Linear Cryptanalysis:
Key Schedules and Tweakable Block Ciphers

Thorsten Kranz, Gregor Leander and Friedrich Wiemer

Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
{thorsten.kranz,gregor.leander,friedrich.wiemer}@rub.de

Abstract. This paper serves as a systematization of knowledge of linear cryptanalysis and provides novel insights in the areas of key schedule design and tweakable block ciphers. We examine in a step by step manner the linear hull theorem in a general and consistent setting. Based on this, we study the influence of the choice of the key scheduling on linear cryptanalysis, a – notoriously difficult – but important subject. Moreover, we investigate how tweakable block ciphers can be analyzed with respect to linear cryptanalysis, a topic that surprisingly has not been scrutinized until now.

Keywords: Linear Cryptanalysis · Key Schedule · Hypothesis of Independent Round Keys · Tweakable Block Cipher

1 Introduction

Block ciphers are among the most important cryptographic primitives. Besides being used for encrypting the major fraction of our sensible data, they are important building blocks in many cryptographic constructions and protocols. Clearly, the security of any concrete block cipher can never be strictly proven, usually not even be reduced to a mathematical problem, i. e. be provable in the sense of provable cryptography. However, the concrete security of well-known ciphers, in particular the AES and its predecessor DES, is very well studied and probably much better scrutinized than many of the mathematical problems on which provable secure schemes are based on.

This been said, there is a clear lack of understanding when it comes to the key schedule part of block ciphers. Let us quickly recall the role of the key schedule algorithm in a block cipher. The key schedule takes as input a master key (in the case of AES-128 this is a 128 bit string) and outputs so-called round keys that are used in each round to mix the current state with the key (most often by simply XORing the round key to the state). In the case of AES-128, the total length of the round keys is $11 \cdot 128 = 1408$ bits, and thus the AES key schedule, as a function, is a mapping from $\{0,1\}^{128}$ to $\{0,1\}^{1408}$.

However, in general, it is not yet clear what properties a good key schedule has to have. There are some general guidelines on what a key schedule should *not* look like. These guidelines are rather basic and ensure mainly that trivial guess-and-determine or meet-in-the-middle attacks are not possible. In a nutshell, it should not be possible to compute large parts of the encryption algorithm, i. e. a large number of rounds in the case of iterated ciphers, without having to know or guess the whole master key. An example of such a trivially bad key schedule is the idea of using two independent (master) keys in order to double the key length, i. e. double encryption.

Similarly, a key schedule should be such that structural attacks (e. g. slide-attacks, symmetries, invariant subspace attacks) are not possible. It is often possible to check, for a given key schedule, if it fulfills this criterion.

Licensed under Creative Commons License CC-BY 4.0.

IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2017, No. 1, pp. 474–505
DOI:10.13154/tosc.v2017.i1.474-505

So while there is an understanding of what a key schedule should provide in terms of structural attacks (and for the key-guessing parts in statistical attacks), the influence of the key schedule on statistical attacks, in particular on linear and differential attacks is to a large extent completely open. In a nutshell, for claiming a cipher secure against linear and differential attacks, one has to demonstrate that the cipher does not possess certain statistical irregularities. In order to be able to do so, it is in many cases necessary to assume that all (round) keys are independently and uniformly chosen. While this is hardly the case for any real cipher, this assumption is on the one hand needed to make the analysis feasible and on the other hand often does not seem problematic as even with the keys not independently and uniformly chosen, most ciphers (experimentally!) do not behave different from the expectation.

Linear cryptanalysis, introduced by Matsui [25], is one of the major statistical attacks on block ciphers. Since its invention in the early 1990s, many extensions and variations have been considered. The most important theoretical investigation is certainly the work of Nyberg [27], where the concept of linear hulls was introduced and the assumption of round-independence needed in Matsui's original approach was clarified. Similar results can be derived by using the concept of correlation matrices, as done by Daemen and Rijmen [17]. A statistical model for estimating the data complexity of various linear attacks is presented in [7]. The concept of the linear hulls in particular shows the key-dependency of the correlation of a given linear approximation. Due to the key-dependency of the distribution, for a complete understanding of the security of a block cipher with respect to a linear attack, one has to understand not only one correlation, but rather the distribution of Fourier coefficients taken over all possible master keys. Only then one is able to estimate the fraction of weak keys, that is keys such that the corresponding correlation is high enough to be exploitable in an attack. Following Nyberg's fundamental theorem, it is possible, at least theoretically, to compute the mean and the variance of this distribution in the case of independent round keys. But it seems hard to derive more information about the distribution. It would be especially interesting to derive bounds on the tails, i. e. the fraction of weak keys. Moreover, even for just estimating the variance in practice, the assumption of independent round keys is crucial [6] (while it is also possible to compute the variance without this assumption, cf. [10], doing so in practice seems to be hard).

Note that the above discussion on the key schedule of course extends, and in some respect becomes even more relevant, when we consider the case of a tweakable block cipher, and discuss how a suitable tweak schedule should be constructed. This analogy is maybe most obvious in the `TWEAKEY` setup [21] where key and tweak are just parts of the same object, but is certainly important for any kind of tweak schedule.

In this paper we aim to systemize the theoretical notions underlying linear cryptanalysis. Furthermore, we take some steps forward to increase our understanding of the influence of key and tweak schedule on the security of a (tweakable) block cipher with respect to linear attacks.

Our Contributions

Systematization of Linear Cryptanalysis

We begin with recapitulating the idea of linear cryptanalysis in Section 2. By doing so, we try to express all terms as Fourier coefficients instead of using correlations, as we feel that this actually gives a more clear picture. This perspective turns out to be especially nice when it comes to the correlation of a linear trail. In many papers on linear cryptanalysis, the correlation of a linear trail is either not well defined (when using the piling-up lemma) or not well motivated (when given as a pure definition). However, in our set-up, the correlation of the linear hull nicely corresponds to a Fourier coefficient. Note that this perspective is implicitly already contained in Nyberg's original paper on the linear hull [27], but we feel that it did not get the attention it deserves.

To support the general understanding, we develop the Fourier coefficient of the linear hull by first considering a generic key-dependent block cipher E_k, then specializing it to a round based structure and finally to the most commonly used key-alternating case. While the corresponding proofs involve only basic techniques, and may have appeared elsewhere, we nevertheless include them in Appendix A, in order to preserve their educational value and to aid researchers unfamiliar with the topic to gain a better understanding. Building on these fundamentals, we then turn to key schedules.

Bizarre Examples: The Tail of the Distributions

First, we start by exploring how the key schedule can influence the distribution of Fourier coefficients. This first part, in Section 3 builds upon the example on Present from [2]. Beside the result of Abdelraheem et al., many papers cover experiments on Present – to name just a few: [8, 10, 14, 20]. As observed in [29] the distribution of the correlation for Present follows closely a normal distribution with mean zero. Moreover, the variance of this distribution fits to what can be expected for independent round keys. The observation in [2] was that, when replacing the key schedule of Present by a key schedule that produces identical round keys in every round, the variance increases significantly. This in particular means that the cipher becomes weaker against linear cryptanalysis, as the fraction of keys that have a large correlation (in absolute terms) increases significantly (cf. Figure 4). However, even so the variance increases, the distribution still follows a normal distribution closely.

By doing extensive experiments with a large set of variants of the Present cipher, we eventually observe many interesting examples of how the key schedule can influence the distribution of Fourier coefficients in a much more dramatic manner. While we show several of those distributions in the appendix, the main interesting conclusion actually follows from an example depicted in Figure 7. Recall from above that one important question is, if we can prove stronger statements about the number of keys with a large absolute Fourier coefficient, beyond what is given by Tchebysheff's general upper bound on any distribution. Now, the example we found leads to a negative conclusion. That is, in general it seems that we cannot hope to prove any stronger statements.

Linear Key Schedule

The next contribution leads to a much more positive, constructive result. Here, in Section 4 we focus on the case of a linear key schedule. Linear key schedules are very common in block ciphers. Besides the DES, many lightweight ciphers actually use the easiest possible linear key schedule, i. e. simply use identical round keys. In order to avoid structural attacks, in particular slide attacks, and in order to break symmetries, it is common sense to add varying round constants to every round key. Now, in Section 4 we prove that any linear key schedule is sound, with respect to linear cryptanalysis, in the following sense: For any given linear key schedule, the average variance of the distribution of the Fourier coefficients, taken over all possible round constants, is exactly the same as for independent round keys. Thus, as a designer, after fixing any linear key schedule of ones choice, one can expect that when adding a randomly chosen set of round constants, the distribution of the Fourier coefficients closely follows the one in the case of independent round keys. This adds some theoretical foundation on the hypothesis of independent round keys criticized above in the case of linear key schedules. We actually back up this theoretical observation by experiments on, guess what, Present.

Tweakable Block Ciphers

Finally, we turn our attention to tweakable block ciphers and how the additional input, i. e. the tweak, possibly helps an attacker. The main possible advantage of the tweak, when it comes to linear cryptanalysis, is that instead of approximating a linear function of the ciphertext by a linear function of the plaintext only, the attacker can now try to approximate (a linear function of) the ciphertext by a linear function of the plaintext *and* a linear function of the tweak.

To study this potentially new attack vector, we elaborate on the linear hull of a tweakable block cipher. We look at the case of a linear tweak schedule and later specialize on tweak-alternating and key-alternating block ciphers. It turns out that the linear hull, and therefore the Fourier coefficient an attacker can use, is actually composed of the same linear trails as in the non-tweaked case. In other words, by adding the tweak, no new linear characteristics are introduced. Thus, protecting a tweakable cipher with linear tweak schedule against linear cryptanalysis basically does not need any additional considerations, but can be done exactly the same way as it is done for non-tweakable block cipher, i. e. by upper bounding the Fourier coefficient of any single linear characteristic. Note that this is in sharp contrast to the differential case, where using a difference in the tweak often leads to differential characteristics with a significantly higher probability.

We like to clearly mention that, from a technical point of view, we mainly reuse standard approaches. Still, our results shed some new light on the wide-open field of the design of a sound key schedule.

2 Systematization of Linear Cryptanalysis

In the course of this section, we develop in a step by step manner the setting of linear cryptanalysis in a general and consistent way. Within our systematization, we also highlight the meaning of a linear trail as this seems to be not well-known.

Let us start by giving some basic notations before we recall the basic concepts of linear cryptanalysis. We denote by $\mathbb{F}_2$ the finite field with two elements and by $\mathbb{F}_2^n$ the n-dimensional vector space over $\mathbb{F}_2$, i. e. the set of all n bit strings together with the bitwise XOR-addition. When dealing with linear cryptanalysis, we need to define a scalar product on $\mathbb{F}_2^n$. For $x, y \in \mathbb{F}_2^n$ by $\langle x, y\rangle$ we denote the canonical scalar product, i. e. $\langle x, y\rangle \coloneqq \sum x_i y_i$. We will often deal with linear mappings on $\mathbb{F}_2^n$ and, given a linear mapping $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$ we denote by L^T its adjoint linear mapping, i. e. the mapping such that

$$\big\langle x, L(y)\big\rangle = \big\langle L^T(x), y\big\rangle \quad \forall x, y \in \mathbb{F}_2^n.$$

Note that, when L is given as an $n \times n$ binary matrix, then L^T is nothing else than the linear mapping corresponding to the transposed matrix.

Linear Cryptanalysis

Next, we recall the basic concepts of linear cryptanalysis. For this, let

$$E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$$

be a block cipher on n bit blocks, indexed by a key k. In classical linear cryptanalysis, we try to approximate a linear Boolean function of the output $E_k(x)$ by a linear Boolean function of the input x. More precisely, we search for a pair of *input and output masks* (α, γ), such that the bias of the linear approximation

$$\langle \gamma, E_k(x)\rangle \approx \langle \alpha, x\rangle$$

is large in absolute terms. We define the bias $\epsilon_{E_k}(\alpha, \gamma)$ by

$$\Pr_x\left[\langle \gamma, E_k(x)\rangle = \langle \alpha, x\rangle\right] = \frac{1}{2} + \epsilon_{E_k}(\alpha, \gamma),$$

and to make linear cryptanalysis successful we have to choose α and γ such that $|\epsilon_{E_k}(\alpha, \gamma)|$ is large since the linear approximation can then be used as a distinguishing property. Due to scaling issues, it is often more convenient to work with the correlation $c_{E_k}(\alpha, \gamma) \coloneqq 2\epsilon_{E_k}(\alpha, \gamma)$ instead of the bias directly.

In this paper, however, we are mainly working with the Fourier (or Walsh) transformation of E_k. The Fourier coefficient of a vectorial Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ at position $\alpha \in \mathbb{F}_2^n$ and $\gamma \in \mathbb{F}_2^m$ is defined as

$$\widehat{f}(\alpha, \gamma) \coloneqq \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \alpha, x\rangle + \langle \gamma, f(x)\rangle}.$$

In terms of linear cryptanalysis, the Fourier coefficient of E_k is nothing else than a scaled version of the bias (and therefore nothing else than a scaled version of the correlation). More precisely it holds that

$$\widehat{E_k}(\alpha, \gamma) = 2^n c_{E_k}(\alpha, \gamma) = 2^{n+1} \epsilon_{E_k}(\alpha, \gamma).$$

As it is usually computationally infeasible to compute the (exact) Fourier coefficient of any reasonable block cipher E_k, we make use of the fact that almost all block ciphers are round based. That is, E_k is then the composition of several (comparably simple) round functions $G_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$. Those round functions are actually key-dependent, but in order to simplify notation, we ignore this key-dependency for now (and come back later to this topic extensively). So instead of computing the exact Fourier coefficient, or correlation, of a linear approximation, one usually focuses on what is called linear trail (synonymously often called linear path or linear characteristic). For an r round cipher

$$E_k(x) = G_{r-1} \circ \cdots \circ G_1 \circ G_0(x)$$

a linear trail θ is a collection of $r+1$ masks

$$\theta = (\theta_0, \theta_1, \ldots, \theta_r)$$

and the correlation of a trail is defined as

$$C_\theta \coloneqq \prod_{i=0}^{r-1} c_{G_i}(\theta_i, \theta_{i+1}). \tag{1}$$

Initially, in his seminal work [25], Matsui derived the correlation of a trail by the so-called piling-up lemma, assuming that the approximations of different rounds behave as independent Boolean random variables. Later, Nyberg [27] showed how this assumption can be avoided by introducing the concept of the linear hull. She also showed that Matsui's famous Algorithm 2, which he used to break DES, was actually making use of the linear hull and not of a single linear trail. This has also nicely been shown for iterated block ciphers by using the technique of correlation matrices [15, 17, 18]. We recall Nyberg's results in terms of the Fourier coefficients of E_k. The first and crucial idea is to consider E_k as a function in two variables, one being the plaintext and the second being the key. For an m bit key k we consider

$$F : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$$

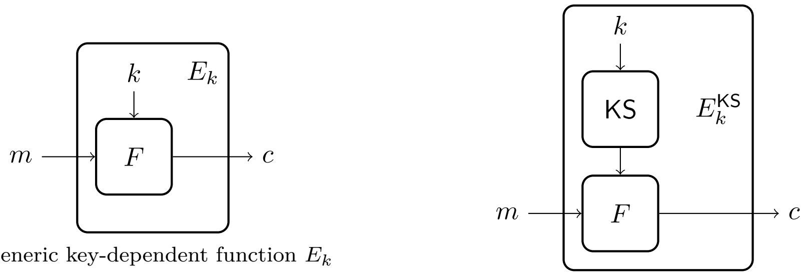
with

$$E_k(x) \coloneqq F(x, k),$$

see also Figure 1(a). Nyberg basically showed that

$$2^m \widehat{E_k}(\alpha, \gamma) = \sum_{\beta \in \mathbb{F}_2^m} (-1)^{\langle \beta, k \rangle} \widehat{F}((\alpha, \beta), \gamma), \tag{2}$$

i. e. the Fourier coefficient of E_k corresponds to the (signed) sum of Fourier coefficients of F over all possible masks for the key-input. This is what is referred to as the *linear hull.* We recall Equation (2) and its key scheduled variant in Proposition 1. In addition to the already mentioned results, Nyberg [26, Theorem 3] also covered this generic influence of a key schedule by the notation of one function having as an input the output of another function.



(a) Generic key-dependent function E_k

(b) and its key scheduled variant E_k^{KS}

Figure 1: Most generic function.

Proposition 1. *Let E_k and E_k^{KS} be the functions (cf. Figures 1(a) and 1(b))*

$$\begin{aligned} E_k &: \mathbb{F}_2^n \to \mathbb{F}_2^n & E_k^{\mathsf{KS}} &: \mathbb{F}_2^n \to \mathbb{F}_2^n \\ E_k(x) &\coloneqq F(x, k) & E_k^{\mathsf{KS}}(x) &\coloneqq E_{\mathsf{KS}(k)}(x) = F(x, \mathsf{KS}(k)) \end{aligned}$$

with $F : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$ and key schedule $\mathsf{KS} : \mathbb{F}_2^\ell \to \mathbb{F}_2^m$. Then

$$\begin{aligned} 2^m \widehat{E_k}(\alpha, \gamma) &= \sum_{\beta \in \mathbb{F}_2^m} (-1)^{\langle \beta, k \rangle} \widehat{F}((\alpha, \beta), \gamma), \\ 2^{\ell+m} \widehat{E_k^{\mathsf{KS}}}(\alpha, \gamma) &= \sum_{\substack{\beta \in \mathbb{F}_2^\ell \\ \beta' \in \mathbb{F}_2^m}} (-1)^{\langle \beta, k \rangle} \widehat{\mathsf{KS}}(\beta, \beta') \widehat{F}((\alpha, \beta'), \gamma). \end{aligned}$$

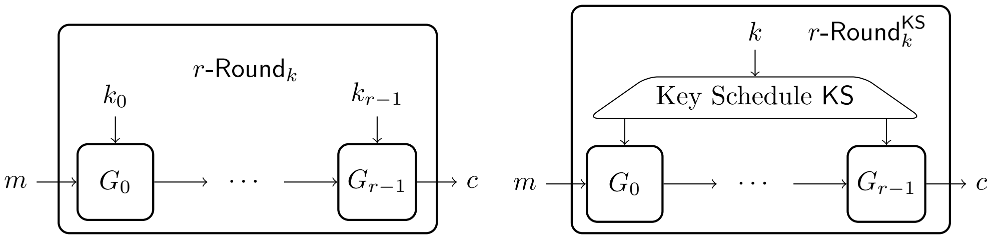
For the proof, refer to Section A.2.

From Equation (2) we can easily deduce the following equation by a simple application of the well-known fact [13, Corollary 2] that the Fourier transform is its own inverse, up to a constant factor.

$$\widehat{F}((\alpha, \beta), \gamma) = \sum_{k \in \mathbb{F}_2^m} (-1)^{\langle \beta, k \rangle} \widehat{E_k}(\alpha, \gamma) \tag{3}$$

Equation (3) might not seem helpful at first sight because it would mean a known-key attack. However, it turns out to be very meaningful in multiple ways. First of all, it will enable us to assert a clear meaning to the definition of a linear trail later in this section. Second, this is already the most basic form of the linear hull theorem for tweakable block ciphers which will be discussed in Section 5 extensively.

Next, we consider the already mentioned case of round-based block ciphers. The linear hull theorem can then be simplified such that the right hand side of the equation only contains Fourier coefficients of the round functions. This specialization of the first proposition has applications for block ciphers that introduce the key material in other ways than simply XORing it onto the state.



(a) Round-based key-dependent function $r\text{-}\mathsf{Round}_k$ (b) and its key scheduled variant $r\text{-}\mathsf{Round}_k^{\mathsf{KS}}$

Figure 2: Round-based functions.

Proposition 2. *Let $r\text{-}\mathsf{Round}_k$ and $r\text{-}\mathsf{Round}_k^{\mathsf{KS}}$ be the functions (cf. Figures 2(a) and 2(b))*

$$r\text{-}\mathsf{Round}_k : \mathbb{F}_2^n \to \mathbb{F}_2^n \qquad\qquad r\text{-}\mathsf{Round}_k^{\mathsf{KS}} : \mathbb{F}_2^n \to \mathbb{F}_2^n$$

$$r\text{-}\mathsf{Round}_k(x) := G_{r-1}(\ldots(G_0(x,k_0),\ldots),k_{r-1}) \qquad r\text{-}\mathsf{Round}_k^{\mathsf{KS}}(x) := r\text{-}\mathsf{Round}_{\mathsf{KS}(k)}(x)$$

with $G_i : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$ and key schedule $\mathsf{KS} : \mathbb{F}_2^\ell \to (\mathbb{F}_2^m)^r$. Then

$$2^{rm+(r-1)n} \widehat{r\text{-}\mathsf{Round}}_k(\alpha,\gamma) = \sum_{\beta \in (\mathbb{F}_2^m)^r} (-1)^{\langle \beta,k \rangle} \sum_{\substack{\theta \in \left(\mathbb{F}_2^n\right)^{r+1} \\ \theta_0=\alpha, \theta_r=\gamma}} \prod_{i=0}^{r-1} \widehat{G_i}((\theta_i,\beta_i),\theta_{i+1}),$$

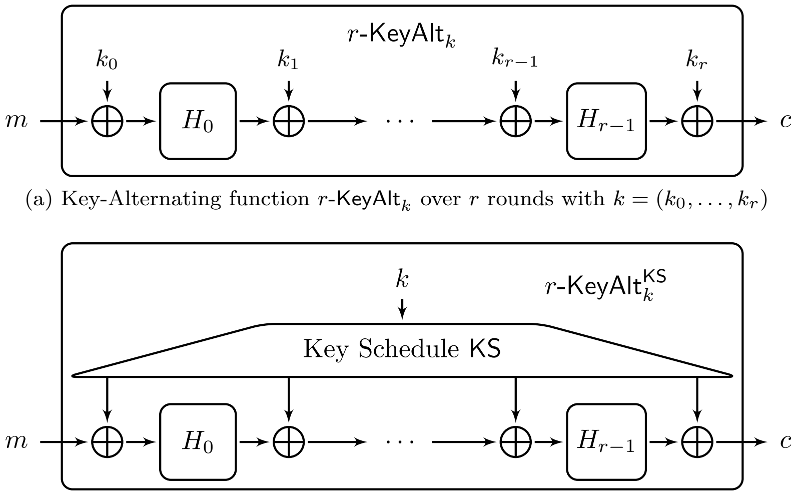
$$2^{\ell+rm+(r-1)n} \widehat{r\text{-}\mathsf{Round}}_k^{\mathsf{KS}}(\alpha,\gamma) = \sum_{\substack{\beta \in \mathbb{F}_2^\ell \\ \beta' \in \left(\mathbb{F}_2^m\right)^r}} (-1)^{\langle \beta,k \rangle} \widehat{\mathsf{KS}}(\beta,\beta') \sum_{\substack{\theta \in \left(\mathbb{F}_2^n\right)^{r+1} \\ \theta_0=\alpha, \theta_r=\gamma}} \prod_{i=0}^{r-1} \widehat{G_i}((\theta_i,\beta'_i),\theta_{i+1}).$$

For the proof, refer to Section A.2.

As this proposition looks a bit puzzling, let us elaborate a bit on it. We only need to pay attention on the rightmost part, the sum over θ and the product over the round functions' Fourier coefficients, as we know the other part already from Proposition 1. So basically $\widehat{r\text{-}\mathsf{Round}}_k$ is the product of the round functions' G_i Fourier coefficients. But instead of having only one possible trail through all round functions, we can choose, after each round, which intermediate mask to use. Eventually we end up with the sum over all possible θ, beginning with α and ending in γ, and thus having a linear hull over the round functions.

Finally, we focus on the case where E_k is round based and the key-dependency is introduced by XORing a key onto the current state in each round, i. e. if E_k is a *key-alternating cipher* as depicted in Figure 3(a). This special case of the linear hull theorem is the most famous one. It is usually cited using the correlation of linear trails as defined in Equation (1).

Another point that is nicely highlighted by this stepwise development via the round based function is the only small difference between Propositions 2 and 3. While we sum over both the key mask β and the round functions input mask θ in the former, the second sum collapses in the latter, as we will see in the next paragraph. This is due to the fact that we cannot say anything about the introduction of key material in a generic round function. But instead, if the key is simply XORed onto the input of the round function, this fixes the corresponding masks $\theta_i = \beta_i$, cf. [4] or [9, Lemma 1].



(a) Key-Alternating function $r\text{-}\mathsf{KeyAlt}_k$ over r rounds with $k = (k_0, \ldots, k_r)$

(b) and its key scheduled variant $r\text{-}\mathsf{KeyAlt}_k^{\mathsf{KS}}$

Figure 3: Key-Alternating functions.

Proposition 3. *Let $r\text{-}\mathsf{KeyAlt}_k$ and $r\text{-}\mathsf{KeyAlt}_k^{\mathsf{KS}}$ be the functions (cf. Figures 3(a) and 3(b))*

$$\begin{aligned} r\text{-}\mathsf{KeyAlt}_k &: \mathbb{F}_2^n \to \mathbb{F}_2^n & r\text{-}\mathsf{KeyAlt}_k^{\mathsf{KS}} &: \mathbb{F}_2^n \to \mathbb{F}_2^n \\ r\text{-}\mathsf{KeyAlt}_k(x) &\coloneqq H_{r-1}(\ldots H_0(x+k_0)+\ldots)+k_r & r\text{-}\mathsf{KeyAlt}_k^{\mathsf{KS}}(x) &\coloneqq r\text{-}\mathsf{KeyAlt}_{\mathsf{KS}(k)} \end{aligned}$$

with $H_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and key schedule $\mathsf{KS} : \mathbb{F}_2^\ell \to (\mathbb{F}_2^n)^{r+1}$. Then

$$\begin{aligned} 2^{(r-1)n} \widehat{r\text{-}\mathsf{KeyAlt}_k}(\alpha,\gamma) &= \sum_{\substack{\beta \in \left(\mathbb{F}_2^n\right)^{r+1} \\ \beta_0=\alpha, \beta_r=\gamma}} (-1)^{\langle \beta, k\rangle} \prod_{i=0}^{r-1} \widehat{H_i}(\beta_i, \beta_{i+1}) \\ &= 2^{rn} \sum_{\substack{\beta \\ \beta_0=\alpha, \beta_r=\gamma}} (-1)^{\langle \beta, k\rangle} C_\beta, \\ 2^{\ell+(r-1)n} \widehat{r\text{-}\mathsf{KeyAlt}_k^{\mathsf{KS}}}(\alpha,\gamma) &= \sum_{\substack{\beta \in \mathbb{F}_2^\ell \\ \beta' \in \left(\mathbb{F}_2^n\right)^{r+1} \\ \beta'_0=\alpha, \beta'_r=\gamma}} (-1)^{\langle \beta, k\rangle} \widehat{\mathsf{KS}}(\beta,\beta') \prod_{i=0}^{r-1} \widehat{H_i}\left(\beta'_i, \beta'_{i+1}\right) \\ &= 2^{rn} \sum_{\substack{\beta, \beta' \\ \beta'_0=\alpha, \beta'_r=\gamma}} (-1)^{\langle \beta, k\rangle} \widehat{\mathsf{KS}}(\beta,\beta') C_{\beta'}. \end{aligned}$$

For the proof, refer to Section A.2.

Furthermore, in the case of a key-alternating cipher with independent round keys, i. e. the case without a key schedule, the following lemma holds:

Lemma 1. *Let $E_k = r\text{-}\mathsf{KeyAlt}_k$ be a* key-alternating cipher*, and F as defined in Proposition 1. Then*

$$2^{-(r+2)n} \widehat{F}((\alpha,\beta),\gamma) = \begin{cases} \prod_{i=0}^{r-1} c_{H_i}(\beta_i, \beta_{i+1}) = C_\beta & \text{, } \textit{if } (\alpha,\gamma) = (\beta_0, \beta_r) \\ 0 & \text{, } \textit{else} \end{cases}.$$

The proof uses Equation (3) and Proposition 3 and can be found in Section A.3.

We like to highlight this fact as we feel it is not well-known, even so it is of course implicitly contained in Nyberg's work, see e. g. [28, Theorem p. 12]: *The correlation of a linear trail is nothing but the Fourier coefficient of* F where $F : \mathbb{F}_2^n \times (\mathbb{F}_2^n)^{(r+1)} \to \mathbb{F}_2^n$ is the key-alternating cipher and the first and last key masks correspond to the message input and message output mask, respectively. Hence, alternatively to Equation (1) we can write

$$C_\theta = 2^{-(r+2)n}\widehat{F}((\theta_0, \theta), \theta_r).$$

This is important to keep in mind as it actually asserts a clear meaning to the correlation of a trail. And it is in contrast to many papers in the literature where either the trail is derived by the piling-up lemma or the correlation of a trail is given directly by using Equation (1) as a definition, as done above to link the usual notation to what we feel is a cleaner way of presenting those connections. Given Lemma 1, one can also easily see the connection of Proposition 1 and 3. Here, all masks that do not start and end in α and γ vanish in the linear hull sum.

Distributions

When applying linear cryptanalysis in practice, we have to compute Fourier coefficients $\widehat{E_k}$ for some fixed key k. But as the Fourier coefficient exhibits a key-dependent behavior, cf. Equation (2), we need to take into account how $\widehat{E_k}$ is distributed over the key space, i. e. what is the probability $\Pr_k\left[\widehat{E_k}(\alpha, \gamma) = X\right]$. In the case of key-alternating block ciphers *with independent round keys*, r-KeyAlt in our notation, the Fourier coefficient follows a normal distribution $\mathcal{N}$ and there are already results about the expected value and the expected squared value, e. g. see [17, pp. 103–108]. Namely,

$$\mathbb{E}\Big(\widehat{r\text{-}\mathsf{KeyAlt}}_k(\alpha, \gamma)\Big) = \frac{1}{2^{(r+1)n}} \sum_{k \in \mathbb{F}_2^{(r+1)n}} \widehat{r\text{-}\mathsf{KeyAlt}}_k(\alpha, \gamma) = 0,$$

and

$$\mathbb{E}\Big(\widehat{r\text{-}\mathsf{KeyAlt}}_k(\alpha, \gamma)^2\Big) = \frac{1}{2^{(r+1)n}} \sum_{k \in \mathbb{F}_2^{(r+1)n}} \widehat{r\text{-}\mathsf{KeyAlt}}_k(\alpha, \gamma)^2 = 2^{2n} \sum_{\substack{\beta \in \left(\mathbb{F}_2^n\right)^{r+1} \\ \beta_0 = \alpha, \beta_r = \gamma}} C_\beta^2.$$

Thus, the mean μ is 0 and the variance $\sigma^2 = 2^{2n} \sum C_\beta^2$.

Daemen and Rijmen [16] did also extensively study the probability distributions for block ciphers with independent round keys in both, the setting for differential and linear cryptanalysis. However, they did not regard possible influences of the key schedule but usually real block ciphers have a (often linear) key schedule to generate round keys. In particular, we are interested in exactly this case, where the key schedule is linear. Such a key schedule can have unexpected influences on our standard assumptions for block cipher designs. In the following section, we investigate the special case of identical round keys.

3 Bizarre Examples

When we design a new cipher, we typically assume independent round keys and analyze the behavior of linear trails in the hope that the behavior when using an actual key schedule does not differ to much in practice. Note that, mainly due to Nyberg [27], the behavior of independent round keys is well scrutinized. Here, one understands theoretically the basic parameters of the distribution of possible Fourier coefficients for varying keys. In particular, the average Fourier coefficients, that is the mean of the distribution, and the

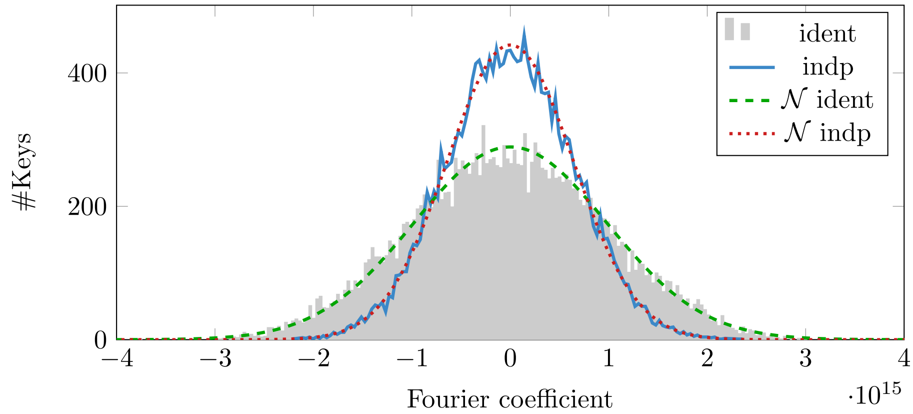


Figure 4: Distribution of Fourier coefficients for standard PRESENT reduced to 10 rounds. Possible Fourier coefficients of the mask (e_{21}, e_{21}) are plotted on the abscissa, while the number of keys that lead to this Fourier coefficient is plotted on the ordinate.

average squared Fourier coefficients can be formalized, as shown in Section 2. Moreover, we often expect the Fourier coefficients of linear trails to follow a normal distribution in the case of independent round keys.

Aside from this general result, only few research was conducted for round keys derived by a key schedule. One rather recent contribution by Abdelraheem *et al.* [2] exhibited Fourier coefficient distributions as in Figure 4. Here, the distribution for identical round keys has a significantly bigger variance than independent round keys, but still follows a normal distribution. Continuing this analysis, we conduct extensive experiments with PRESENT variants and report the observed distributions.

The main motivation behind those experiments was to explore if one can bound the fraction of weak keys, that is keys with a large absolute bias, tighter than by using the very general result by Tchebysheff's bound. In other words, we are interested in studying what can be said about the tails of the Fourier coefficient distribution over the keys.

Recall that, for any probability distribution *Tchebysheff's inequality* gives a result about deviations from the distribution's mean. Let D be a distribution with mean μ and variance σ^2. Then for any random variable $x \sim D$,

$$\Pr_x\left[|x-\mu| \geq k \cdot \sigma\right] \leq \frac{1}{k^2}.$$

While this is a general result for *any* probability distribution, we know much stronger results for some common distributions. In particular for the normal distribution that often seems a good approximation of the distribution of Fourier coefficients, much stronger bounds can be proven. More precisely, when considering a normal distribution, the cumulative distribution function (CDF) results in the well-known *68–95–99.7 rule* (or *three-sigma rule of thumb* [19]) that says

- 68 % of the probability mass lies within one,
- 95 % lies within two, and
- 99.7 % lies within three standard deviations away from the mean.

The remainder of the section discusses our results for some selected S-boxes, while additionally all results are given in the appendix.

Experimental setting

PRESENT is a classical Substitution-Permutation-Network with a 64 bit block size and uses a substitution layer based on a 4 bit S-box with optimal properties regarding differential and linear cryptanalysis, together with a bit permutation based linear layer. In [22], the authors classified all 4 bit S-boxes and found 16 so-called *optimal* equivalence classes and 20 *Serpent-type* equivalence classes.[1] While optimal 4 bit S-boxes exhibit the best uniformity and linearity possible, the notion of Serpent-type S-boxes also include desired attributes to ensure a higher number of active S-boxes for differential cryptanalysis. The PRESENT S-box was chosen from one of these Serpent-type equivalence classes.

In order to better understand the behavior of identical round keys, we conducted extensive experiments with modified PRESENT versions. Our modifications are of the following form. We substituted the used S-box within the encryption with each of the optimal representatives O_0 to O_{15} and Serpent-type representatives R_0 to R_{19} given in [22]. Additionally we reduced the encryption to 10 rounds. For each experimental distribution we then computed the Fourier coefficients of one bit trails for 20 000 independent and 20 000 identical round keys.

Before discussing our results, let us recall some observations of PRESENT. In [29] Ohkuma has shown that one bit trails dominate the linear hull in the case of PRESENT, at least for a limited number of rounds. Later, Abdelraheem [1] showed that with an increasing number of rounds, one has to take into account more trails in order to get good estimates of the total Fourier coefficient. A *one bit trail* $\theta = (\theta_0, \dots, \theta_r)$ is a trail, for which all intermediate masks θ_i have Hamming weight 1, i. e. $\mathrm{wt}(\theta_i) = 1$.

We build on these findings and run our experiments under the following assumption:

Assumption 1. *One bit trails dominate the linear hull of* PRESENT.

We discuss the validity of this assumption in the next subsection, cf. Figure 7 and 8.

As we consider a small number of rounds, we can thus approximate the Fourier coefficient of PRESENT by

$$\widehat{E_k}(\alpha, \gamma) = \sum_{\substack{\theta \in \left(\mathbb{F}_2^n\right)^{r+1} \\ \theta_0 = \alpha, \theta_r = \gamma}} (-1)^{\langle \theta, k \rangle} C_\theta \approx \sum_{\substack{\theta \\ \theta_0 = \alpha, \theta_r = \gamma \\ \mathrm{wt}(\theta_i) = 1}} (-1)^{\langle \theta, k \rangle} C_\theta .$$

We can exploit this observation in our experiments in two ways. First, as we have to consider only one bit trails, computing the Fourier coefficient becomes very efficient compared to computing Fourier coefficients of all trails. The reason for the reduced complexity is the following. Normally we utilize correlation matrices [18] to compute the trail's Fourier coefficient. But as we restrict the trails to one bit masks only, we also greatly reduce the size of the corresponding correlation matrix. Additionally, we can use the resulting matrix as an intuitive illustration of the Fourier coefficient-influencing parts of the cipher. For that purpose, we interpret the correlation matrix restricted to one bit trails as an adjacency matrix of a graph $\mathcal{G}$. We call $\mathcal{G}$ *the induced graph.* Standard PRESENT induces the graph depicted in Figure 5. A vertex in $\mathcal{G}$ corresponds to a bit in the cipher's state, an edge from α to γ to a trail over one round with non-zero Fourier coefficient. That is, α is connected to γ by an edge if

$$\widehat{H}(e_\alpha, e_\gamma) \neq 0,$$

where H denotes the PRESENT round function, and e_j the jth unit vector.

Note that finding one bit trails over r rounds now reduces to finding paths in $\mathcal{G}$ of length r. $\mathcal{G}$ can be reduced in size, if we discard vertices not covered by paths of length r. Counting the number of one bit trails from α to γ over r rounds can now simply be done,

[1] Actually a more general classification was already published in [12]

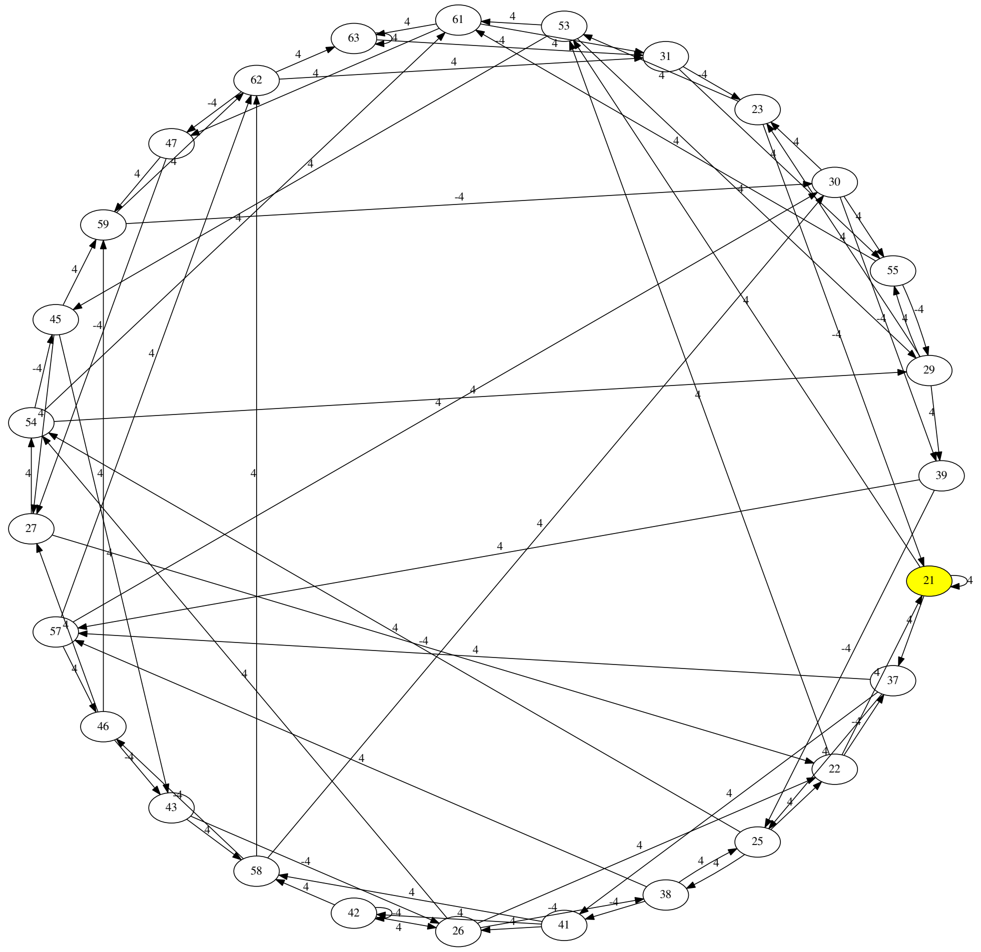


Figure 5: Graph induced by PRESENT. Vertices α, γ correspond to possible one bit masks (e_α, e_γ) and are thus connected by an edge, if the Fourier coefficient at (e_α, e_γ) is non-zero. The highest number of trails is achieved by starting and ending in the marked vertex, $e_\alpha = e_\gamma = e_{21}$.

by raising the adjacency matrix to the rth power. The resulting element at position (α, γ) is the number looked for.

Returning to Ohkuma's observations, the second advantage of this phenomenon is, it limits the number of keys that result in a different behavior. Consider Equation (2) and only one bit trails. The key-dependent sign of the Fourier coefficient now depends only on the few key bits masked by one bit trails. In the case of PRESENT there are actually 27 out of the possible 64 bits of each round key. Thus, significantly fewer key bits influence the Fourier coefficient, and further, all keys which are equal in these masked bits behave identically. For some S-boxes, we can then compute the distribution of Fourier coefficients of one bit trails over all keys.

The induced graph can differ significantly in size for different S-boxes. Figure 6 shows the graph induced by PRESENT and R_1. Compared to standard PRESENT, only 8 of the originally 27 key bits influence the Fourier coefficient.

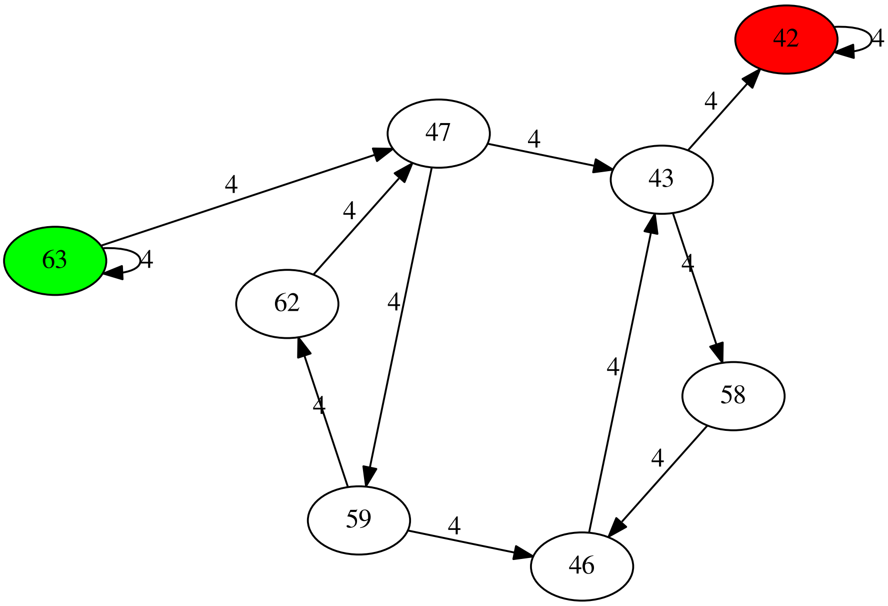


Figure 6: Graph induced by PRESENT and R_1. Vertices α, γ correspond to possible one bit masks (e_α, e_γ) and are thus connected by an edge, if the Fourier coefficient at (e_α, e_γ) is non-zero. The highest number of trails is achieved for $(e_\alpha = e_{63}, e_\gamma = e_{42})$.

Resulting distributions and behavior over several rounds

In our experiments various distributions occur. For some S-boxes we observe the same behavior as for standard PRESENT. Several other S-boxes exhibit unexpected distributions. We do not want to cover every individual distribution in detail here, but plots for each can be found in the appendix. Instead, we concentrate on R_1 (cf. Table 1), which actually is the most interesting example with respect to our initial question, i. e. to study the tails of the distribution. Figure 7 shows the resulting distribution of one bit Fourier coefficients for R_1, cf. the bar plot. In Figure 8 we plot the CDF, which has the advantage that the scaling issue of Figure 7 vanishes.

Clearly, the resulting distribution does not follow a normal distribution.

When observing such a different distribution to the expected normal distribution, the question arises if Ohkuma's initial observation on PRESENT's behavior still is correct. That is, do the one bit trails still dominate the distribution of the Fourier coefficient?

In order to investigate this, we computed the distribution for all two bit trails on top of the one bit trails. As can be seen in Figure 7 the one bit trails still dominate the general shape of the distribution. The two bit trails alone roughly follow a normal distribution with a relative small variance. In total, this has the effect that the two bit trails together with the one bit trails differ from the one bit trails by changing the isolated discrete distribution into roughly bell-shaped parts. Thus, we can still see a clear dominance of the one bit trails in the two bit trail distribution, which supports the underlying assumption. In particular the tail of the distribution is still far from following the normal distribution.

Most importantly in our context of studying the tails of the distributions, Figure 7 exhibits two deviates "quite far" from the distribution's mean. Indeed, those outliers are

Table 1: S-box representative for the equivalence class R_1 that is used in our experiments.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$R_1(x)$	0	3	5	8	6	9	10	7	11	12	14	2	1	15	13	4

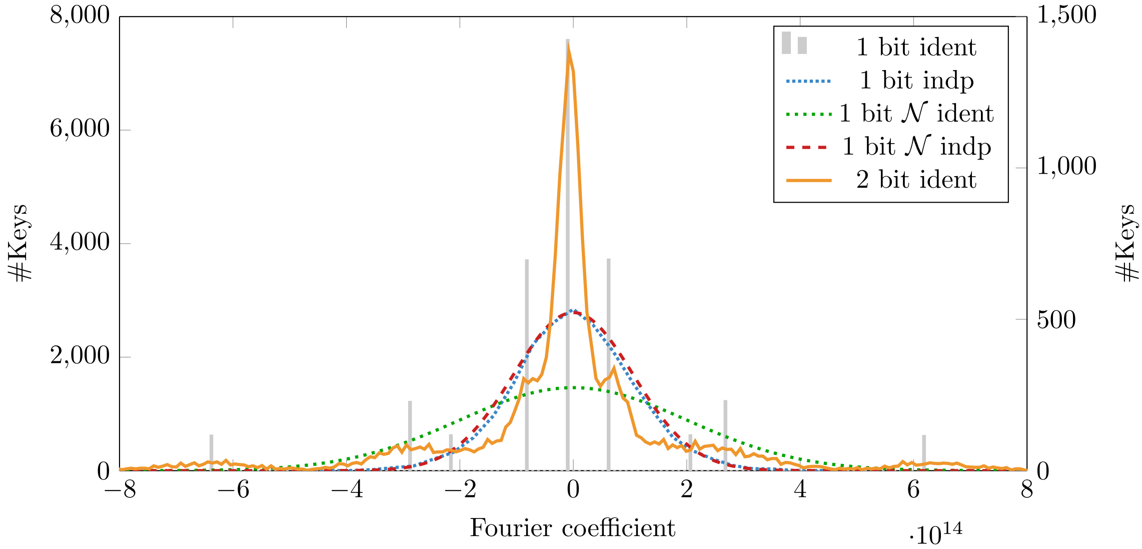


Figure 7: Distribution of Fourier coefficients for Present with R_1 S-box, reduced to 10 rounds. Again, Fourier coefficients for the mask (e_{63}, e_{42}) are plotted on the x-axis, the corresponding number of keys on the y-axis. Note that the plot for two bit trails is plotted against the right y axis.

more than three standard deviations away from the mean and have a joint probability of roughly 3 %. For the standardly assumed normal distribution the corresponding probability to lie outside of $3 \cdot \sigma$ is roughly a factor of 10 smaller, i. e. approximately 0.3 %, cf. the 68–95–99.7 rule. Moreover, when assuming independent round keys (which implies a significantly smaller variance) and a normal distribution, the fraction of keys with an absolute bias larger than $3 \cdot \sigma$ would be roughly 2^{-25}, that is an underestimation by a factor of roughly 2^{20}.

Table 2 summarizes the probabilities of these outliers for ten and twelve rounds.

When increasing the number of rounds further, it can be expected that at some point the dominance of the one bit trails vanishes, especially when correlation of the one bit trails drops below $2^{-n/2}$. However, for increasing number of rounds, the one bit trails show a fascinating behavior that we like to shortly elaborate below.

We normalize the Fourier coefficient by the distribution's standard deviation. For increasing number of rounds, the above mentioned outliers then converge to four standard deviations. Recall that for R_1, only $2^8 = 256$ keys exhibit distinct Fourier coefficients, due to the fact that we only consider one bit trails. The outliers cover 16 out of the 256 possible keys, converging to the following distribution D_{lim}, cf. Figure 9:

$$\widehat{E_k}(\alpha, \gamma) \sim D_{\text{lim}} \begin{cases} -4\sigma & \text{with probability } \frac{1}{32} \\ 0 & \text{with probability } \frac{15}{16} \\ 4\sigma & \text{with probability } \frac{1}{32} \end{cases} .$$

Thus, this distribution fulfills Tchebysheff's bound with equality:

$$256 \cdot \Pr\left[\left|\widehat{E_k}(\alpha, \gamma)\right| \geq 4 \cdot \sigma\right] = 256 \cdot \left(\frac{1}{32} + \frac{1}{32}\right) = 256 \cdot \frac{1}{4^2} = 16.$$

Table 2: Probability of outliers deviating more than $3 \cdot \sigma$, or $\Pr[|X| > 3 \cdot \sigma]$, for one and two bit distributions. For $X \sim \mathcal{N}(0, \sigma)$, $\Pr[|X| > 3 \cdot \sigma] = 0.0027$.

Rounds	$\log_2(\sigma)$	$\log_2(\sigma_{\mathcal{N}})$	$\Pr_{1\text{bit}}$	$\Pr_{2\text{bit}}$	$\log_2(\Pr_{\mathcal{N}})$
10	-16.50	-17.41	0.03130	0.0343	-25.59
12	-19.76	-21.01	0.03205	0.0342	-40.14

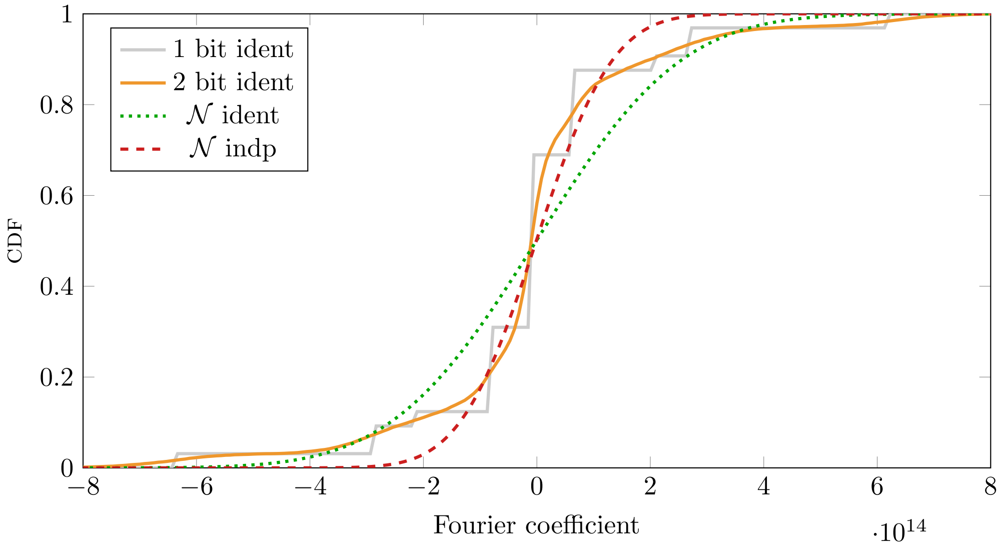


Figure 8: Distribution's CDF of one and two bit Fourier coefficients, and the corresponding normal distribution for identical and independent round keys.

From our perspective of cipher designers, this is a worst case behavior, as such a distribution not only exhibits a wider variance, but also shows a maximal fraction of weak keys possible for a given variance.

Although this resulting distribution is quite contrary to what we typically expect, we have to keep in mind that identical round keys can per se be insecure due to slide [5], invariant subspace [23], or nonlinear invariant attacks [30]. Therefore designs usually involve round constants. The next section takes their influence into account.

While this section points out interesting examples and strange behaviour of the resulting distributions, we clearly lack insights on what causes those peculiarities exactly. We think that it is an interesting and challenging task for future research to theoretically explain our observations. In particular one might ask, why R_1 shows such a peculiar behavior and if there is a connection between the linear approximation table and the resulting distributions.

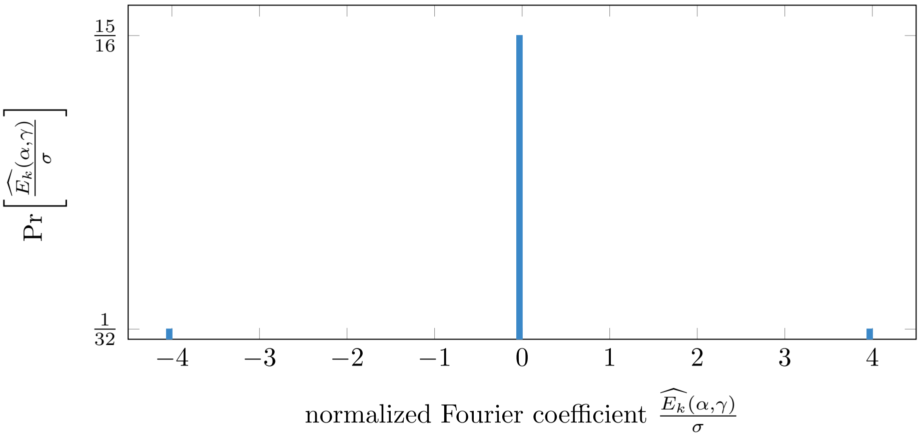


Figure 9: Convergence distribution for Present with R_1 S-box and many rounds. Here, the Fourier coefficient of (e_{63}, e_{42}) is normalized by the standard deviation σ (x-axis), while the corresponding probability to obtain such a Fourier coefficient is denoted on the ordinate.

4 Linear Key Schedules

As mentioned above, in cipher design one typically makes use of the *hypothesis of independent round keys*, which states that the analyzed cipher shows a similar behavior when instantiated with the key schedule or with independent round keys. However, as discussed in the previous section, this assumption might actually be wrong.

Here we show that for any linear key schedule together with randomly chosen round constants, those distributions where the variance is significantly larger than for independent round keys are rare exceptions. That is, we theoretically back-up the use of linear key schedules as a sound design approach with respect to linear cryptanalysis. Interestingly, from a technical point of view, this observation is almost trivial.

We consider a key-alternating cipher and analyze the effect of a key schedule that consists of a linear function followed by the addition of a constant. Thus, the key schedule $\mathsf{KS} : \mathbb{F}_2^\ell \times (\mathbb{F}_2^n)^{r+1} \to (\mathbb{F}_2^n)^{r+1}$ is given as

$$\mathsf{KS}(k, c) = \mathsf{KS}_c(k) = L(k) + c,$$

where $L : \mathbb{F}_2^\ell \to (\mathbb{F}_2^n)^{r+1}$ is a linear function, and $c \in (\mathbb{F}_2^n)^{(r+1)}$. The constant has the form $c = (c_0, \ldots, c_r)$, where the $c_i \in \mathbb{F}_2^n$ are called the round constants.

Let us look at the key-alternating cipher r-KeyAlt using the key schedule KS_c, that is r-$\mathsf{KeyAlt}^{\mathsf{KS}_c}$. First, we note that all constants from the same coset of the linear subspace $U = L(\mathbb{F}_2^\ell)$ result in the same key schedule up to a permutation. Namely, given two constants $c_1 = L(k_1) + d$ and $c_2 = L(k_2) + d$, it holds that $\mathsf{KS}_{c_1}(k) = \mathsf{KS}_{c_2}(k + k_1 + k_2)$. Accordingly, when analyzing the squared Fourier coefficient over the keys, the choice of the constant c can be reduced to the choice of a coset $U + d$.

Applying the linear hull theorem (cf. Proposition 3), we can compute the average squared Fourier coefficient over the keys, that is the variance of the distribution for fixed input and output masks (α, γ) as

$$\begin{aligned}
\mathrm{Var}(c) &:= 2^{-\ell} \sum_{k\in\mathbb{F}_2^{\ell}} r\text{-}\widehat{\mathsf{KeyAlt}}_k^{\mathsf{KS}_c}(\alpha,\gamma)^2 \\
&= 2^{2n-\ell} \sum_{\substack{\theta,\theta'\in\left(\mathbb{F}_2^n\right)^{r+1}\\ \theta_0=\theta'_0=\alpha\\ \theta_r=\theta'_r=\gamma}} (-1)^{\left\langle\theta+\theta',c\right\rangle} C_\theta C_{\theta'} \sum_k (-1)^{\left\langle\theta,L(k)\right\rangle+\left\langle\theta',L(k)\right\rangle} \\
&= 2^{2n-\ell} \sum_{\substack{\theta,\theta'\in\left(\mathbb{F}_2^n\right)^{r+1}\\ \theta_0=\theta'_0=\alpha\\ \theta_r=\theta'_r=\gamma}} (-1)^{\left\langle\theta+\theta',c\right\rangle} C_\theta C_{\theta'} \sum_k (-1)^{\left\langle k,L^T(\theta)+L^T\left(\theta'\right)\right\rangle} \\
&= 2^{2n} \sum_{\substack{\theta,\theta'\\ \theta_0=\theta'_0=\alpha\\ \theta_r=\theta'_r=\gamma\\ L^T(\theta)=L^T(\theta')}} (-1)^{\left\langle\theta+\theta',c\right\rangle} C_\theta C_{\theta'}.
\end{aligned}$$

Next, we look at the average variance over all possible constants c. As discussed above, except for a factor, this is actually the same as summing over one representative for each coset. We have

$$\begin{aligned}
\mathbb{E}_c\left(\mathrm{Var}(c)\right) &= 2^{-(r+1)n} \sum_{c\in(\mathbb{F}_2^n)^{r+1}} \mathrm{Var}(c) \\
&= 2^{2n-(r+1)n} \sum_c \sum_{\substack{\theta,\theta'\\ \theta_0=\theta'_0=\alpha\\ \theta_r=\theta'_r=\gamma\\ L^T(\theta)=L^T(\theta')}} (-1)^{\left\langle\theta+\theta',c\right\rangle} C_\theta C_{\theta'} \\
&= 2^{2n-(r+1)n} \sum_{\substack{\theta,\theta'\\ \theta_0=\theta'_0=\alpha\\ \theta_r=\theta'_r=\gamma\\ L^T(\theta)=L^T(\theta')}} C_\theta C_{\theta'} \sum_c (-1)^{\left\langle\theta+\theta',c\right\rangle} \\
&= 2^{2n} \sum_{\substack{\theta\\ \theta_0=\alpha\\ \theta_r=\gamma}} C_\theta^2.
\end{aligned}$$

Thus, *the average variance over all constants is the same variance as for independent round keys.*

While this is actually quite clear as in both cases we eventually sum over all possible $2^{(r+1)n}$ bit round keys, this observation has an important implication for cipher design.

Having a key-alternating cipher, *any* linear key scheduling can be turned into a key schedule which is on average as good as having independent round keys (in terms of the variance of the distribution, and thus in terms of the fraction of weak keys): Simply choose random round constants.

Known ciphers that actually deploying this approach (for different reasons) include the low-latency cipher PRINCE [11] and the cipher LOWMC [3].

We conducted experiments on how the distributions actually vary for different choices of random round constants. As we will see in the following, in this case not only the variance behaves as in the independent round key set-up, but the whole distribution does.

Experiments

We experimentally verified our results in the same setting as discussed in Section 3. Figure 10 plots the resulting Fourier coefficient distribution.

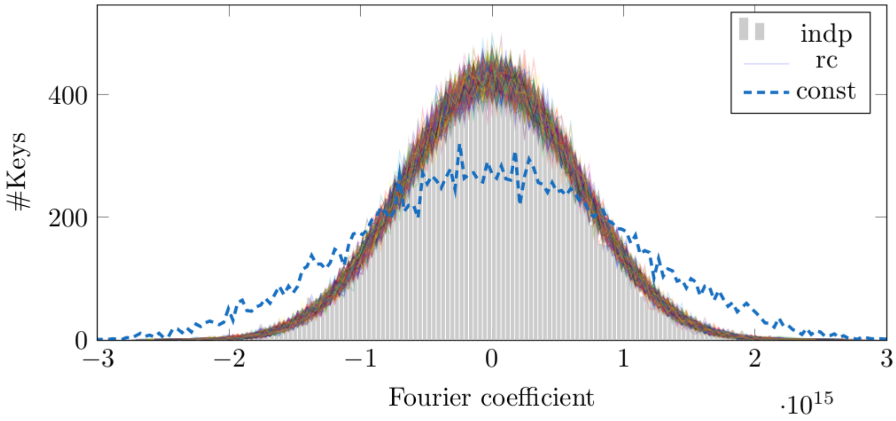


Figure 10: Experimental distributions for Present with different key schedules. The gray histogram is for independent random round keys. The dashed line is for identical round keys and an all zero round constant. All other lines, are for identical round keys and independent random round constants.

The gray histogram in the background represents the distribution for independent random round keys. It smoothly follows a normal distribution as expected. The dashed line in the foreground depicts the distribution for identical round keys with an all zero round constant. This distribution is similar to the independent round key case, but exhibits a wider variance, as already observed in [2] and discussed in Section 3. According to our results from above, this behavior must be a clear outlier.

Indeed, all other lines correspond to identical round keys with a random round constant, and all exhibit the same behavior following a normal distribution. While the plot only shows 256 different round constants, we conducted the same experiment for several thousand random round constants, each resulting in the same behavior.

5 Linear Approximations of Tweakable Block Ciphers

Tweakable block ciphers, introduced by Liskov et al. [24], are an important cryptographic primitive. A traditional block cipher takes as input a key and a message and computes a ciphertext. For each fixed key, the function mapping the message to a ciphertext is a permutation (to allow decryption) and thus, a block cipher indexed by the key can be seen as a family of permutations. The idea of a tweakable block cipher is that besides the key and the message, a tweak is taken as an input. Informally, the intuition is that each tweak selects a different block cipher, that is a different, unrelated, family of permutations. While the key is, obviously, assumed to be unknown to an attacker, the tweak, as well as the message, is usually assumed to be under full control of an adversary. That is, the adversary is usually allowed to query the tweakable block cipher under a message and tweak of her choice. Tweakable block ciphers have many important applications, e. g. ciphers for memory-encryption can use the memory-address as a tweak, further applications are efficient authenticated encryption and online ciphers.

For a tweakable block cipher the attacker is no longer restricted to linear approximations from the plaintext to the ciphertext, but can also make use of the tweak. In this section, we develop a formula for the linear hull of a tweakable block cipher and discuss its implications. We again develop our formulas in a top-down manner starting with a generic tweakable block cipher and then looking at the more special cases step by step.

A tweakable block cipher takes as input a key k, a tweak t, and a message x and computes the ciphertext c. It can then be written as a function

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

where m denotes the tweak size and, for simplicity, we hide the key-dependency in the function F itself. This means that we do not explicitly mention the key-dependency of F in our notation.

As in the case of keys, usually real block ciphers do not have independent tweaks, but rather have a tweak schedule generating the round tweaks. For the tweak schedule

$$\mathsf{TS} : \mathbb{F}_2^\ell \to \mathbb{F}_2^m$$

we define

$$F^{\mathsf{TS}} : \mathbb{F}_2^n \times \mathbb{F}_2^\ell \to \mathbb{F}_2^n$$

as

$$F^{\mathsf{TS}}(x,t) \coloneqq F(x, \mathsf{TS}(t)).$$

Analogous to Section 2, we define $E_t^{\mathsf{TS}}(x) \coloneqq F^{\mathsf{TS}}(x,t)$.

With the plaintext and the tweak, there are now two public inputs. Accordingly, an input mask for a linear approximation now consists of two parts, (α, β), the plaintext mask α and the tweak mask β. The main question is now how to express the Fourier coefficient of this linear approximation, that is how to compute $\widehat{F^{\mathsf{TS}}}((\alpha,\beta),\gamma)$. While the most basic case of this linear hull for tweakable block ciphers was already discussed in Equation (3), one can observe the following relation for a linear tweak schedule:

Proposition 4. *With the notation from above, for a linear tweak schedule L, it holds that*

$$\widehat{F^L}((\alpha,\beta),\gamma) = 2^{\ell-m} \sum_{\substack{\theta \in \mathbb{F}_2^m \\ L^T(\theta)=\beta}} \widehat{F}((\alpha,\theta),\gamma).$$

Proof. As we have used the notation of a block cipher in two variables already intensively in Section 2, we can now reuse the results for tweakable ciphers. Accordingly, the basic ingredients for the proof are already known from that section. We first apply Equation (3), then Equation (2) and eventually use some basic summation techniques.

$$\begin{aligned}
\widehat{F^L}((\alpha,\beta),\gamma) &= \sum_{t \in \mathbb{F}_2^\ell} (-1)^{\langle \beta, t\rangle} \widehat{E_t^L}(\alpha,\gamma) \\
&= 2^{-m} \sum_{t \in \mathbb{F}_2^\ell} (-1)^{\langle \beta, t\rangle} \sum_{\theta \in \mathbb{F}_2^m} (-1)^{\langle \theta, L(t)\rangle} \widehat{F}((\alpha,\theta),\gamma) \\
&= 2^{-m} \sum_{\theta \in \mathbb{F}_2^m} \widehat{F}((\alpha,\theta),\gamma) \sum_{t \in \mathbb{F}_2^\ell} (-1)^{\langle \beta, t\rangle + \langle \theta, L(t)\rangle} \\
&= 2^{-m} \sum_{\theta \in \mathbb{F}_2^m} \widehat{F}((\alpha,\theta),\gamma) \sum_{t \in \mathbb{F}_2^\ell} (-1)^{\left\langle \beta + L^T(\theta), t\right\rangle} \\
&= 2^{\ell-m} \sum_{\substack{\theta \in \mathbb{F}_2^m \\ L^T(\theta)=\beta}} \widehat{F}((\alpha,\theta),\gamma)
\end{aligned}$$

□

In the following, we will consider what we call *tweak-alternating ciphers* analogous to key-alternating ciphers. Actually, all tweakable block ciphers we are aware of, including secondary constructions, are of this form. A tweak-alternating cipher is defined as

$$r\text{-}\mathsf{TweakAlt}(x,t) : \mathbb{F}_2^n \times \left(\mathbb{F}_2^n\right)^{r+1} \to \mathbb{F}_2^n$$
$$r\text{-}\mathsf{TweakAlt}(x,t) \coloneqq H_{r-1}(\dots H_0(x+t_0)+\dots)+t_r$$

with $H_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$. Analogous to above, the key-dependency is hidden in the round functions H_i.

Substituting this definition in Proposition 4 and using Lemma 1 directly gives the following corollary:

Corollary 1.

$$r\text{-}\widehat{\mathsf{TweakAlt}}^L((\alpha,\beta),\gamma) = 2^{\ell-(r+1)n} \sum_{\substack{\theta\in\left(\mathbb{F}_2^n\right)^{r+1}\\ L^T(\theta)=\beta}} r\text{-}\widehat{\mathsf{TweakAlt}}((\alpha,\theta),\gamma) = 2^{\ell+n} \sum_{\substack{\theta\in\left(\mathbb{F}_2^n\right)^{r+1}\\ L^T(\theta)=\beta\\ \theta_0=\alpha,\theta_r=\gamma}} \prod_{i=0}^{r-1} c_{H_i}(\theta_i,\theta_{i+1})$$

Note that we cannot yet write the last product as a trail correlation C_θ at this point because the influence of the key is still hidden in the round functions H_i. However, looking at a cipher that is not only tweak-alternating but also key-alternating, we can finally express the linear hull in terms of the trail correlations.

Corollary 2. *Let r-$\mathsf{TweakAlt}^L$ be a tweak-alternating cipher where the round keys $k = (k_0,\dots,k_r)$ are added in a key-alternating way. It holds that*

$$r\text{-}\widehat{\mathsf{TweakAlt}}^L((\alpha,\beta),\gamma) = 2^{\ell+n} \sum_{\substack{\theta\\ L^T(\theta)=\beta\\ \theta_0=\alpha,\theta_r=\gamma}} (-1)^{\langle\theta,k\rangle} C_\theta.$$

The crucial observation of Proposition 4 is that tweaking a block cipher with a linear tweak schedule does not introduce any new linear trails. In other words, the tweakable block cipher's linear hulls consists of linear trails that already exist in the linear hulls for the non-tweakable cipher. In the case of Corollary 2, this effect is even more obvious. As explained in the introduction, this stands in contrast to differential trails, where it might well be that adding a difference in the tweak leads to new differential characteristics with a significantly higher probability than any differential characteristic for the non-tweaked version of the cipher.

In particular, from a designer's point of view, protecting a tweakable block cipher with linear tweak schedule against linear cryptanalysis is not more difficult than for non-tweaked ciphers. In almost all settings, the best one can do as a designer, is to bound the correlation of single trails. As those trails are valid both for the tweaked as for the non-tweaked version, no special attention has to be payed to the additional freedom of the attacker. However, and this is important to note, in a tweakable block cipher, the attacker is potentially able to collect more data than in a traditional cipher, where the data complexity is clearly bounded by the block size. Thus, while the bounds are valid for both scenarios, a tweakable block cipher might require stronger bounds on the correlation of trails in order to argue its security. Again, the method of obtaining this bound stays the same when moving from a non-tweaked to a tweakable block cipher.

It is an interesting question how the new degrees of freedom influence the linear hull in concrete examples. As the linear hull is composed differently than before, it might in some cases still enable the attacker to run a better linear attack than originally, although the underlying linear trails have not changed. To this end, future work could consist in experimentally analyzing and comparing the success of these attacks.

Acknowledgements

We would like to thank Anne Canteaut and Kaisa Nyberg for interesting discussions and valuable comments on earlier versions of this paper. This work was supported by the German Research Foundation and the DFG Research Training Group GRK 1817 UbiCrypt.

References

[1] Mohamed Ahmed Abdelraheem. "Estimating the Probabilities of Low-Weight Differential and Linear Approximations on PRESENT-Like Ciphers." In: *ICISC*. Vol. 7839. LNCS. Springer, 2012, pp. 368–382. DOI: 10.1007/978-3-642-37682-5_26.

[2] Mohamed Ahmed Abdelraheem, Martin Ågren, Peter Beelen, and Gregor Leander. "On the Distribution of Linear Biases: Three Instructive Examples." In: *CRYPTO'12*. Vol. 7417. LNCS. Springer, 2012, pp. 50–67. DOI: 10.1007/978-3-642-32009-5_4.

[3] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. "Ciphers for MPC and FHE." In: *EUROCRYPT'15*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 430–454. DOI: 10.1007/978-3-662-46800-5_17.

[4] Eli Biham. "On Matsui's Linear Cryptanalysis." In: *EUROCRYPT'94*. Vol. 950. LNCS. Springer, 1994, pp. 341–355. DOI: 10.1007/BFb0053449.

[5] Alex Biryukov and David Wagner. "Slide Attacks." In: *FSE'99*. Vol. 1636. LNCS. Springer, 1999, pp. 245–259. DOI: 10.1007/3-540-48519-8_18.

[6] Céline Blondeau and Kaisa Nyberg. "Improved Parameter Estimates for Correlation and Capacity Deviates in Linear Cryptanalysis." In: *IACR Transactions on Symmetric Cryptology* 2016.2 (2017), pp. 162–191. DOI: 10.13154/tosc.v2016.i2.162-191.

[7] Céline Blondeau and Kaisa Nyberg. "Joint Data and Key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and Its Impact to Data Complexity." In: *Designs, Codes and Cryptography* 82.1 (2017), pp. 319–349. DOI: 10.1007/s10623-016-0268-6.

[8] Céline Blondeau, Thomas Peyrin, and Lei Wang. "Known-Key Distinguisher on Full PRESENT." In: *CRYPTO'15*. Vol. 9215. LNCS. Springer, 2015, pp. 455–474.

[9] Andrey Bogdanov and Vincent Rijmen. "Linear hulls with correlation zero and linear cryptanalysis of block ciphers." In: *Designs, Codes and Cryptography* 70.3 (2014), pp. 369–383. DOI: 10.1007/s10623-012-9697-z.

[10] Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre. "Multivariate Linear Cryptanalysis: The Past and Future of PRESENT." In: *IACR Cryptology ePrint Archive* 2016.667 (2016).

[11] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. "PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract." In: *ASIACRYPT'12*. Vol. 7658. LNCS. Springer, 2012, pp. 208–225. DOI: 10.1007/978-3-642-34961-4_14.

[12] Christophe De Cannière. "Analysis and design of symmetric encryption algorithms." PhD thesis. Katholieke Universiteit Leuven, 2007.

[13] Claude Carlet. "Boolean Functions for Cryptography and Error Correcting Codes." In: *Boolean Methods and Models*. Ed. by Yves Crama and Peter Hammer. Cambridge University Press, 2007.

[14] Joo Yeon Cho. “Linear Cryptanalysis of Reduced-Round PRESENT.” In: *CT-RSA'10*. Vol. 5985. LNCS. Springer, 2010, pp. 302–317. DOI: 10.1007/978-3-642-11925-5_21.

[15] Joan Daemen. “Cipher and Hash Function Design Strategies based on linear and differential cryptanalysis.” PhD thesis. Katholieke Universiteit Leuven, Mar. 1995.

[16] Joan Daemen and Vincent Rijmen. “Probability distributions of correlation and differentials in block ciphers.” In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 221–242. DOI: 10.1515/JMC.2007.011.

[17] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. ISBN: 3-540-42580-2. DOI: 10.1007/978-3-662-04722-4.

[18] Joan Daemen, René Govaerts, and Joos Vandewalle. “Correlation Matrices.” In: *FSE'94*. Vol. 1008. LNCS. Springer, 1994, pp. 275–285. DOI: 10.1007/3-540-60590-8_21.

[19] Erik W Grafarend. *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models*. Walter de Gruyter, 2006. ISBN: 3-110-16216-4.

[20] Jialin Huang, Serge Vaudenay, Xuejia Lai, and Kaisa Nyberg. “Capacity and Data Complexity in Multidimensional Linear Attack.” In: *CRYPTO'15*. Vol. 9215. LNCS. Springer, 2015, pp. 141–160. DOI: 10.1007/978-3-662-47989-6_7.

[21] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. “Tweaks and Keys for Block Ciphers: The TWEAKEY Framework.” In: *ASIACRYPT'14*. Vol. 8874. LNCS. Springer, 2014, pp. 274–288. DOI: 10.1007/978-3-662-45608-8_15.

[22] Gregor Leander and Axel Poschmann. “On the Classification of 4 Bit S-Boxes.” In: *WAIFI'07*. Ed. by Claude Carlet and Berk Sunar. Vol. 4547. LNCS. Springer, Heidelberg, June 2007, pp. 159–176. DOI: 10.1007/978-3-540-73074-3_13.

[23] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. “A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack.” In: *CRYPTO'11*. Vol. 6841. LNCS. Springer, 2011, pp. 206–221. DOI: 10.1007/978-3-642-22792-9_12.

[24] Moses Liskov, Ronald L. Rivest, and David Wagner. “Tweakable Block Ciphers.” In: *CRYPTO'02*. Vol. 2442. LNCS. Springer, 2002, pp. 31–46. DOI: 10.1007/3-540-45708-9_3.

[25] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher.” In: *EUROCRYPT'93*. Vol. 765. LNCS. Springer, 1993, pp. 386–397. DOI: 10.1007/3-540-48285-7_33.

[26] Kaisa Nyberg. “Correlation theorems in cryptanalysis.” In: *Discrete Applied Mathematics* 111.1-2 (2001), pp. 177–188. DOI: 10.1016/S0166-218X(00)00351-6.

[27] Kaisa Nyberg. “Linear Approximation of Block Ciphers.” In: *EUROCRYPT'93*. Vol. 950. LNCS. Springer, 1994, pp. 439–444. DOI: 10.1007/BFb0053460.

[28] Kaisa Nyberg. *Linear Cryptanalysis (Lecture Notes)*. SAC 2015 Summer School. Available at http://mta.ca/sac2015/S3-linear-all.pdf. Aug. 2015.

[29] Kenji Ohkuma. “Weak Keys of Reduced-Round PRESENT for Linear Cryptanalysis.” In: *SAC'08*. Vol. 5867. LNCS. Springer, 2009, pp. 249–265. DOI: 10.1007/978-3-642-05445-7_16.

[30] Yosuke Todo, Gregor Leander, and Yu Sasaki. “Nonlinear Invariant Attack - Practical Attack on Full SCREAM, iSCREAM, and Midori64.” In: *ASIACRYPT'16*. Vol. 10032. LNCS. Springer, 2016, pp. 3–33. DOI: 10.1007/978-3-662-53890-6_1.

A Proofs

All proofs involve only basic summation techniques. We nevertheless include each, for the sake of completeness and their educational purpose. While all propositions can be proven directly, we only do so for the first proposition. For the following ones, we use well-known lemmas, which we introduce first. Again, we include these lemmas, because they provide a valuable set of tools for proofs regarding linear cryptanalysis.

A.1 Tools for Linear Cryptanalysis Proofs

The following lemma was proven by Nyberg [26, Theorem 3].

Lemma 2 (Consecutive Functions)**.**

Given

$$f : \mathbb{F}_2^n \times \mathbb{F}_2^\ell \to \mathbb{F}_2^k, \quad g : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^k, \quad h : \mathbb{F}_2^\ell \to \mathbb{F}_2^m,$$
$$f(x, y) \coloneqq g(x, h(y)).$$

Then

$$2^m \widehat{f}((\alpha, \beta), \gamma) = \sum_{\beta' \in \mathbb{F}_2^m} \widehat{g}((\alpha, \beta'), \gamma) \cdot \widehat{h}(\beta, \beta').$$

Proof. We only need the well-known fact that for the dot product it holds:

$$\sum_{\beta \in \mathbb{F}_2^n} (-1)^{\langle \beta, x \rangle} = \begin{cases} 2^n & \text{, if } x = 0 \\ 0 & \text{, else} \end{cases} .$$

Hence:

$$\begin{aligned} \sum_{\beta' \in \mathbb{F}_2^m} \widehat{g}((\alpha, \beta'), \gamma) \cdot \widehat{h}(\beta, \beta') &= \sum_{\beta'} \sum_{\substack{x \in \mathbb{F}_2^n \\ y \in \mathbb{F}_2^m}} (-1)^{\left\langle \alpha, x \right\rangle + \left\langle \beta', y \right\rangle + \left\langle \gamma, g(x,y) \right\rangle} \sum_{z \in \mathbb{F}_2^\ell} (-1)^{\left\langle \beta, z \right\rangle + \left\langle \beta', h(z) \right\rangle} \\ &= \sum_{x,y,z} (-1)^{\left\langle \alpha, x \right\rangle + \left\langle \beta, z \right\rangle + \left\langle \gamma, g(x,y) \right\rangle} \sum_{\beta'} (-1)^{\left\langle \beta', y + h(z) \right\rangle} \\ &= 2^m \sum_{x,z} (-1)^{\langle \alpha, x \rangle + \langle \beta, z \rangle + \langle \gamma, g(x, h(z)) \rangle} \\ &= 2^m \widehat{f}((\alpha, \beta), \gamma) \end{aligned}$$

□

The next lemma was discussed by Daemen *et al.* [18, Eq. (15)].

Lemma 3 (Function Composition)**.**

Given

$$f : \mathbb{F}_2^n \to \mathbb{F}_2^k, \quad g : \mathbb{F}_2^n \to \mathbb{F}_2^m, \quad h : \mathbb{F}_2^m \to \mathbb{F}_2^k,$$
$$f \coloneqq h \circ g$$

Then

$$2^m \widehat{f}(\alpha, \gamma) = \sum_{\beta \in \mathbb{F}_2^m} \widehat{g}(\alpha, \beta) \cdot \widehat{h}(\beta, \gamma).$$

Proof.

$$\begin{aligned}\sum_{\beta\in\mathbb{F}_2^m}\widehat{g}(\alpha,\beta)\cdot\widehat{h}(\beta,\gamma) &= \sum_{\beta}\sum_{x\in\mathbb{F}_2^n}(-1)^{\langle\alpha,x\rangle+\langle\beta,g(x)\rangle}\sum_{y\in\mathbb{F}_2^m}(-1)^{\langle\beta,y\rangle+\langle\gamma,h(y)\rangle}\\ &= \sum_{x,y}(-1)^{\langle\alpha,x\rangle+\langle\gamma,h(y)\rangle}\sum_{\beta}(-1)^{\langle\beta,y+g(x)\rangle}\\ &= 2^m\sum_{x}(-1)^{\langle\alpha,x\rangle+\langle\gamma,h(g(x))\rangle}\\ &= 2^m\widehat{f}(\alpha,\gamma)\end{aligned}$$

□

We can easily prove a variant of this lemma for functions with an other, independent input.

Lemma 4.

Given

$$f:\mathbb{F}_2^n\times\left(\mathbb{F}_2^{\ell_1}\times\mathbb{F}_2^{\ell_2}\right)\to\mathbb{F}_2^k,\quad g:\mathbb{F}_2^n\times\mathbb{F}_2^{\ell_1}\to\mathbb{F}_2^m,\quad h:\mathbb{F}_2^m\times\mathbb{F}_2^{\ell_2}\to\mathbb{F}_2^k,$$
$$f(x,(y,z)) := h(g(x,y),z).$$

Then, for $\beta=(\beta_0,\beta_1)$

$$2^m\widehat{f}((\alpha,\beta),\gamma)=\sum_{\theta\in\mathbb{F}_2^m}\widehat{g}((\alpha,\beta_0),\theta)\cdot\widehat{h}((\theta,\beta_1),\gamma).$$

Proof.

$$\begin{aligned}\sum_{\theta\in\mathbb{F}_2^m}\widehat{g}((\alpha,\beta_0),\theta)\cdot\widehat{h}((\theta,\beta_1),\gamma) &= \sum_{\theta}\sum_{\substack{x\in\mathbb{F}_2^n\\ z\in\mathbb{F}_2^{\ell_1}}}(-1)^{\big\langle\alpha,x\big\rangle+\big\langle\beta_0,z\big\rangle+\big\langle\theta,g(x,z)\big\rangle}\sum_{\substack{y\in\mathbb{F}_2^m\\ z'\in\mathbb{F}_2^{\ell_2}}}(-1)^{\big\langle\theta,y\big\rangle+\big\langle\beta_1,z'\big\rangle+\big\langle\gamma,h\big(y,z'\big)\big\rangle}\\ &= \sum_{\substack{x,y\\ z,z'}}(-1)^{\big\langle\alpha,x\big\rangle+\big\langle\beta_0,z\big\rangle+\big\langle\beta_1,z'\big\rangle+\big\langle\gamma,h\big(y,z'\big)\big\rangle}\sum_{\theta}(-1)^{\big\langle\theta,y+g(x,z)\big\rangle}\\ &= 2^m\sum_{x,z,z'}(-1)^{\big\langle\alpha,x\big\rangle+\big\langle\beta_0,z\big\rangle+\big\langle\beta_1,z'\big\rangle+\big\langle\gamma,h\big(g(x,z),z'\big)\big\rangle}\\ &= 2^m\widehat{f}((\alpha,\beta),\gamma)\end{aligned}$$

□

Bogdanov and Rijmen [9, Lemma 1] studied how the XOR operation influences linear cryptanalysis.

Lemma 5 (XOR at input)**.**

Given

$$g:\mathbb{F}_2^n\times\mathbb{F}_2^n\to\mathbb{F}_2^k,\ \textit{and}\ f:\mathbb{F}_2^n\to\mathbb{F}_2^k,$$
$$g(x,y):=f(x+y).$$

Then

$$\widehat{g}((\alpha,\beta),\gamma)=\begin{cases}2^n\widehat{f}(\alpha,\gamma) & ,\ \textit{if}\ \alpha=\beta\\ 0 & ,\ \textit{else}\end{cases}.$$

Proof.

$$\begin{aligned}
\widehat{g}((\alpha,\beta),\gamma) &= \sum_{x,y\in\mathbb{F}_2^n} (-1)^{\langle\alpha,x\rangle+\langle\beta,y\rangle+\langle\gamma,f(x+y)\rangle} \\
&= \sum_{x',y} (-1)^{\left\langle\alpha,x'+y\right\rangle+\left\langle\beta,y\right\rangle+\left\langle\gamma,f\left(x'\right)\right\rangle} \\
&= \sum_{x',y} (-1)^{\left\langle\alpha,x'\right\rangle+\left\langle\alpha,y\right\rangle+\left\langle\beta,y\right\rangle+\left\langle\gamma,f\left(x'\right)\right\rangle} \\
&= \sum_{x'} (-1)^{\left\langle\alpha,x'\right\rangle+\left\langle\gamma,f\left(x'\right)\right\rangle} \sum_{y} (-1)^{\langle\alpha+\beta,y\rangle} \\
&= \widehat{f}(\alpha,\gamma)\cdot\sum_{y} (-1)^{\langle\alpha+\beta,y\rangle} \\
&= \begin{cases} 2^n\widehat{f}(\alpha,\gamma) & \text{, if } \alpha=\beta \\ 0 & \text{, else} \end{cases}
\end{aligned}$$

□

Lemma 6 (XOR at ouput)**.**

Given

$$g:\mathbb{F}_2^k\times\mathbb{F}_2^n\to\mathbb{F}_2^n,\ \textit{and}\ f:\mathbb{F}_2^k\to\mathbb{F}_2^n,$$
$$g(x,y):=f(x)+y.$$

Then

$$\widehat{g}((\alpha,\beta),\gamma)=\begin{cases} 2^n\widehat{f}(\alpha,\gamma) & \textit{, if } \beta=\gamma \\ 0 & \textit{, else} \end{cases}.$$

Proof. The proof works analogous to the proof of Lem. 5. □

Using the above lemmas, most of the remaining proofs are straightforward.

A.2 Proofs of Propositions 1–3

Proposition 1

We compute directly

$$\begin{aligned}
\sum_{\beta}(-1)^{\langle\beta,k\rangle}\widehat{F}((\alpha,\beta),\gamma) &= \sum_{\beta}(-1)^{\left\langle\beta,k\right\rangle}\sum_{x,k'}(-1)^{\left\langle\alpha,x\right\rangle+\left\langle\beta,k'\right\rangle+\left\langle\gamma,F\left(x,k'\right)\right\rangle} \\
&= \sum_{\beta,x,k'}(-1)^{\left\langle\alpha,x\right\rangle+\left\langle\beta,k+k'\right\rangle+\langle\gamma,E_{k'}(x)\rangle} \\
&= \sum_{x,k'}(-1)^{\left\langle\alpha,x\right\rangle+\left\langle\gamma,E_{k'}(x)\right\rangle}\sum_{\beta}(-1)^{\left\langle\beta,k+k'\right\rangle} \\
&= 2^m\sum_{x}(-1)^{\langle\alpha,x\rangle+\langle\gamma,E_k(x)\rangle} \\
&= 2^m\widehat{E_k}(\alpha,\gamma).
\end{aligned}$$

For the key scheduled variant: $E_k^{\mathsf{KS}}(x) = F^{\mathsf{KS}}(x,k) = F(x, \mathsf{KS}(k))$. With the first part of Prop. 1 for the non key scheduled variant we have

$$2^{\ell}\widehat{E_k^{\mathsf{KS}}}(\alpha,\gamma) = \sum_{\beta}(-1)^{\langle\beta,k\rangle}\widehat{F^{\mathsf{KS}}}((\alpha,\beta),\gamma),$$

applying Lem. 2 results in

$$2^{\ell+m}\widehat{E_k^{\mathsf{KS}}}(\alpha,\gamma) = \sum_{\beta,\beta'}(-1)^{\langle\beta,k\rangle}\widehat{\mathsf{KS}}(\beta,\beta')\widehat{F}((\alpha,\beta'),\gamma),$$

which concludes the proof. □

Proposition 2

Recall $r\text{-}\mathsf{Round}_k(x) = G_{r-1}(\ldots(G_0(x,k_0),\ldots),k_{r-1})$. With the first part of Prop. 1 for the non key scheduled variant it holds

$$2^{rm}\widehat{r\text{-}\mathsf{Round}}_k(\alpha,\gamma) = \sum_{\beta}(-1)^{\langle\beta,k\rangle}\widehat{F}((\alpha,\beta),\gamma).$$

Applying Lem. 4 iteratively $r-1$ times we then get

$$2^{rm+(r-1)n}\widehat{r\text{-}\mathsf{Round}}_k(\alpha,\gamma) = \sum_{\beta}(-1)^{\langle\beta,k\rangle}\sum_{\substack{\theta\\ \theta_0=\alpha,\theta_r=\gamma}}\prod_{i=0}^{r-1}\widehat{G_i}((\theta_i,\beta_i),\theta_{i+1}).$$

The key scheduled variant follows from the second part of Prop. 1 and again applying Lem. 4 iteratively $r-1$ times. □

Proposition 3

Using Prop. 1, Lem. 6, and Lem. 4 results in

$$2^{(2r-1)n}\widehat{r\text{-}\mathsf{KeyAlt}}_k(\alpha,\gamma) = \sum_{\substack{\beta\\ \beta_r=\gamma}}(-1)^{\langle\beta,k\rangle}\sum_{\substack{\theta\\ \theta_0=\alpha,\theta_r=\gamma}}\prod_{i=0}^{r-1}\widehat{G_i}((\theta_i,\beta_i),\theta_{i+1}),$$

and applying Lem. 5 for each round yields

$$2^{(r-1)n}\widehat{r\text{-}\mathsf{KeyAlt}}_k(\alpha,\gamma) = \sum_{\substack{\beta\\ \beta_0=\alpha,\beta_r=\gamma}}(-1)^{\langle\beta,k\rangle}\prod_{i=0}^{r-1}\widehat{H_i}(\beta_i,\beta_{i+1}).$$

□

The key scheduled variant follows analogously from the second part of Proposition 1.

A.3 Proof of Lemma 1

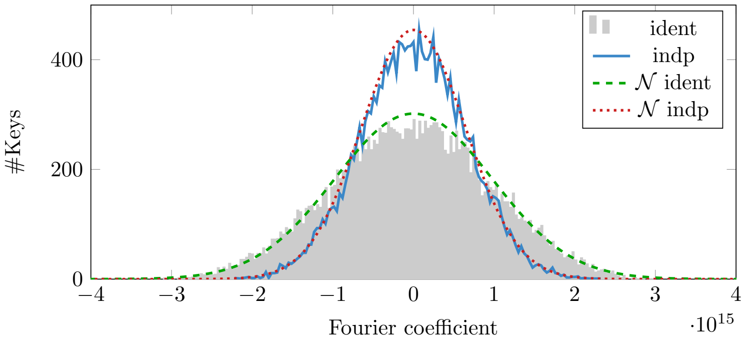
$$2^{-(r+2)n}\widehat{F}((\alpha,\beta),\gamma) = \begin{cases}\prod_{i=0}^{r-1} c_{H_i}(\beta_i,\beta_{i+1}) = C_\beta & \text{, for } (\alpha,\gamma) = (\beta_0,\beta_r)\\ 0 & \text{, else}\end{cases}.$$

With Equation (3):

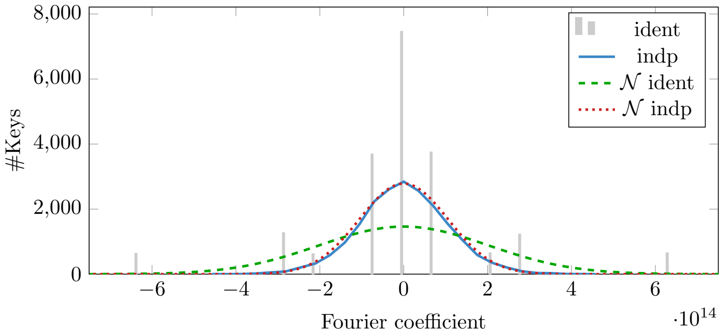
$$
\begin{aligned}
\widehat{F}((\alpha,\beta),\gamma) &= \sum_{k} (-1)^{\langle \beta,k\rangle} \widehat{E_k}(\alpha,\gamma) \\
&= \sum_{k} (-1)^{\langle \beta,k\rangle} \left(2^n \sum_{\substack{\beta' \\ \beta'_0=\alpha,\beta'_r=\gamma}} (-1)^{\langle \beta',k\rangle} \prod_{i=0}^{r-1} c_{H_i}\left(\beta'_i,\beta'_{i+1}\right) \right) \\
&= 2^n \sum_{\beta',k} (-1)^{\langle \beta+\beta',k\rangle} \prod_{i=0}^{r-1} c_{H_i}\left(\beta'_i,\beta'_{i+1}\right) \\
&= \begin{cases} 2^{(r+2)n} \prod_{i=0}^{r-1} c_{H_i}\left(\beta'_i,\beta'_{i+1}\right) & \text{, for } (\alpha,\gamma)=(\beta_0,\beta_r) \\ 0 & \text{, else} \end{cases} .
\end{aligned}
$$

□

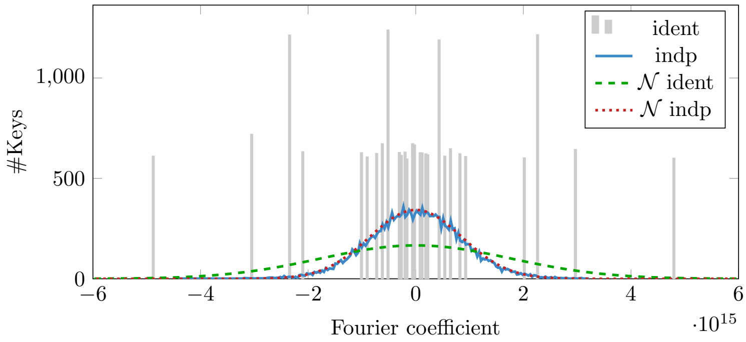
B Plots for Serpent-type S-boxes



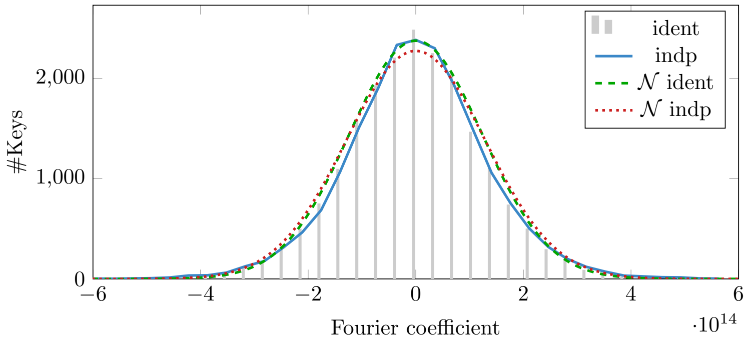
Distribution of Fourier coefficient for Present with R_0 and 10 rounds.



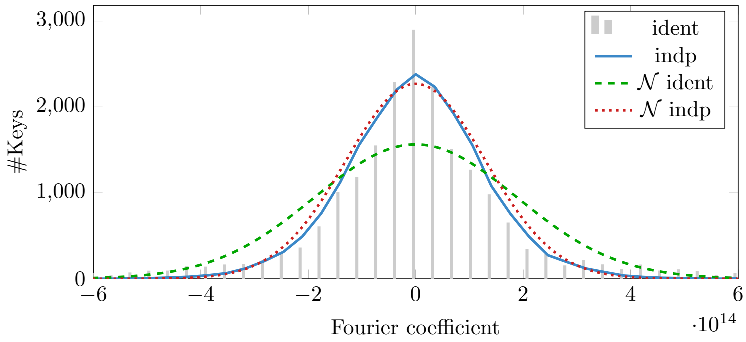
Distribution of Fourier coefficient for Present with R_1 and 10 rounds.



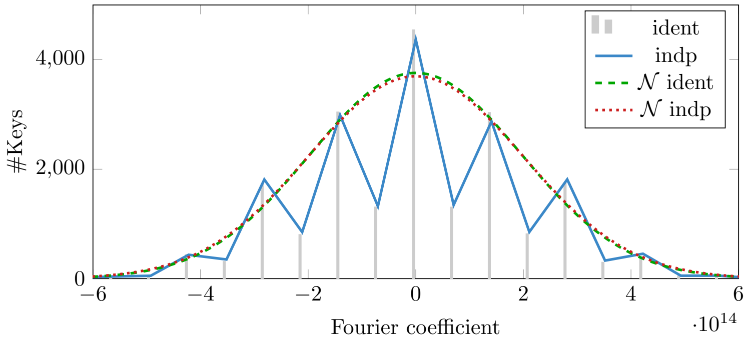
Distribution of Fourier coefficient for PRESENT with R_2 and 10 rounds.



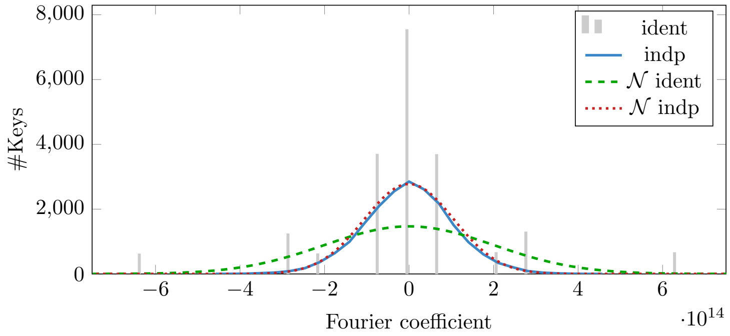
Distribution of Fourier coefficient for PRESENT with R_3 and 10 rounds.



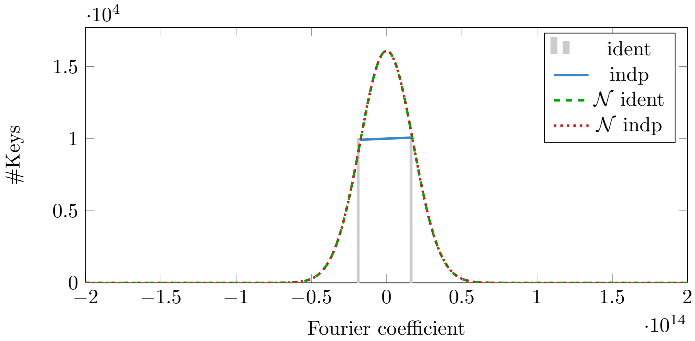
Distribution of Fourier coefficient for PRESENT with R_4 and 10 rounds.



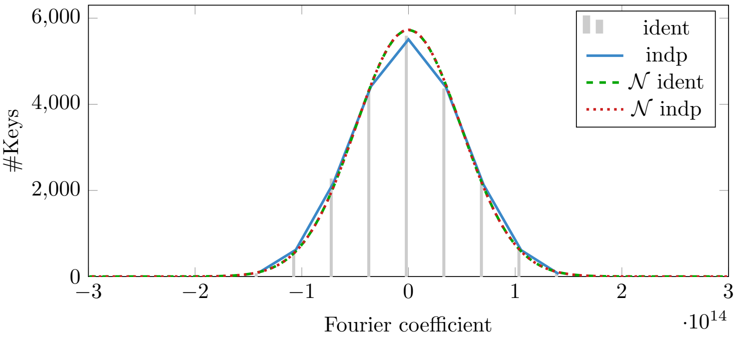
Distribution of Fourier coefficient for PRESENT with R_5 and 10 rounds.



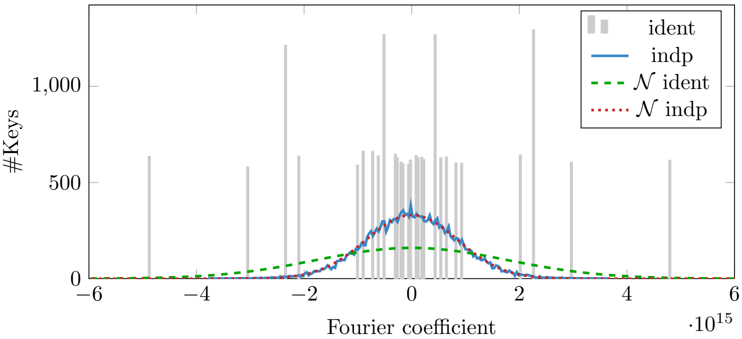
Distribution of Fourier coefficient for Present with R_6 and 10 rounds.



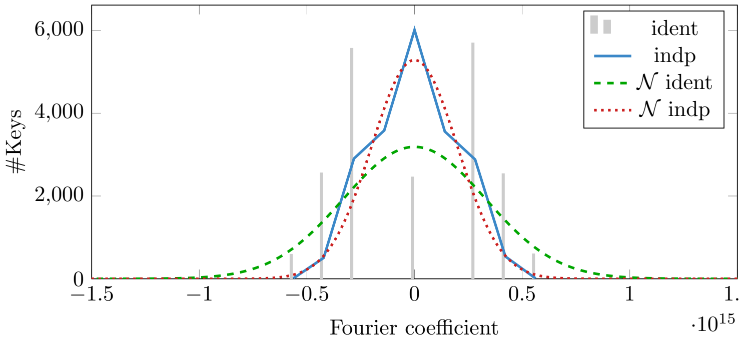
Distribution of Fourier coefficient for Present with R_7 and 10 rounds.



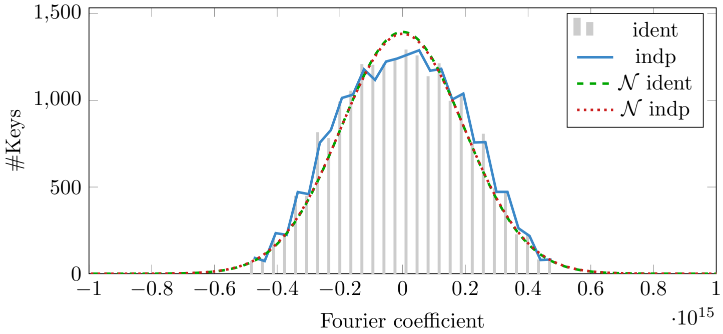
Distribution of Fourier coefficient for Present with R_8 and 10 rounds.



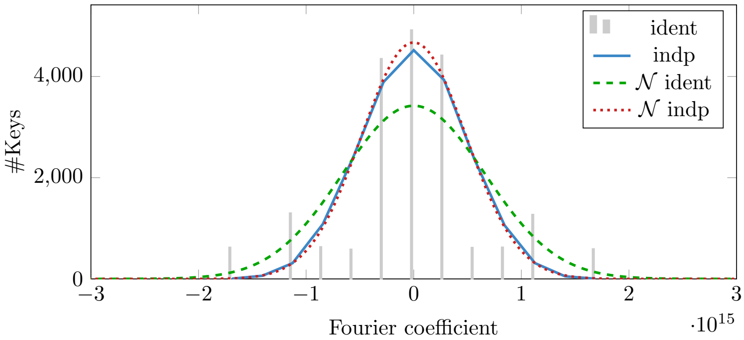
Distribution of Fourier coefficient for Present with R_9 and 10 rounds.



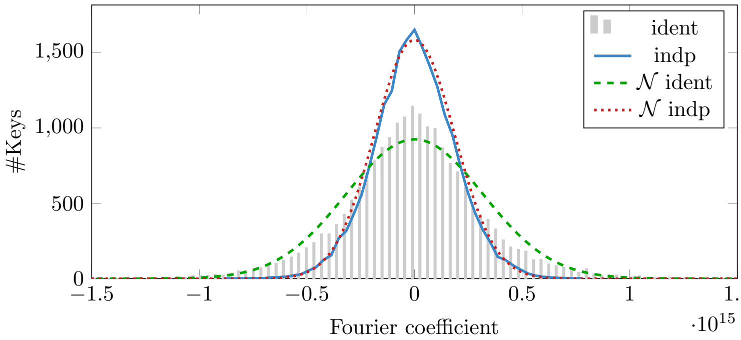
Distribution of Fourier coefficient for PRESENT with R_{10} and 10 rounds.



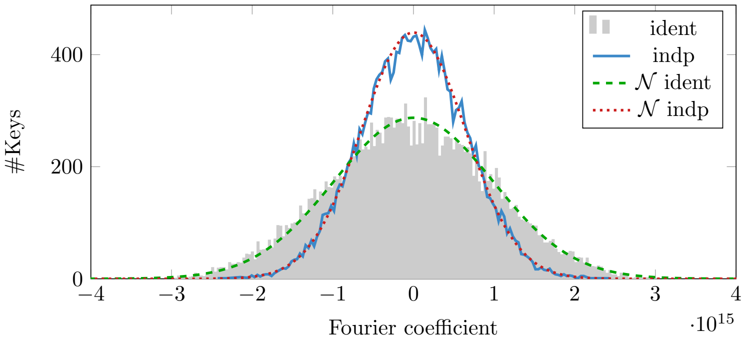
Distribution of Fourier coefficient for PRESENT with R_{11} and 10 rounds.



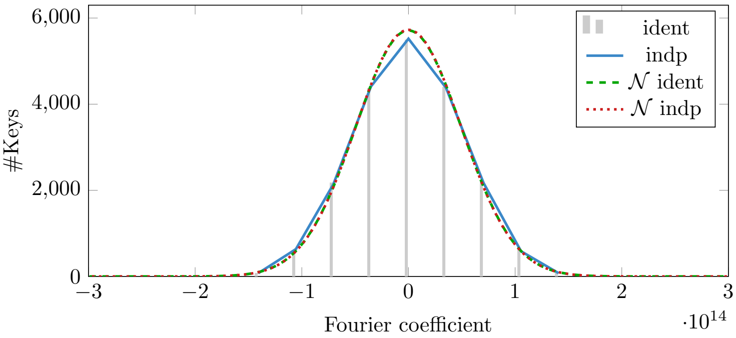
Distribution of Fourier coefficient for PRESENT with R_{12} and 10 rounds.



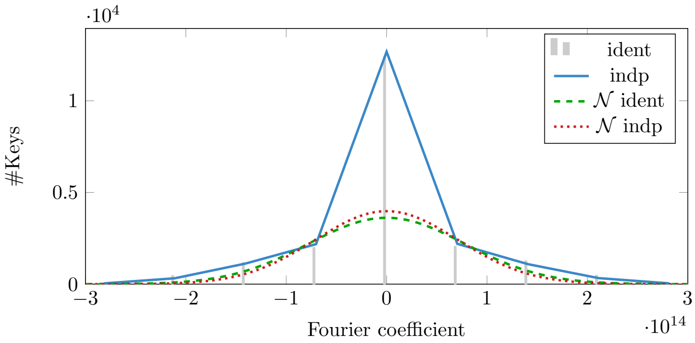
Distribution of Fourier coefficient for PRESENT with R_{13} and 10 rounds.



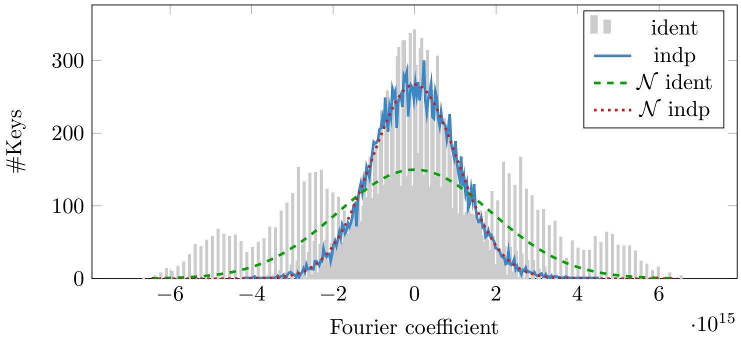
Distribution of Fourier coefficient for Present with R_{14} and 10 rounds.



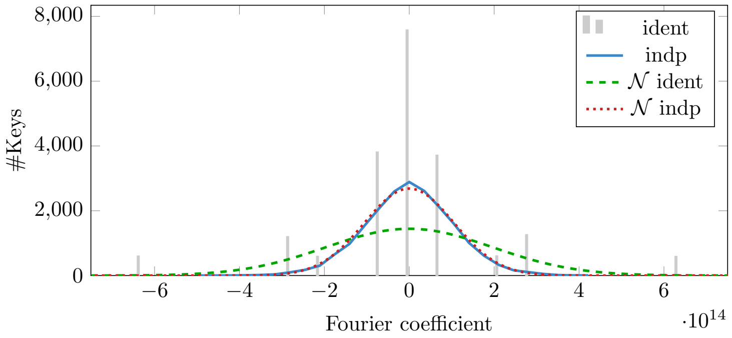
Distribution of Fourier coefficient for Present with R_{15} and 10 rounds.



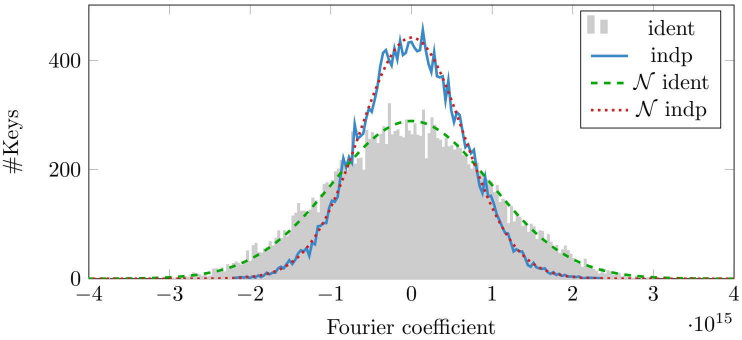
Distribution of Fourier coefficient for Present with R_{16} and 10 rounds.



Distribution of Fourier coefficient for Present with R_{17} and 10 rounds.



Distribution of Fourier coefficient for Present with R_{18} and 10 rounds.



Distribution of Fourier coefficient for Present with R_{19} and 10 rounds.