.* In this setting, the security assumptions

of the black-box model clearly no longer hold. However, it may still be desirable that the adversary should be unable to extract the secret key of the cryptographic algorithm under attack.

This model is relevant in the context of software distribution, whenever a piece of software containing sensitive cryptographic information (such as an encryption algorithm) is to be widely distributed, and hence can be downloaded and analyzed by adverse parties. The most prominent application occurs in Digital Rights Management, where attackers may wish to recover a decryption key used to protect copyrighted content (digital music, TV broadcasts, video games, *etc*). A successful attacker is then able to distribute the secret key to unauthorized users, providing them with illegitimate access to the protected content. In effect, the goal is to protect sensitive functions within the deployed software, such as cryptographic algorithms, in much the same way that a trusted environment would protect security-critical functions in a hardware context. Ideally, white-box cryptography would thus achieve the software equivalent of trusted enclaves, specialized to particular cryptographic algorithms.

In order to achieve this goal, white-box cryptography techniques attempt to obfuscate the implementation of the target cryptographic algorithm. Ideally, an attacker in possession of the obfuscated cipher should be unable to interact with it in any meaningful way, beside simply executing it on chosen inputs. While Barak *et al.* have shown that general program obfuscation is impossible [Bar+01], the context of white-box cryptography presents two key differences. The first is that white-box cryptography merely attempts to obfuscate particular function families (such as block ciphers), which Barak *et al.*'s result has no bearing on. Another key difference is that white-box models do not generally require guarantees as strong as those offered by black-box obfuscation: in the case of a white-box implementation of AES for instance, it may be enough that the adversary is unable to recover the secret key (for a detailed discussion of white-box models, see [Del+13; Fou+16]).

The CEJO framework

In their original 2002 articles, Chow *et al.* proposed such a white-box scheme for DES and AES [Cho+02a; Cho+02b]. While their proposals were quickly broken [JBF02; BGE04], their work opened the path to white-box encryption. Follow-up works often reused the same general framework, which we will call the "CEJO framework".

In the CEJO framework, round functions are obfuscated by being composed with

carefully crafted input and output encodings. In the white-box implementation of a cipher, each round function $E^{(r)}$ at round r is replaced by ${f^{(r+1)}}^{-1} \circ E^{(r)} \circ f^{(r)}$, where ${f^{(r+1)}}^{-1}$, $f^{(r)}$ are bijections called respectively the *input* and *output encoding*. By design, the output encoding of each round is canceled out by the input encoding of the next round.

$$\cdots \circ \underbrace{{f^{(r+1)}}^{-1} \circ E^{(r)} \circ f^{(r)}}_{F^{(r)}} \circ \underbrace{{f^{(r)}}^{-1} \circ E^{(r-1)} \circ f^{(r-1)}}_{F^{(r-1)}} \circ \ldots$$

Figure 4.2 – The CEJO framework.

For each round, the white-box implementation gives access to the encoded version of the round function $F^{(r)} = {f^{(r+1)}}^{-1} \circ E^{(r)} \circ f^{(r)}$, but not directly to the underlying round function $E^{(r)}$. The full implementation of the cipher can thus be written as $E^{(R)} \circ \cdots \circ E^{(1)}$.

Chow *et al.* proposed to define the encodings $f^{(r)}$ as the composition of a non-linear mapping and an affine mapping. The idea is to follow a classic concept in symmetric cryptography : the non-linear mapping will add some *confusion* on the intermediate values of the state, while the affine mapping will add some *diffusion* (see Sec. 3.3 and 3.4 in [Cho+02b]). In addition, in a typical SPN block cipher, round keys are XORed into the inner state of the cipher. In that case, whenever the constant of the affine encoding is uniformly random, a single obfuscated round completely hides the value of the round key, which implies that a successful key-recovery attack must target multiple rounds simultaneously. Thus the CEJO framework is a natural approach to attempt to obfuscate a block cipher, especially an SPN cipher such as `AES`.

In addition to the above, some external input/output encodings M_{out}/M_{in} can be added before and after the cipher. In that case, the implementation provides a map from encoded plaintexts to encoded ciphertexts. These encodings are merged into the tables used for the initial and final encoded round function. The implementation is then equivalent to an encoded version of the cipher, which can be expressed as

$$M_{out} \circ E^{(R)} \circ \cdots \circ E^{(1)} \circ M_{in}.$$

External encodings can be used to increase security, as the attacker is denied direct access to raw plaintexts/ciphertexts. On the other hand, external encodings assume that the implementation surrounding the white-box cipher takes these encodings into account. As such, a white-box implementation with external encodings is not properly speaking an

implementation of the cipher it contains. For this reason, in this work, we shall explicitly signal the presence of external encodings, and use the term white-box implementation *with external encodings* when appropriate.

It is crucial that, given the encoded round function $F^{(r)}$, the adversary should be unable to compute and peel off the encodings ${f^{(r+1)}}^{-1}$ and $f^{(r)}$. Indeed, for typical ciphers such as AES, granting direct access to a single round E would allow the adversary to easily recover the corresponding round key, and from there the secret key of the cipher. However attacks on white-box implementations typically achieve precisely this, by taking advantage of the specific structure of the encodings A and B. In white-box implementations following the CEJO framework, encodings are composed of a very simple non-linear layer, together with a more complex affine layer. Attacks generally peel off the non-linear component, then proceed to recover the affine layer. This is typically achieved in an ad-hoc way, by exploiting specific properties of the scheme under attack.

4.2.2 Results

In a joint work with Fouque, Lambin and Minaud published at TCHES [Der+18b], we proposed a generic algorithm to recover affine encodings for any white-box implementation of a cipher following the CEJO framework, independent of the way the encodings are built. More generally, our algorithm solves the affine equivalence problem (given two maps F and S with the promise that they are affine equivalent, compute affine maps $\mathcal{A}$, $\mathcal{B}$, such that $F = \mathcal{B} \circ S \circ \mathcal{A}$) whenever one of the two maps is composed of the parallel application of distinct S-boxes.

Our main algorithm is very similar to one of the steps of the structural cryptanalysis of SASAS by Biryukov and Shamir [BS01], combined with a generic affine equivalence algorithm; for this purpose, we use the recent algorithm by Dinur [Din18], but the same attack would also work with the classic affine equivalence algorithm by Biryukov, De Cannière, Braeken and Preneel [Bir+03]. Thus the components we use are not essentially new. However, to the best of our knowledge, the fact that they enable breaking all white-box schemes following the design of Chow *et al.* in a generic way has not yet been explicitly pointed out in the literature, or analyzed in detail, despite the fact that the SASAS algorithm predates both these schemes and their attacks. As a result, in our experience, this fact is also largely ignored by practitioners in the industry.

By design, our attack applies to a large class of white-box schemes following the CEJO framework, including [Cho+02a; Cho+02b; Kar10]. Beyond the previously cited schemes,

which were already broken by ad-hoc attacks, we illustrate our attack on a new white-box design by Baek, Cheon and Hong [BCH16]. One distinctive feature of this design that makes it particularly attractive to illustrate our attack (beside not being previously cryptanalyzed) is that it increases the state size by obfuscating two parallel rounds of AES, precisely to prevent generic attacks from being able to recover the affine encodings of the scheme. Indeed Baek *et al.* estimated the security level of their proposal to 110 bits based on their own specialized version of an affine equivalence algorithm. However our generic attack on this scheme requires only about 2^{35} basic operations.

As a second contribution, we analyzed the scheme by Baek *et al.* more closely, and introduced another technique able to break this scheme. This new technique extracts and solves a standalone problem from the scheme by Baek *et al.*. Ultimately, it is able to recover the secret key of the scheme in time complexity 2^{31}. This is verified with an implementation. This dedicated attack on Baek *et al.*'s scheme is also more powerful as it allows us to fully recover the key, while the generic attack only creates a decryption function without recovering the key.

Affine Equivalence Problem

In an SPN cipher, a round function is composed of an affine layer (in which we include key addition), and a non-linear S-box layer. The S-box layer S consists of the application of k parallel m-bit S-boxes, where $n = km$ is the block size. As a result, when encoding a round function using affine encodings, the encoded round function may be written as $F = \mathcal{B} \circ S \circ \mathcal{A}$, folding the affine layer into one of the encodings. A natural problem in this setting is the *affine equivalence problem*: namely, to recover affine encodings $\mathcal{A}$ and $\mathcal{B}$, given $F = \mathcal{B} \circ S \circ \mathcal{A}$, and knowing S. More precisely, since $\mathcal{A}$ and $\mathcal{B}$ may not be uniquely defined, the problem can be stated as: given S and F as before, find affine maps $\mathcal{A}'$, $\mathcal{B}'$ such that $F = \mathcal{B}' \circ S \circ \mathcal{A}'$.

The general affine equivalence algorithm by Dinur [Din18] solves precisely this problem whenever the degree of S is maximal while the classic algorithm by Biryukov *et al.* [Bir+03] assumes no special structure on S). However its complexity is $\mathcal{O}\left(n^3 2^n\right)$, which makes it unsuitable for recovering encodings on a typical block size of 128 bits. In contrast, we focused on the case where S is made up of k parallel m-bit S-boxes. In this setting, we proposed an algorithm that solves the affine equivalence problem with a (typically much lower) time complexity of $\mathcal{O}\left(2^m n^3 + \frac{n^4}{m} + 2^m m^2 n\right)$. For the AES parameters $n = 128$,

$m = 8$, $k = 16$, this yields a time complexity of 2^{32} basic operations [2] (to be compared with 2^{149} basic operations if the generic algorithm by Dinur were applied naively).

As noted earlier, due to its genericity, our attack applies to essentially all white-box schemes following the CEJO framework: this includes the original designs by Chow *et al.* [Cho+02a; Cho+02b], and later proposals [XL09; Kar10]. In the case of Karroumi's scheme [Kar10], while it does not seem to follow the CEJO framework at first glance, it has been later shown that this scheme is equivalent to the CEJO framework [Lep+13; MRP13], and hence our technique applies directly.

The main limitation of our attack is that it only targets affine encodings, whereas most white-box schemes following the CEJO framework also use non-linear encodings in addition to affine encodings [Cho+02a; Cho+02b; Kar10; BCH16]. When non-linear encodings are used, our attack does not break the scheme by itself. However, even in the presence of non-linear encodings, the first step of attacks typically consists in peeling off the non-linear encoding layer first [BGE04; BCH16], which does not apply to the state as a whole, and leaves the attacker with an instance of the previous problem. In this context, our algorithm provides a powerful tool, which is able to recover affine encodings in a very general setting.

Dedicated attack

As a second contribution, we took a closer look at the scheme by Baek *et al.*. We identified another angle from which the scheme can be attacked. At the core of this second approach lies the following problem. Let F, h_1, h_2 be three non-linear mappings from m bits to m bits, and let A_1, A_2 be two linear mappings on m bits. Given oracle access to $G(x, y) = F(A_1(x) \oplus A_2(y)) \oplus h_1(x) \oplus h_2(y)$, recover A_1 and A_2 (up to equivalence). We solved this problem and deduced an attack against the white-box scheme by Baek *et al.* with time complexity $\sim 2^{31}$ operations. We implemented the full attack, and were able to recover the secret key (and external encodings) in about 12 seconds on a standard desktop computer.

2. In practice the constants hidden in the $\mathcal{O}$ notation for our algorithm are quite small, and we disregard them when giving complexity estimates.

4.3 Cryptanalysis of the GPRS Encryption Algorithms GEA-1 and GEA-2

General Packet Radio Service (GPRS) is a mobile data standard based on the GSM (2G) technology. With its large deployments during the early 2000s worldwide, GPRS (including EDGE) was the technology for many of us, which provided us the first mobile Internet connection. While some countries are about to sunset 2G technology (or have already done so), other countries rely on GPRS as a fallback data connection. Consequently, the security of those connections was and is still relevant for a large user base. In the wireless medium, an attacker conducts an eavesdropping attack by merely sniffing the traffic in the victim's vicinity. To protect against eavesdropping GPRS between the phone and the base station, a stream cipher is used and initially two proprietary encryption algorithms GEA-1 and GEA-2 were specified.

In 2011, Nohl and Melette analyzed the security of GPRS traffic and showed that GPRS signals could easily be eavesdropped [NM11]. This was reported as a serious weakness, especially since some providers did not activate encryption at all. However, according to the authors, most operators at that time employed the proprietary encryption algorithms GEA-1 or GEA-2 for encrypting the GPRS traffic. They also reported the reverse-engineering of those encryption algorithms. Without presenting all of the specification details, the following properties of the design of GEA-1 have been shown:

- It is a stream cipher which works on an internal state of 96 bits and uses a 64-bit key.
- A non-linear function is employed for initialization. [3]
- The state is kept in three registers of sizes 31, 32, and 33 bits. [4]
- The state update function is linear, i.e., the registers are LFSRs.
- The function that generates the output stream has algebraic degree 4.

For GEA-2, it was reported that it employs a similar algebraic structure to its predecessor GEA-1. While the key size for GEA-2 is 64 bits as well, the internal state was reported to be of size 125 bits.

Nohl and Melette claimed that GEA-1 has severe weaknesses against algebraic attacks, mainly due to the linearity of the state update function and the availability of a long

3. See minute 32:15 of the recorded talk.
4. The size of the registers are visible in the live state-recovery attack, see minute 48:25 of the recorded talk.

keystream to the adversary. Live on stage, a state-recovery attack was performed that took less than 15 minutes using "a Gaussian equation solver based on some SAT solver ideas" (minute 48:40 of the recorded talk). However, details of this attack are not available.

Interestingly, the ETSI prohibited the implementation of `GEA-1` in mobile phones in 2013, while `GEA-2` and the non-encrypted mode are still mandatory to be implemented today [ETS18].

4.3.1 Description of `GEA-1` and `GEA-2`

Despite the hints of deliberately weakening `GEA-1` for export and a demonstrated attack, a public cryptanalysis of `GEA-1` and `GEA-2` was still missing. Hopefully, we obtained the detailed description of the two algorithms `GEA-1` and `GEA-2` from an anonymous source. Therefore we verified the correctness of the algorithms by *a*) using test vectors that are available on github [Med] and *b*) checking the interoperability with commercial phones using the osmocom project [osm]. Both experiments confirmed the correct functionality; thus, we can assume that the provided algorithms are accurate with a high degree of certainty.

For the encryption, the `GEA` algorithms take the following input parameters: the plaintext, which is the GPRS LLC (Logical Link Control) frame, the key (K), the direction bit (uplink/downlink), and the IV (Input) that consists of an increasing counter for each frame.

As we will see, `GEA-2` is an extension of `GEA-1` – with slight but crucial modifications. For this reason, we describe `GEA-1` first and explain the differences and extensions for `GEA-2` in a second step. An overview of the keystream generation of `GEA-1` and `GEA-2` is shown in Figure 4.3.

`GEA-1`

`GEA-1` is built from three linear feedback shift registers over $\mathbb{F}_2$, called A, B and C, together with a non-linear filter function, called f. The registers A, B, C have lengths $31, 32$ and 33, respectively, and f is a Boolean function of seven variables of degree 4. The registers work in Galois mode. This means that if the bit that is shifted out of a register is 1, the bits in a specified set of positions in the register are flipped. The specification of

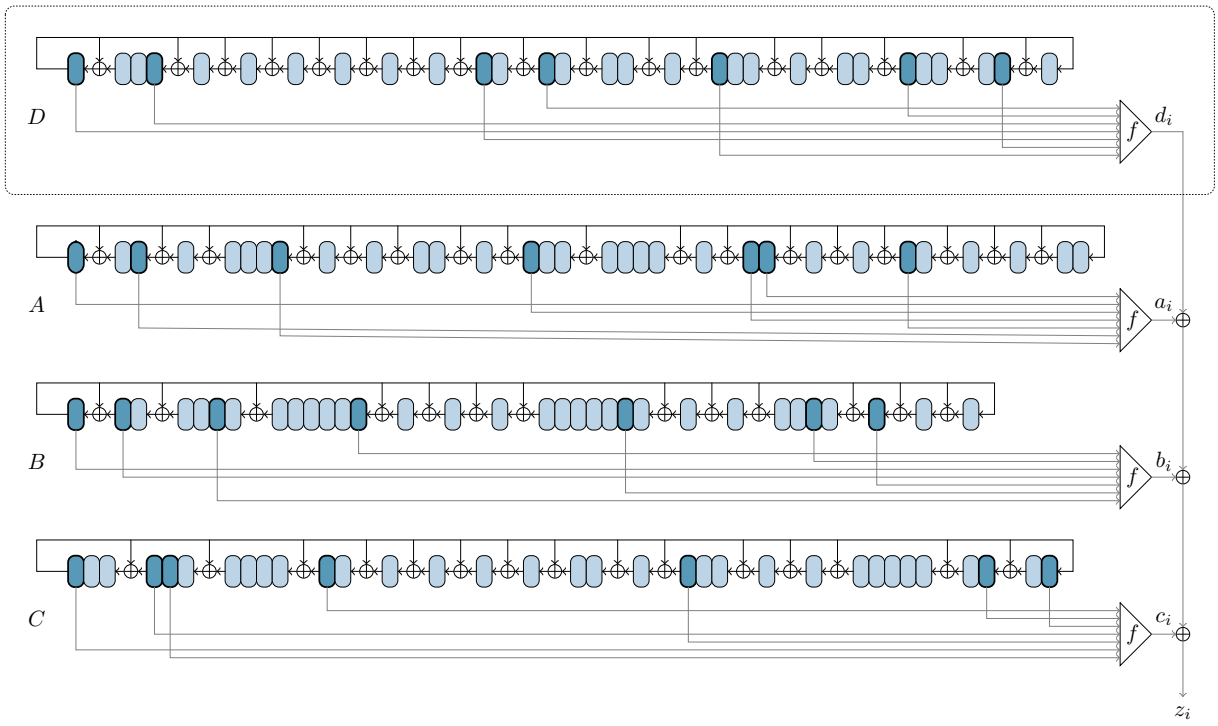

Figure 4.3 – Overview of the keystream generation of GEA-1 and GEA-2. The D register is only present in GEA-2.

$f = f(x_0, x_1, \ldots, x_6)$ is given in algebraic normal form as follows:

$$
\begin{aligned}
&x_0x_2x_5x_6 + x_0x_3x_5x_6 + x_0x_1x_5x_6 + x_1x_2x_5x_6 + x_0x_2x_3x_6 + x_1x_3x_4x_6 \\
&+ x_1x_3x_5x_6 + x_0x_2x_4 + x_0x_2x_3 + x_0x_1x_3 + x_0x_2x_6 + x_0x_1x_4 + x_0x_1x_6 \\
&+ x_1x_2x_6 + x_2x_5x_6 + x_0x_3x_5 + x_1x_4x_6 + x_1x_2x_5 + x_0x_3 + x_0x_5 + x_1x_3 \\
&+ x_1x_5 + x_1x_6 + x_0x_2 + x_1 + x_2x_3 + x_2x_5 + x_2x_6 + x_4x_5 + x_5x_6 + x_2 + x_3 + x_5
\end{aligned}
$$

Initialization. The cipher is initialized via a non-linear feedback shift register of length 64, denoted as S. This register is filled with 0-bits at the start of the initialization process. The input for initializing GEA-1 consists of a public 32-bit initialization vector IV, one public bit dir (indicating direction of communication), and a 64-bit secret key K. The initialization starts by clocking S 97 times, feeding in one input bit with every clock. The input bits are introduced in the sequence $IV_0, IV_1, \ldots, IV_{31}, dir, K_0, K_1, \ldots, K_{63}$. When all input bits have been loaded, the register is clocked another 128 times with 0-bits as input. The feedback function consists of f, xored with the bit that is shifted out and the

next bit from the input sequence. See Figure 4.4 for particular tap positions.

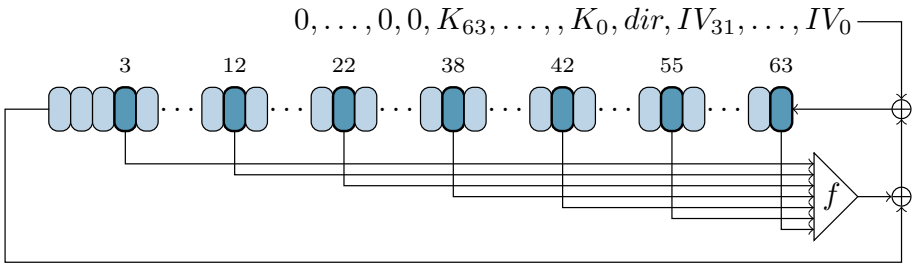

Figure 4.4 – Initialization of register S

After S has been clocked 225 times, the content of the register is taken as a 64-bit string $s = s_0, \ldots, s_{63}$. This string is taken as a seed for initializing A, B and C as follows. First, all three registers are initialized to the all-zero state. Then each register is clocked 64 times, with an s_i-bit xored onto the bit that is shifted out before feedback. Register A inserts the bits from s in the natural order $s_0, s_1, \ldots, s_{63}$. The sequence s is cyclically shifted by 16 positions before being inserted to register B, so the bits are entered in the order $s_{16}, s_{17}, \ldots, s_{63}, s_0, \ldots, s_{15}$. For register C the sequence s is cyclically shifted by 32 positions before insertion starts. Figure 4.5 depicts the process for register B. If any of the registers A, B or C end up in the all-zero state, the bit in position 0 of the register is forcibly set to 1 before keystream generation starts.

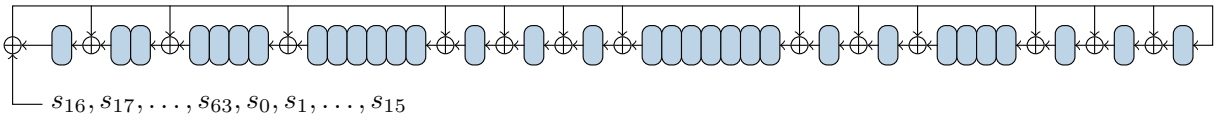

Figure 4.5 – Initialization of register B

Keystream Generation. When all registers have been initialized, the actual keystream generation starts. This is done by taking the bits in seven specified positions in each register to be the input to f. The three outputs from the f-functions are xored together to produce one bit of the keystream. Figure 4.3 shows the particular feedback positions of each register, as well as showing which positions form which input to f. In Figure 4.3, the topmost arrow in the input to f represents x_0, and the input at the bottom is x_6. After calculating the keystream bit, all registers are clocked once each before the process repeats.

GEA-2

The cipher GEA-2 is a simple extension of GEA-1. A fourth register of length 29, called D, is added to the system together with an instance of f. During keystream generation, the output of f from the D register is added to the keystream together with the three others at each clock, as shown in Figure 4.3. The initialization process of GEA-2 follows the same mode as for GEA-1, but it is done in a longer register that is clocked more times.

Initializing GEA-2. As for GEA-1, the initialization of GEA-2 is done via a non-linear feedback shift register, called W. The length of W is 97, and uses f as its feedback function. The inputs to GEA-2 are the same as for GEA-1; a 32-bit IV and a direction bit dir that are public, and a secret 64-bit key K.

Initialization starts with W being set to the all-zero state. Next, it is clocked 97 times, inserting one bit from the input sequence for each clock. The order for inserting IV, dir and K is the same as for GEA-1. After K_{63} is inserted, W is clocked another 194 times, with 0 as input. This process, together with the particular tap positions for f, is shown in Figure 4.6.

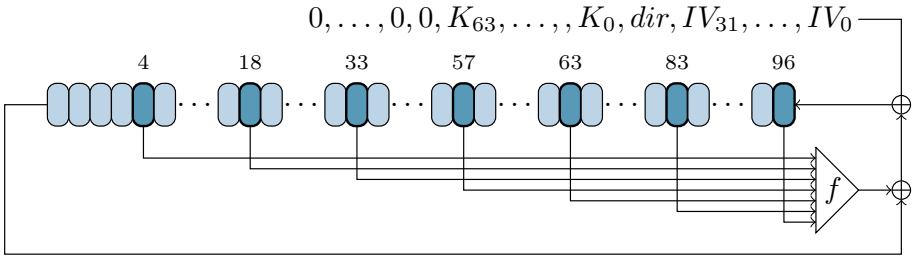

Figure 4.6 – Initialization of register W

The content of W is now taken as a 97-bit string $w = w_0, \ldots, w_{96}$, and inserted in A, B, C and D in much the same way as with GEA-1. The four registers start from the all-zero state, and are filled with the bits of w in the same way as shown in Figure 4.5. The offsets of where in the sequence w each register starts is different than for GEA-1. Register D inserts the bits of w in the natural order $w_0, \ldots, w_{96}$, whereas the registers A, B and C start with bits w_{16}, w_{33} and w_{51}, respectively. Again, if any of the registers happens to end up in the all-zero state after initialization, the bit in position 0 is hard-coded to 1 before key generation starts.

4.3.2 An Attack on GEA-1

First we recall some basic facts about LFSRs in Galois mode, as depicted in Figure 4.7. For further reading we refer to ([Sch96, p. 378 ff.],[HK04, p. 227]).

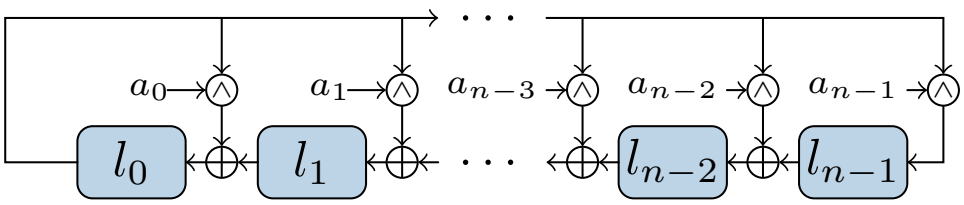

Figure 4.7 – An LFSR in Galois mode.

Given an LFSR L in Galois mode of length n with entries in $\mathbb{F}_2$, clocking the inner state $l = l_0, \ldots, l_{n-1}$ is equivalent to the matrix-vector multiplication

$$G_L \cdot l \coloneqq \begin{pmatrix} a_0 & 1 & 0 & \ldots & 0 \\ a_1 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-2} & 0 & 0 & \ldots & 1 \\ a_{n-1} & 0 & 0 & \ldots & 0 \end{pmatrix} \cdot \begin{pmatrix} l_0 \\ l_1 \\ \vdots \\ l_{n-2} \\ l_{n-1} \end{pmatrix} = \begin{pmatrix} a_0 l_0 + l_1 \\ a_1 l_0 + l_2 \\ \vdots \\ a_{n-2} l_0 + l_{n-1} \\ a_{n-1} l_0 \end{pmatrix}$$

and the characteristic polynomial of G_L is

$$g(X) \coloneqq X^n + a_0 X^{n-1} + \cdots + a_{n-2} X + a_{n-1} \, .$$

Throughout this work, we consider the case in which g is primitive. The characteristic polynomial $g(X)$ is equal to the minimal polynomial of G_L if and only if $a_{n-1} = 1$. Vice versa, given a primitive polynomial $g(X) \coloneqq X^n + a_0 X^{n-1} + \cdots + a_{n-2} X + a_{n-1}$, then

$$G_L \coloneqq \begin{pmatrix} a_0 & 1 & 0 & \ldots & 0 \\ a_1 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-2} & 0 & 0 & \ldots & 1 \\ a_{n-1} & 0 & 0 & \ldots & 0 \end{pmatrix}$$

is the companion matrix of an LFSR in Galois mode with minimal polynomial g. We

call such a matrix the *Galois matrix* and the corresponding minimal polynomial the *Galois polynomial* in the sequel. Moreover, given an LFSR L in Galois mode with minimal (primitive) polynomial g, we denote the Galois matrix by G_g. In the case of GEA-1 the Galois polynomials are

$$\begin{aligned} g_A(X) = &X^{31} + X^{30} + X^{28} + X^{27} + X^{23} + X^{22} + X^{21} + X^{19} + X^{18} + X^{15} \\ &+ X^{11} + X^{10} + X^8 + X^7 + X^6 + X^4 + X^3 + X^2 + 1 \,, \\ g_B(X) = &X^{32} + X^{31} + X^{29} + X^{25} + X^{19} + X^{18} + X^{17} + X^{16} + X^9 + X^8 \\ &+ X^7 + X^3 + X^2 + X + 1 \,, \\ g_C(X) = &X^{33} + X^{30} + X^{27} + X^{23} + X^{21} + X^{20} + X^{19} + X^{18} + X^{17} + X^{15} \\ &+ X^{14} + X^{11} + X^{10} + X^9 + X^4 + X^2 + 1 \,. \end{aligned}$$

The initialization process of the registers A, B and C with the string s is obviously linear. Hence there exist three matrices $M_A \in \mathbb{F}_2^{31\times 64}$, $M_B \in \mathbb{F}_2^{32\times 64}$ and $M_C \in \mathbb{F}_2^{33\times 64}$ such that

$$\begin{aligned} \alpha &= M_A s \,, \\ \beta &= M_B s \,, \\ \gamma &= M_C s \,, \end{aligned}$$

where α, β and γ denote the states of the three LFSRs after the initialization phase. We exclude here the unlikely case that α, β or γ is still in the all-zero state after the shifted insertion of s.

We are now interested in the number of possible starting states of the registers after this initialization. The first observation is that all the three matrices have full rank. This implies that the number of possible starting states after initialization is maximal when each LFSR is considered independently, i.e. there are 2^{31} possible states for register A, 2^{32} possible states for register B, and 2^{33} possible states for register C, as should be expected. However, when considering pairs of registers, the picture changes drastically. In particular, the number of possible joint states after initialization of the registers A and C is much smaller than expected. For this it is convenient to consider the kernels of the linear mappings. Clearly, the corresponding linear mappings represented by M_A, M_B and M_C have kernels of dimension of at least 33, 32 and 31, respectively. If we denote $T_{AC} \coloneqq \ker(M_A) \cap \ker(M_C)$ and $U_B \coloneqq \ker(M_B)$ then, curiously enough, we have

1. $\dim(T_{AC}) = 24$ and $\dim(U_B) = 32$,

2. $U_B \cap T_{AC} = \{0\}$.

From this it directly follows that $\mathbb{F}_2^{64}$ can be decomposed into the direct sum $U_B \oplus T_{AC} \oplus V$, where V is of dimension 8. Thus, for the key-dependent and secret string s, there exists a *unique* representation $s = u + t + v$ with $u \in U_B$, $t \in T_{AC}$, $v \in V$ and

$$\begin{aligned}\beta &= M_B(u + t + v) = M_B(t + v)\\ \alpha &= M_A(u + t + v) = M_A(u + v)\\ \gamma &= M_C(u + t + v) = M_C(u + v)\ .\end{aligned}$$

From this decomposition, s can be computed with a Divide-and-Conquer attack with a complexity [5] of 2^{37} `GEA-1` evaluations to build (and sort) 2^8 tables with 2^{24} entries of size 89 bits and a brute-force step of complexity 2^{40} `GEA-1` evaluations for each new session key $K_0, \ldots, K_{63}$. In other words, the joint state of A and C can be described with only 40 bits and thus can take only 2^{40} possible values. This is the key observation of the attack and such weakness is highly unlikely to occur unintentionally.

Since `GEA-1` was designed to be exportable within the export restrictions in European countries in the late 1990s, this might be an indication that a security level of 40 bits was a barrier for cryptographic algorithms to obtain the necessary authorizations. Ultimately, the weak design of `GEA-1` brings security problems for today's communication, even if it is not being actively used by the operators.

4.3.3 An Attack on `GEA-2`

`GEA-2` does not suffer from the same problems as `GEA-1` for initialization. However, it is still possible to mount an attack on `GEA-2` that does not target initialization, but keystream generation.

The algebraic degree of the filtering function f is 4. The filtering function also has an algebraic immunity of 4. But, as the 4 registers are never mixed, the number of monomials present in the system of equations formed by the relations between the keystream and the initial state is very limited. More precisely, this number is upper bounded by

$$1 + \sum_{i=1}^{4} \binom{29}{i} + \binom{31}{i} + \binom{32}{i} + \binom{33}{i} = 152682\,.$$

5. The complexity will be measured by the amount of operations that are roughly as complex as `GEA-1` evaluations (for generating a keystream of size ≤ 128 bit).

This relatively small number would directly imply a powerful attack, just by using a linearisation technique, or, even more powerful, by applying the Berlekamp-Massey algorithm [Ber68; Mas69], as this value is naturally an upper bound to the linear complexity of the output sequence (a direct consequence of Blahut's Theorem [Bla83]).

However, each session in GEA-2 (or GEA-1) is limited to 1600 bytes, that is 12800 bits. This data limitation frustrates direct algebraic cryptanalysis, as the linearization technique is impossible when we have less equations than monomials.

Hopefully, combining a guess-and-determine algorithm, the linearisation technique as well as a clever organization of computations, we were able to fully break GEA-2. Our attack requires approximately 1468 consecutive keystream bits to be faster than an exhaustive search of the key and reaches a complexity equivalent to $2^{45.1}$ GEA-2 encryption if the 1600 keystream bytes of one session are all known.

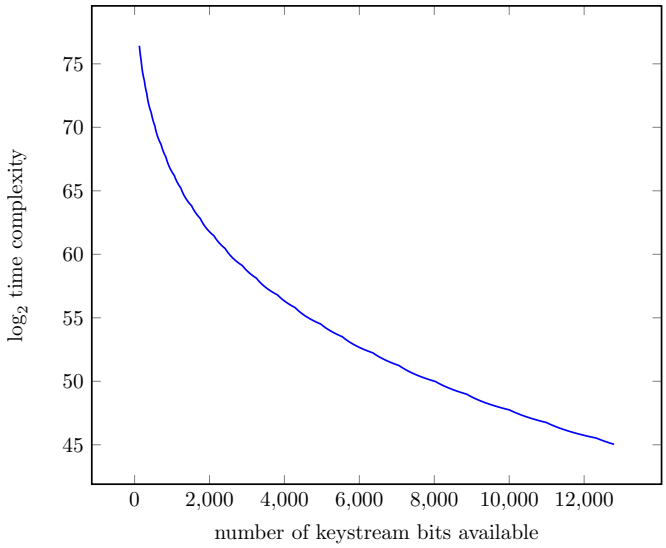

Figure 4.8 – Time complexity of our attack against GEA-2 as a function of the number of consecutive keystream bits available.

Note that both those attacks have recently been improved by Amzaleg and Dinur in [AD22].

List of Publications

Journal Articles

1. Mathilde Badoual, Patrick Derbez, Marine Aubert, Basile Grammaticos: **Simulating the migration and growth patterns of Bacillus subtilis**. Physica A: Statistical Mechanics and its Applications 388-4 (2009)
2. Charles Bouillaguet, Patrick Derbez, Orr Dunkelman, Pierre-Alain Fouque, Nathan Keller, Vincent Rijmen: **Low-Data Complexity Attacks on AES**. IEEE Transactions on Information Theory 58-11 (2012)
3. Patrick Derbez, Tetsu Iwata, Ling Sun, Siwei Sun, Yosuke Todo, Haoyang Wang, Meiqin Wang: **Cryptanalysis of AES-PRF and Its Dual**. ToSC 2018-2
4. Patrick Derbez, Pierre-Alain Fouque, Baptiste Lambin, Brice Minaud: **On Recovering Affine Encodings in White-Box Implementations**. TCHES 2018-3
5. Brice Minaud, Patrick Derbez, Pierre-Alain Fouque, Pierre Karpman: **Key-Recovery Attacks on ASASA**. Journal of Cryptology 31-3 (2018)
6. Patrick Derbez, Pierre-Alain Fouque, Baptiste Lambin, Victor Mollimard: **Efficient Search for Optimal Diffusion Layers of Generalized Feistel Networks**. ToSC 2019-2
7. Patrick Derbez, Pierre-Alain Fouque: **Increasing Precision of Division Property**. ToSC 2020-4
8. Stéphanie Delaune, Patrick Derbez, Mathieu Vavrille: **Catching the Fastest Boomerangs Application to SKINNY**. ToSC 2020-4
9. Patrick Derbez, Pierre-Alain Fouque, Victor Mollimard: **Fake Near Collisions Attacks**. ToSC 2020-4
10. Patrick Derbez, Léo Perrin: **Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE**. Journal of Cryptology 33-3 (2020)
11. Stefan Kölbl, Elmar Tischhauser, Patrick Derbez, Andrey Bogdanov: **Troika: a ternary cryptographic hash function**. DCC 88-1 (2020)

12. Baptiste Lambin, Patrick Derbez, Pierre-Alain Fouque: **Linearly equivalent S-boxes and the division property**. DCC 88-10 (2020)

Conference and Workshop Papers

1. Charles Bouillaguet, Patrick Derbez, Pierre-Alain Fouque: **Automatic Search of Attacks on Round-Reduced AES and Applications**. CRYPTO 2011
2. Patrick Derbez, Pierre-Alain Fouque, Delphine Leresteux: **Meet-in-the-Middle and Impossible Differential Fault Analysis on AES**. CHES 2011
3. Patrick Derbez, Pierre-Alain Fouque, Jérémy Jean: **Faster Chosen-Key Distinguishers on Reduced-Round AES**. INDOCRYPT 2012
4. Patrick Derbez, Pierre-Alain Fouque: **Exhausting Demirci-Selçuk Meet-in-the-Middle Attacks Against Reduced-Round AES**. FSE 2013
5. Patrick Derbez, Pierre-Alain Fouque, Jérémy Jean: **Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting**. EUROCRYPT 2013
6. Patrick Derbez, Léo Perrin: **Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE**. FSE 2015
7. Alex Biryukov, Patrick Derbez, Léo Perrin: **Differential Analysis and Meet-in-the-Middle Attack Against Round-Reduced TWINE**. FSE 2015
8. Brice Minaud, Patrick Derbez, Pierre-Alain Fouque, Pierre Karpman: **Key-Recovery Attacks on ASASA**. ASIACRYPT 2015
9. Patrick Derbez: **Note on Impossible Differential Attacks**. FSE 2016
10. Patrick Derbez, Pierre-Alain Fouque: **Automatic Search of Meet-in-the-Middle and Impossible Differential Attacks**. CRYPTO 2016
11. Patrick Derbez, Pierre-Alain Fouque, Jérémy Jean, Baptiste Lambin: **Variants of the AES Key Schedule for Better Truncated Differential Bounds**. SAC 2018
12. Danping Shi, Siwei Sun, Patrick Derbez, Yosuke Todo, Bing Sun, Lei Hu: **Programming the Demirci-Selçuk Meet-in-the-Middle Attack with Constraints**. ASIACRYPT 2018

13. Patrick Derbez, Virginie Lallemand, Aleksei Udovenko: **Cryptanalysis of SKINNY in the Framework of the SKINNY 2018-2019 Cryptanalysis Competition**. SAC 2019
14. Patrick Derbez, Paul Huynh, Virginie Lallemand, María Naya-Plasencia, Léo Perrin, André Schrottenloher: **Cryptanalysis Results on Spook - Bringing Full-Round Shadow-512 to the Light**. CRYPTO 2020
15. Stéphanie Delaune, Patrick Derbez, Arthur Gontier, Charles Prud'homme: **A Simpler Model for Recovering Superpoly on Trivium**. SAC 2021
16. Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent, Håvard Raddum, Yann Rotella, David Rupprecht, Lukas Stennes: **Cryptanalysis of the GPRS Encryption Algorithms GEA-1 and GEA-2**. EUROCRYPT 2021
17. Stéphanie Delaune, Patrick Derbez, Paul Huynh, Marine Minier, Victor Mollimard, Charles Prud'homme: **Efficient Methods to Search for Best Differential Characteristics on SKINNY**. ACNS 2021

BIBLIOGRAPHY

[AD22] Dor Amzaleg and Itai Dinur, *Refined Cryptanalysis of the GPRS Ciphers GEA-1 and GEA-2*, Cryptology ePrint Archive, Paper 2022/424, `https://eprint.iacr.org/2022/424`, 2022.

[Ank+17] Ralph Ankele, Subhadeep Banik, Avik Chakraborti, Eik List, Florian Mendel, Siang Meng Sim, and Gaoli Wang, « Related-Key Impossible-Differential Attack on Reduced-Round Skinny », *in*: *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, ed. by Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, vol. 10355, Lecture Notes in Computer Science, Springer, 2017, pp. 208–228.

[Aok+00] Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita, « Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis », *in*: *Selected Areas in Cryptography, 7th Annual International Workshop, SAC 2000, Waterloo, Ontario, Canada, August 14-15, 2000, Proceedings*, ed. by Douglas R. Stinson and Stafford E. Tavares, vol. 2012, Lecture Notes in Computer Science, Springer, 2000, pp. 39–56.

[Ban+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni, « Midori: A Block Cipher for Low Energy », *in*: *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, ed. by Tetsu Iwata and Jung Hee Cheon, vol. 9453, Lecture Notes in Computer Science, Springer, 2015, pp. 411–436.

[Bar+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang, « On the (Im)possibility of Obfuscating Programs », *in*: *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August*

19-23, 2001, Proceedings, ed. by Joe Kilian, vol. 2139, Lecture Notes in Computer Science, Springer, 2001, pp. 1–18.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir, « Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials », *in*: *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, ed. by Jacques Stern, vol. 1592, Lecture Notes in Computer Science, Springer, 1999, pp. 12–23.

[BCH16] Chung Hun Baek, Jung Hee Cheon, and Hyunsook Hong, « White-box AES implementation revisited », *in*: *Journal of Communications and Networks* 18.*3* (2016), pp. 273–287.

[BDF11] Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque, « Automatic Search of Attacks on Round-Reduced AES and Applications », *in*: *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, ed. by Phillip Rogaway, vol. 6841, Lecture Notes in Computer Science, Springer, 2011, pp. 169–187.

[BDP15] Alex Biryukov, Patrick Derbez, and Léo Perrin, « Differential Analysis and Meet-in-the-Middle Attack Against Round-Reduced TWINE », *in*: *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, ed. by Gregor Leander, vol. 9054, Lecture Notes in Computer Science, Springer, 2015, pp. 3–27.

[Bea+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers, « The SIMON and SPECK Families of Lightweight Block Ciphers », *in*: *IACR Cryptology ePrint Archive* 2013 (2013), p. 404.

[Bei+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim, « The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS », *in*: *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, ed. by Matthew Robshaw and Jonathan Katz, vol. 9815, Lecture Notes in Computer Science, Springer, 2016, pp. 123–153.

[Bei+21] Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent, Håvard Raddum, Yann Rotella, David Rupprecht, and Lukas Stennes, « Cryptanalysis of the GPRS Encryption Algorithms GEA-1 and GEA-2 », *in*: *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*, ed. by Anne Canteaut and François-Xavier Standaert, vol. 12697, Lecture Notes in Computer Science, Springer, 2021, pp. 155–183.

[Ber68] Elwyn R. Berlekamp, *Algebraic coding theory*, McGraw-Hill series in systems science, McGraw-Hill, 1968, ISBN: 0070049033.

[BGE04] Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi, « Cryptanalysis of a White Box AES Implementation », *in*: *Selected Areas in Cryptography, 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers*, ed. by Helena Handschuh and M. Anwar Hasan, vol. 3357, Lecture Notes in Computer Science, Springer, 2004, pp. 227–240.

[Bih+15] Eli Biham, Orr Dunkelman, Nathan Keller, and Adi Shamir, « New Attacks on IDEA with at Least 6 Rounds », *in*: *J. Cryptology* 28.*2* (2015), pp. 209–239.

[Bir+03] Alex Biryukov, Christophe De Cannière, An Braeken, and Bart Preneel, « A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms », *in*: *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, ed. by Eli Biham, vol. 2656, Lecture Notes in Computer Science, Springer, 2003, pp. 33–50.

[BK09] Alex Biryukov and Dmitry Khovratovich, « Related-Key Cryptanalysis of the Full AES-192 and AES-256 », *in*: *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, ed. by Mitsuru Matsui, vol. 5912, Lecture Notes in Computer Science, Springer, 2009, pp. 1–18.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic, « Distinguisher and Related-Key Attack on the Full AES-256 », *in*: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa*

Barbara, CA, USA, August 16-20, 2009. Proceedings, ed. by Shai Halevi, vol. 5677, Lecture Notes in Computer Science, Springer, 2009, pp. 231–249.

[Bla83] Richard E Blahut, *Theory and practice of error control codes*, Addison-Wesley, 1983.

[BN10] Alex Biryukov and Ivica Nikolic, « Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others », *in*: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, ed. by Henri Gilbert, vol. 6110, Lecture Notes in Computer Science, Springer, 2010, pp. 322–344.

[BNS14] Christina Boura, Marıa Naya-Plasencia, and Valentin Suder, « Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon », *in*: *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, ed. by Palash Sarkar and Tetsu Iwata, vol. 8873, Lecture Notes in Computer Science, Springer, 2014, pp. 179–199.

[BS01] Alex Biryukov and Adi Shamir, « Structural Cryptanalysis of SASAS », *in*: *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, ed. by Birgit Pfitzmann, vol. 2045, Lecture Notes in Computer Science, Springer, 2001, pp. 394–405.

[BS90] Eli Biham and Adi Shamir, « Differential Cryptanalysis of DES-like Cryptosystems », *in*: *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, ed. by Alfred Menezes and Scott A. Vanstone, vol. 537, Lecture Notes in Computer Science, Springer, 1990, pp. 2–21.

[CGT19] Victor Cauchois, Clément Gomez, and Gaël Thomas, « General Diffusion Analysis: How to Find Optimal Permutations for Generalized Type-II Feistel Schemes », *in*: *IACR Trans. Symmetric Cryptol.* 2019.*1* (2019), pp. 264–301.

[Cho+02a] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot, « A White-Box DES Implementation for DRM Applications », *in*: *Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002, Washington, DC, USA, November 18, 2002, Revised Papers*, ed. by Joan Feigenbaum, vol. 2696, Lecture Notes in Computer Science, Springer, 2002, pp. 1–15.

[Cho+02b] Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot, « White-Box Cryptography and an AES Implementation », *in*: *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, ed. by Kaisa Nyberg and Howard M. Heys, vol. 2595, Lecture Notes in Computer Science, Springer, 2002, pp. 250–270.

[Cho+11] Jiali Choy, Aileen Zhang, Khoongming Khoo, Matt Henricksen, and Axel Poschmann, « AES Variants Secure against Related-Key Differential and Boomerang Attacks », *in*: *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication - 5th IFIP WG 11.2 International Workshop, WISTP 2011, Heraklion, Crete, Greece, June 1-3, 2011. Proceedings*, ed. by Claudio A. Ardagna and Jianying Zhou, vol. 6633, Lecture Notes in Computer Science, Springer, 2011, pp. 191–207.

[Cid+17] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song, « A Security Analysis of Deoxys and its Internal Tweakable Block Ciphers », *in*: *IACR Trans. Symmetric Cryptol.* 2017.*3* (2017), pp. 73–107.

[Cid+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song, « Boomerang Connectivity Table: A New Cryptanalysis Tool », *in*: *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, ed. by Jesper Buus Nielsen and Vincent Rijmen, vol. 10821, Lecture Notes in Computer Science, Springer, 2018, pp. 683–714.

[DDV20] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille, « Catching the Fastest Boomerangs Application to SKINNY », *in*: *IACR Trans. Symmetric Cryptol.* 2020.*4* (2020), pp. 104–129.

[Del+13] Cécile Delerablée, Tancrède Lepoint, Pascal Paillier, and Matthieu Rivain, « White-Box Security Notions for Symmetric Encryption Schemes », *in*: *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, ed. by Tanja Lange, Kristin E. Lauter, and Petr Lisonek, vol. 8282, Lecture Notes in Computer Science, Springer, 2013, pp. 247–264.

[Der+18a] Patrick Derbez, Pierre-Alain Fouque, Jérémy Jean, and Baptiste Lambin, « Variants of the AES Key Schedule for Better Truncated Differential Bounds », *in*: *Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers*, ed. by Carlos Cid and Michael J. Jacobson Jr., vol. 11349, Lecture Notes in Computer Science, Springer, 2018, pp. 27–49.

[Der+18b] Patrick Derbez, Pierre-Alain Fouque, Baptiste Lambin, and Brice Minaud, « On Recovering Affine Encodings in White-Box Implementations », *in*: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.*3* (2018), pp. 121–149.

[Der+18c] Patrick Derbez, Tetsu Iwata, Ling Sun, Siwei Sun, Yosuke Todo, Haoyang Wang, and Meiqin Wang, « Cryptanalysis of AES-PRF and Its Dual », *in*: *IACR Trans. Symmetric Cryptol.* 2018.*2* (2018), pp. 161–191.

[Der+19] Patrick Derbez, Pierre-Alain Fouque, Baptiste Lambin, and Victor Mollimard, « Efficient Search for Optimal Diffusion Layers of Generalized Feistel Networks », *in*: *IACR Trans. Symmetric Cryptol.* 2019.*2* (2019), pp. 218–240.

[DES77] DES, « Data Encryption Standard », *in*: *FIPS PUB 46, Federal information processing standards publication 46* (1977).

[DF13] Patrick Derbez and Pierre-Alain Fouque, « Exhausting Demirci-Selçuk Meet-in-the-Middle Attacks Against Reduced-Round AES », *in*: *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, ed. by Shiho Moriai, vol. 8424, Lecture Notes in Computer Science, Springer, 2013, pp. 541–560.

[DF16] Patrick Derbez and Pierre-Alain Fouque, « Automatic Search of Meet-in-the-Middle and Impossible Differential Attacks », *in*: *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference,*

Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, ed. by Matthew Robshaw and Jonathan Katz, vol. 9815, Lecture Notes in Computer Science, Springer, 2016, pp. 157–184.

[DF20] Patrick Derbez and Pierre-Alain Fouque, « Increasing Precision of Division Property », *in*: *IACR Trans. Symmetric Cryptol.* 2020.*4* (2020), pp. 173–194.

[DFJ13] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean, « Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting », *in*: *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, ed. by Thomas Johansson and Phong Q. Nguyen, vol. 7881, Lecture Notes in Computer Science, Springer, 2013, pp. 371–387.

[Din18] Itai Dinur, « An Improved Affine Equivalence Algorithm for Random Permutations », *in*: *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, ed. by Jesper Buus Nielsen and Vincent Rijmen, vol. 10820, Lecture Notes in Computer Science, Springer, 2018, pp. 413–442.

[DKK06] Orr Dunkelman, Nathan Keller, and Jongsung Kim, « Related-Key Rectangle Attack on the Full SHACAL-1 », *in*: *Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers*, ed. by Eli Biham and Amr M. Youssef, vol. 4356, Lecture Notes in Computer Science, Springer, 2006, pp. 28–44.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen, « The Block Cipher Square », *in*: *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, ed. by Eli Biham, vol. 1267, Lecture Notes in Computer Science, Springer, 1997, pp. 149–165.

[DKS14] Orr Dunkelman, Nathan Keller, and Adi Shamir, « A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony », *in*: *J. Cryptology* 27.*4* (2014), pp. 824–849.

[DLU19] Patrick Derbez, Virginie Lallemand, and Aleksei Udovenko, « Cryptanalysis of SKINNY in the Framework of the SKINNY 2018-2019 Cryptanalysis Competition », *in*: *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, ed. by Kenneth G. Paterson and Douglas Stebila, vol. 11959, Lecture Notes in Computer Science, Springer, 2019, pp. 124–145.

[DP15] Patrick Derbez and Léo Perrin, « Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE », *in*: *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, ed. by Gregor Leander, vol. 9054, Lecture Notes in Computer Science, Springer, 2015, pp. 190–216.

[DP20] Patrick Derbez and Léo Perrin, « Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE », *in*: *J. Cryptol.* 33.*3* (2020), pp. 1184–1215.

[DS08] Hüseyin Demirci and Ali Aydin Selçuk, « A Meet-in-the-Middle Attack on 8-Round AES », *in*: *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, ed. by Kaisa Nyberg, vol. 5086, Lecture Notes in Computer Science, Springer, 2008, pp. 116–126.

[Esk+18] Zahra Eskandari, Andreas Brasen Kidmose, Stefan Kölbl, and Tyge Tiessen, « Finding Integral Distinguishers with Ease », *in*: *Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers*, ed. by Carlos Cid and Michael J. Jacobson Jr., vol. 11349, Lecture Notes in Computer Science, Springer, 2018, pp. 115–138.

[ETS18] ETSI, *Digital cellular telecommunications system (Phase 2+) (GSM); Security related network functions (3GPP TS 43.020 version 15.0.0 Release 15)*, Technical Specification. Available at `https://www.etsi.org/deliver/etsi_ts/143000_143099/143020/15.00.00_60/ts_143020v150000p.pdf` (accessed October 8, 2020), 2018.

[Fer+00] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David A. Wagner, and Doug Whiting, « Improved Cryptanalysis of Rijndael », *in*: *Fast Software Encryption, 7th International Workshop, FSE 2000,*

New York, NY, USA, April 10-12, 2000, Proceedings, ed. by Bruce Schneier, vol. 1978, Lecture Notes in Computer Science, Springer, 2000, pp. 213–230.

[FJP13] Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin, « Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128 », *in*: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, ed. by Ran Canetti and Juan A. Garay, vol. 8042, Lecture Notes in Computer Science, Springer, 2013, pp. 183–203.

[Fou+16] Pierre-Alain Fouque, Pierre Karpman, Paul Kirchner, and Brice Minaud, « Efficient and Provable White-Box Primitives », *in*: *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, ed. by Jung Hee Cheon and Tsuyoshi Takagi, vol. 10031, Lecture Notes in Computer Science, 2016, pp. 159–188.

[Gér+18] David Gérault, Pascal Lafourcade, Marine Minier, and Christine Solnon, « Revisiting AES related-key differential attacks with constraint programming », *in*: *Inf. Process. Lett.* 139 (2018), pp. 24–29.

[GP10] Henri Gilbert and Thomas Peyrin, « Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations », *in*: *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, ed. by Seokhie Hong and Tetsu Iwata, vol. 6147, Lecture Notes in Computer Science, Springer, 2010, pp. 365–383.

[Hao+20] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang, « Modeling for Three-Subset Division Property Without Unknown Subset - Improved Cube Attacks Against Trivium and Grain-128AEAD », *in*: *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, ed. by Anne Canteaut and Yuval Ishai, vol. 12105, Lecture Notes in Computer Science, Springer, 2020, pp. 466–495.

[Heb+20] Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo, « Lower Bounds on the Degree of Block Ciphers », *in*: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 1051.

[HK04] Kenneth Hoffman and Ray A. Kunze, *Linear Algebra*, PHI Learning, 2004, ISBN: 8120302702.

[Hon+06] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee, « HIGHT: A New Block Cipher Suitable for Low-Resource Device », *in*: *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, ed. by Louis Goubin and Mitsuru Matsui, vol. 4249, Lecture Notes in Computer Science, Springer, 2006, pp. 46–59.

[JBF02] M. Jacob, D. Boneh, and E. Felten, *Attacking an obfuscated cipher by injecting faults*, ACM Workshop on Digital Rights Management, 2002.

[JN16] Jérémy Jean and Ivica Nikolic, « Efficient Design Strategies Based on the AES Round Function », *in*: *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, ed. by Thomas Peyrin, vol. 9783, Lecture Notes in Computer Science, Springer, 2016, pp. 334–353.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin, « Tweaks and Keys for Block Ciphers: The TWEAKEY Framework », *in*: *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, ed. by Palash Sarkar and Tetsu Iwata, vol. 8874, Lecture Notes in Computer Science, Springer, 2014, pp. 274–288.

[Kar10] Mohamed Karroumi, « Protecting White-Box AES with Dual Ciphers », *in*: *Information Security and Cryptology - ICISC 2010 - 13th International Conference, Seoul, Korea, December 1-3, 2010, Revised Selected Papers*, ed. by Kyung Hyune Rhee and DaeHun Nyang, vol. 6829, Lecture Notes in Computer Science, Springer, 2010, pp. 278–291.

[Kho+17] Khoongming Khoo, Eugene Lee, Thomas Peyrin, and Siang Meng Sim, « Human-readable Proof of the Related-Key Security of AES-128 », *in*: *IACR Trans. Symmetric Cryptol.* 2017.*2* (2017), pp. 59–83.

[Kir15] Aleksandar Kircanski, « Analysis of Boomerang Differential Trails via a SAT-Based Constraint Solver URSA », *in*: *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers*, ed. by Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, vol. 9092, Lecture Notes in Computer Science, Springer, 2015, pp. 331–349.

[Knu98] Lars Knudsen, « DEAL-a 128-bit block cipher », *in*: (1998).

[KW02] Lars R. Knudsen and David A. Wagner, « Integral Cryptanalysis », *in*: *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, ed. by Joan Daemen and Vincent Rijmen, vol. 2365, Lecture Notes in Computer Science, Springer, 2002, pp. 112–127.

[LDF20] Baptiste Lambin, Patrick Derbez, and Pierre-Alain Fouque, « Linearly equivalent S-boxes and the division property », *in*: *Des. Codes Cryptogr.* 88.*10* (2020), pp. 2207–2231.

[Lep+13] Tancrède Lepoint, Matthieu Rivain, Yoni De Mulder, Peter Roelse, and Bart Preneel, « Two Attacks on a White-Box AES Implementation », *in*: *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, ed. by Tanja Lange, Kristin E. Lauter, and Petr Lisonek, vol. 8282, Lecture Notes in Computer Science, Springer, 2013, pp. 265–285.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song, « Security Analysis of SKINNY under Related-Tweakey Settings (Long Paper) », *in*: *IACR Trans. Symmetric Cryptol.* 2017.*3* (2017), pp. 37–72.

[LS19] Yunwen Liu and Yu Sasaki, « Related-Key Boomerang Attacks on GIFT with Automated Trail Search Including BCT Effect », *in*: *Information Security and Privacy - 24th Australasian Conference, ACISP 2019, Christchurch, New Zealand, July 3-5, 2019, Proceedings*, ed. by Julian Jang-Jaccard and Fuchun Guo, vol. 11547, Lecture Notes in Computer Science, Springer, 2019, pp. 555–572.

[Lu09] Jiqiang Lu, « Related-key rectangle attack on 36 rounds of the XTEA block cipher », *in*: *Int. J. Inf. Sec.* 8.*1* (2009), pp. 1–11.

[Mas69] James L. Massey, « Shift-register synthesis and BCH decoding », *in*: *IEEE Trans. Inf. Theory* 15.*1* (1969), pp. 122–127.

[Mat94] Mitsuru Matsui, « On Correlation Between the Order of S-boxes and the Strength of DES », *in*: *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, ed. by Alfredo De Santis, vol. 950, Lecture Notes in Computer Science, Springer, 1994, pp. 366–375.

[Mat97] Mitsuru Matsui, « New Block Encryption Algorithm MISTY », *in*: *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, ed. by Eli Biham, vol. 1267, Lecture Notes in Computer Science, Springer, 1997, pp. 54–68.

[Med] MediaTek, *Test Vector GEA1/2 — MediaTek-HelioX10-Baseband*, `https://github.com/Dude100/MediaTek-HelioX10-Baseband/blob/591772a0d659ef0f7bba1953d18f8fe7c18b11de/(FDD)MT6795.MOLY.LR9.W1423.MD.LWTG.MP.V24/driver/cipher/include/gcu_ut.h`, (accessed March 4, 2021).

[MRP13] Yoni De Mulder, Peter Roelse, and Bart Preneel, « Revisiting the BGE Attack on a White-Box AES Implementation », *in*: *IACR Cryptol. ePrint Arch.* (2013), p. 450.

[Mur11] Sean Murphy, « The Return of the Cryptographic Boomerang », *in*: *IEEE Trans. Inf. Theory* 57.*4* (2011), pp. 2517–2521.

[Nik10] Ivica Nikolic, « Tweaking AES », *in*: *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, ed. by Alex Biryukov, Guang Gong, and Douglas R. Stinson, vol. 6544, Lecture Notes in Computer Science, Springer, 2010, pp. 198–210.

[Nik17] Ivica Nikolic, « How to Use Metaheuristics for Design of Symmetric-Key Primitives », *in*: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, ed. by Tsuyoshi Takagi and Thomas Peyrin, vol. 10626, Lecture Notes in Computer Science, Springer, 2017, pp. 369–391.

[NM11] Karsten Nohl and Luca Melette, *GPRS Intercept: Wardriving your country*, Chaos Communication Camp, 2011. Slides available at `http://events.ccc.de/camp/2011/Fahrplan/attachments/1868_110810.SRLabs-Camp-GRPS_Intercept.pdf` (accessed October 8, 2020). 2011.

[Nyb96] Kaisa Nyberg, « Generalized Feistel Networks », *in*: *Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings*, ed. by Kwangjo Kim and Tsutomu Matsumoto, vol. 1163, Lecture Notes in Computer Science, Springer, 1996, pp. 91–104.

[osm] osmocom, *osmocom — Cellular Network Infrastructure*, `https://osmocom.org/projects/cellular-infrastructure`, (accessed March 4, 2021).

[Sch+98] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson, « Twofish: A 128-bit block cipher », *in*: *NIST AES Proposal* 15 (1998), p. 23.

[Sch96] Bruce Schneier, *Applied cryptography - protocols, algorithms, and source code in C, 2nd Edition*, Wiley, 1996, ISBN: 978-0-471-11709-4.

[Shi+11] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai, « Piccolo: An Ultra-Lightweight Blockcipher », *in*: *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, ed. by Bart Preneel and Tsuyoshi Takagi, vol. 6917, Lecture Notes in Computer Science, Springer, 2011, pp. 342–357.

[Shi+18] Danping Shi, Siwei Sun, Patrick Derbez, Yosuke Todo, Bing Sun, and Lei Hu, « Programming the Demirci-Selçuk Meet-in-the-Middle Attack with Constraints », *in*: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, ed. by Thomas Peyrin and Steven D. Galbraith, vol. 11273, Lecture Notes in Computer Science, Springer, 2018, pp. 3–34.

[SM10] Tomoyasu Suzaki and Kazuhiko Minematsu, « Improving the Generalized Feistel », *in*: *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, ed. by

Seokhie Hong and Tetsu Iwata, vol. 6147, Lecture Notes in Computer Science, Springer, 2010, pp. 19–39.

[SQH19] Ling Song, Xianrui Qin, and Lei Hu, « Boomerang Connectivity Table Revisited. Application to SKINNY and AES », *in*: *IACR Trans. Symmetric Cryptol.* 2019.*1* (2019), pp. 118–141.

[Suz+12] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi, « TWINE : A Lightweight Block Cipher for Multiple Platforms », *in*: *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, ed. by Lars R. Knudsen and Huapeng Wu, vol. 7707, Lecture Notes in Computer Science, Springer, 2012, pp. 339–354.

[TM16] Yosuke Todo and Masakatu Morii, « Bit-Based Division Property and Application to Simon Family », *in*: *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, ed. by Thomas Peyrin, vol. 9783, Lecture Notes in Computer Science, Springer, 2016, pp. 357–377.

[Tod15a] Yosuke Todo, « Integral Cryptanalysis on Full MISTY1 », *in*: *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, ed. by Rosario Gennaro and Matthew Robshaw, vol. 9215, Lecture Notes in Computer Science, Springer, 2015, pp. 413–432.

[Tod15b] Yosuke Todo, « Structural Evaluation by Generalized Integral Property », *in*: *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, ed. by Elisabeth Oswald and Marc Fischlin, vol. 9056, Lecture Notes in Computer Science, Springer, 2015, pp. 287–314.

[Wag99] David A. Wagner, « The Boomerang Attack », *in*: *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, ed. by Lars R. Knudsen, vol. 1636, Lecture Notes in Computer Science, Springer, 1999, pp. 156–170.

[WKD07] Gaoli Wang, Nathan Keller, and Orr Dunkelman, « The Delicate Issues of Addition with Respect to XOR Differences », *in*: *Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers*, ed. by Carlisle M. Adams, Ali Miri, and Michael J. Wiener, vol. 4876, Lecture Notes in Computer Science, Springer, 2007, pp. 212–231.

[Xia+16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin, « Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers », *in*: *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, ed. by Jung Hee Cheon and Tsuyoshi Takagi, vol. 10031, Lecture Notes in Computer Science, 2016, pp. 648–678.

[XL09] Yaying Xiao and Xuejia Lai, « A secure implementation of white-box AES », *in*: *Computer Science and its Applications, 2009. CSA'09. 2nd International Conference on*, IEEE, 2009, pp. 1–6.

[ZD19] Rui Zong and Xiaoyang Dong, « MILP-Aided Related-Tweak/Key Impossible Differential Attack and its Applications to QARMA, Joltik-BC », *in*: *IEEE Access* 7 (2019), pp. 153683–153693.

[ZDY19] Baoyu Zhu, Xiaoyang Dong, and Hongbo Yu, « MILP-Based Differential Attack on Round-Reduced GIFT », *in*: *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, ed. by Mitsuru Matsui, vol. 11405, Lecture Notes in Computer Science, Springer, 2019, pp. 372–390.

[ZMI89] Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai, « On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses », *in*: *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, ed. by Gilles Brassard, vol. 435, Lecture Notes in Computer Science, Springer, 1989, pp. 461–480.

[ZR19] Wenying Zhang and Vincent Rijmen, « Division cryptanalysis of block ciphers with a binary diffusion layer », *in*: *IET Information Security* 13.*2* (2019), pp. 87–95.

UNIVERSITÉ DE RENNES 1

Titre : Outils et Algorithmes pour la Cryptanalyse

Résumé : En cryptographie symétrique, la sécurité des algorithmes de chiffrement et des fonctions de hachage est établi de manière empirique, par la non-découverte d'attaques contre ces primitives. Plus précisément, une primitive symétrique doit résister à toutes les techniques connues de cryptanalyse dans le sens où les propriétés de sécurité qu'elle est censée offrir ne doivent pas être mises en défaut. La principale difficulté est que trouver la façon optimale d'appliquer une technique de cryptanalyse à une primitive est loin d'être un problème trivial. Dans ce manuscrit, je décris plusieurs outils et algorithmes pour résoudre ce problème de manière effective pour plusieurs techniques de cryptanalyse et classes de primitives. Je propose aussi plusieurs algorithmes pour concevoir certains composants internes des fonctions de chiffrements offrant une sécurité optimale contre plusieurs classes d'attaques. Enfin je décris plusieurs attaques pratiques sur des constructions symétriques et en particulier sur les algorithmes de chiffrement utilisés dans les protocoles 2G.

Title: Tools and Algorithms for Cryptanalysis

Abstract: The security of symmetric-key primitives as block ciphers and hash functions is established in an empiric manner, by the non-discovery of any attacks or unexpected behavior. More precisely, a symmetric primitive must be secure against all known cryptanalysis techniques and none of its security claims should be broken. The main difficulty is that finding the optimal settings in which a cryptanalysis technique should be applied against a primitive is far from being trivial. In this thesis, I describe several tools and algorithms to efficiently solve this problem for several cryptanalysis techniques and classes of primitives. I present as well several algorithms to design core components of block ciphers providing optimal resistance against various types of attacks. Finally, I give practical attacks against symmetric primitives, including both the stream ciphers used to protect data in 2G protocols.