Improving Linear Key Recovery Attacks using Walsh Spectrum Puncturing

Antonio Flórez-Gutiérrez[0000−0001−7749−8925], Yosuke Todo[0000−0002−6839−4777]

NTT Social Informatics Laboratories, Japan
antonio.florezgutierrez@gmail.com, yosuke.todo@ntt.com

Abstract. In some linear key recovery attacks, the function which determines the value of the linear approximation from the plaintext, ciphertext and key is replaced by a similar map in order to improve the time or memory complexity at the cost of a data complexity increase. We propose a general framework for key recovery map substitution, and introduce *Walsh spectrum puncturing*, which consists of removing carefully-chosen coefficients from the Walsh spectrum of this map. The capabilities of this technique are illustrated by describing improved attacks on reduced-round Serpent (including the first 12-round attack on the 192-bit key variant), GIFT-128 and Noekeon, as well as the full DES.

Keywords: Linear cryptanalysis, Serpent, GIFT, Noekeon, DES

1 Introduction

Linear cryptanalysis [37] is one of the most popular techniques in the analysis of symmetric cryptographic primitives such as block ciphers. It exploits the statistical bias or correlation of one or more *linear approximations*, which are linear combinations of bits of the input and output of the cipher. These linear approximations can be extended over additional rounds by guessing all possible values of a segment of the key and computing the experimental value of the correlation for each one, as we expect the correct guess to exhibit a larger bias.

Algorithm 2. The linear key recovery attack was introduced by Matsui as Algorithm 2 [37]. There have been multiple improvements, such as the introduction of a distillation phase [38] and the fast Walsh transform or FFT technique [24]. Recently, the Walsh transform pruning approach was introduced [30]. The map describing the value of the linear approximation as a function of the plaintext, ciphertext and key is the *key recovery map*. In [30], the complexity of the attack is highly-dependant on the structure of the non-zero values of this map.

Modifying the key recovery map. There are several examples of substitution of the key recovery map for an approximation which lowers the attack complexity, which can be another Boolean function which is highly correlated to the original [2, 11, 5], or a function which rejects some plaintext-ciphertext pairs [39, 11, 30]. This substitution is compensated by increasing the data complexity.

Our contribution. We propose a statistical model for key recovery map approximation which generalises the aforementioned situations and which can be used to compute the required data complexity increase. We introduce a third type of key recovery map approximation, which we call *Walsh spectrum puncturing*. It consists of removing nonzero coefficients of the Walsh spectrum to reduce the cost of the pruned Walsh transform-based attack of [30] and other key recovery algorithms. We find that removing a fraction of ε of the squared 2-norm of the Walsh spectrum (that is, deleting Walsh coefficients so that the sum of their squares is a proportion ε of the total sum), the data complexity must be increased by a factor of $\frac{1}{1-\varepsilon}$. We also describe some puncturing strategies which can be used in common block cipher cryptanalysis scenarios.

As applications, we present improved attacks against Serpent [7, 8], GIFT-128 [4], NOEKEON [27], and DES [1], as summarized in Table 1. Of particular siginificance is, to the best of our knowledge, the first key recovery attack on 12-round Serpent-192. We also improve the best linear attack against GIFT-128, although the best attacks on GIFT-128 are differential rather than linear [47]. Nevertheless, unlike differential cryptanalysis, linear cryptanalysis is still applicable when GIFT-128 is used in COFB mode. The attack on this setting is improved from 16 to 17 rounds. We reduce the memory complexity of the best attack on DES [30] from 3.3TB to 186.1 GB. We also provide the best attack against 12-round NOEKEON in terms of data and time complexities.

Paper layout. Section 2 includes preliminary notions on pseudoboolean functions and their spectra, as well as linear cryptanalysis. Section 3 introduces the statistical model for key recovery map approximation, applies it to spectrum puncturing, and shows some validation experiments. Section 4 discusses some puncturing strategies for common cipher constructions. Sections 5, 6, 7 and 8 briefly describe the applications to Serpent, GIFT-128, the DES, and NOEKEON, respectively. These attacks are explained in more detail in the Appendices.

2 Preliminaries

This section covers notions used in the paper, including some definitions and notations about Boolean functions and their Walsh spectra as covered in books like [20, 43], and some essential concepts on linear cryptanalysis.

2.1 Binary Vector Spaces

We denote the field with two elements as $\mathbb{F}_2 = \{0, 1\}$. For clarity, we use the typeface x, y, u, v to denote vectors of binary vector spaces $\mathbb{F}_2^l$, and the typeface $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$ to denote (column) vectors of real vector spaces $\mathbb{R}^l$. We use x_i and $\mathbf{x}[i]$ to denote the ith coordinates of a binary vector x and a real vector $\mathbf{x}$, respectively. The rightmost, least significant bit of x is x_0. The top coordinate of $\mathbf{x}$ is $\mathbf{x}[0]$. The sum in $\mathbb{F}_2^l$ is denoted by $+$ and by $\oplus$ when confusion is possible

Table 1: Comparison of attacks on the application target ciphers.

Target	Attack	Rds.	Complexity Data	Time	Memory	P_S	Source
Serpent (192-bit)	Linear	11	$2^{121.23}$ KP	$2^{121.23}$	2^{108}	79%	[24, 40]
	Diff-lin	11	$2^{125.7}$ CC	$2^{125.7}$	$2^{99.00}$	85%	[36]
	Linear†	**12**	$2^{127.5}$ KP	$2^{189.74}$	$2^{133.00}$	80%	Sect. 5
	Linear	**12**	$2^{127.5}$ KP	$2^{189.74}$	$2^{182.00}$	80%	Sect. 5
Serpent (256-bit)	Multdim-lin	12	$2^{125.8}$ KP	$2^{253.8}$	$2^{125.8}$	79%	[41, 40, 18]
	Multdim-lin	12	$2^{125.8}$ KP	2^{242}	2^{236}	79%	[41, 40, 18]
	Diff-lin	12	2^{127} CC	2^{251}	2^{127}	77%	[36, 18]
	Diff-lin†	12	$2^{127.92}$ CP	$2^{233.55}$	$2^{127.92}$	10%	[18]
	Diff-lin†	12	$2^{125.74}$ CP	$2^{236.91}$	$2^{125.74}$	10%	[18]
	Diff-lin†	12	$2^{118.40}$ CP	$2^{242.93}$	$2^{118.40}$	10%	[18]
	Linear	12	$2^{125.16}$ KP	$\mathbf{2^{214.36}}$	$2^{125.16}$	81%	Sect. 5
	Linear	12	$2^{126.30}$ KP	$\mathbf{2^{210.36}}$	$2^{125.16}$	80%	Sect. 5
GIFT-128 (General)	Differential	27	$2^{123.53}$ CP	$2^{124.83}$	$2^{80.00}$	-	[47]
	Linear	25	$2^{124.75}$ KP	$2^{126.77}$	$2^{96.00}$	50%	[46]
	Linear	25	$2^{125.75}$ KP	$2^{127.77}$	$2^{96.00}$	75%	[46]
	Linear	25	$\mathbf{2^{123.02}}$ KP	$\mathbf{2^{124.61}}$	$2^{112.00}$	**80%**	Sect. 6
GIFT-128 (COFB)	Linear⋆	16	$2^{62.10}$ KP	$2^{122.80}$	$2^{62.10}$	80%	[46]
	Linear	**17**	$2^{62.10}$ KP	$2^{125.09}$	$2^{62.10}$	80%	Sect. 6
DES‡	Differential	Full	$2^{47.00}$ CP	$2^{37.00}$	$\mathcal{O}(1)$	58%	[12]
	Linear	Full	$2^{43.00}$ KP	$2^{39.00}$	$2^{26.00}$	50%	[38]
	Multiple-lin	Full	$2^{42.78}$ KP	$2^{38.86}$	$2^{30.00}$	85%	[16]
	Conditional-lin	Full	$2^{42.00}$ KP	$2^{42.00}$	$2^{28.00}$	90%	[11]
	Linear	Full	$2^{41.62}$ KP	$2^{41.76}$	$\mathbf{2^{34.54}}$	70%	Sect. 7
	Linear	Full	$2^{41.50}$ KP	$2^{42.13}$	$2^{38.75}$	70%	[30]
Noekeon	Linear	12	$2^{122.35}$ KP	$2^{123.82}$	$2^{121.00}$	80%	[19]
	Linear	12	$\mathbf{2^{119.55}}$ KP	$\mathbf{2^{120.63}}$	$\mathbf{2^{115.00}}$	80%	Sect. 8

† The attack assumes that a-bit advantage of the subkey implies a-bit advantage of the master key without any extra cost. ‡ For the DES application, the data collection cost is excluded from the time complexity for historical reasons. For other applications, the time includes the data collection. ⋆ We have corrected the memory complexity for the sake of comparison. The authors of [46] insisted that the memory complexity is 2^{47}. However, the attack accesses each plaintext-ciphertext pair multiple times. Therefore, storing the data is necessary. We contacted the authors and confirmed that they did not consider the cost of storing the data.

and in cipher specifications. The inner product of binary vectors is

$$\langle x, y\rangle = \sum_{i=0}^{l-1} x_i \cdot y_i.$$

The inner product is linear: $\langle x+y, z\rangle = \langle x, z\rangle\langle y, z\rangle$ and $\langle x, y+z\rangle = \langle x, y\rangle\langle x, z\rangle$. If $\langle x, y\rangle = 0$, we say that x and y are orthogonal and write $x \perp y$.

2.2 Pseudoboolean Functions and their Walsh Spectra

A *pseudoboolean function* is a map $f : \mathbb{F}_2^l \longrightarrow \mathbb{R}$. If $f(x) \in \{1, -1\}$ for all $x \in \mathbb{F}_2^l$, it is a Boolean function. This is because we can identify $0 \in \mathbb{F}_2$ with $(-1)^0 \in \mathbb{R}$ and $1 \in \mathbb{F}_2$ with $(-1)^1 \in \mathbb{R}$ so that addition in $\mathbb{F}_2$ is the same as multiplication in $\mathbb{R}$. Pseudoboolean functions form a real vector space of dimension 2^l, which is denoted $\mathbb{R}\mathbb{F}_2^l$ and can be identified with $\mathbb{R}^{2^l}$. Given a pair of pseudoboolean functions $f, g : \mathbb{F}_2^l \longrightarrow \mathbb{R}$, their inner product and 2-norm are:

$$\langle f, g\rangle = \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} f(x)g(x), \qquad \|f\|_2 = \sqrt{\langle f, f\rangle}.$$

If f, g are traditional Boolean functions, their inner product is often denoted $\mathrm{cor}(f, g)$ and called *correlation*. The 2-norm of Boolean functions is always 1. If $\langle f, g\rangle = 0$, we say that f and g are orthogonal and write $f \perp g$.

The *Hadamard basis* of $\mathbb{R}\mathbb{F}_2^l$ consists of the *parity functions* $\mathbf{h}_u : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2$, where $\mathbf{h}_u(x) = (-1)^{\langle u, x\rangle}$. It satisfies $\mathbf{h}_u \perp \mathbf{h}_v$ if $u \neq v$, and $\langle \mathbf{h}_u, \mathbf{h}_v\rangle = 1$ if $u = v$. Given $f : \mathbb{F}_2^l \longrightarrow \mathbb{R}$, its *Walsh spectrum* is the map $\widehat{f} : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ given by

$$\widehat{f}(u) = \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} (-1)^{\langle u, x\rangle} f(x).$$

The Walsh spectrum $\widehat{f}$ is the representation of f in the Hadamard basis:

$$f = \sum_{u \in \mathbb{F}_2^l} \widehat{f}(u)\mathbf{h}_u$$

A pseudoboolean function is *balanced* if $\widehat{f}(0) = \sum_{x \in \mathbb{F}_2^l} f(x) = 0$.

The Walsh spectrum of a function defined in $\mathbb{F}_2^l$ can be obtained with the fast Walsh transform algorithm [25] in $l2^l$ additions and subtractions.

The Walsh spectrum follows several properties:

Involutivity:	$\hat{\hat{f}} = 2^{-l} f,$
Linearity:	$\widehat{af + bg} = a\widehat{f} + b\widehat{g},$
Parseval Identity:	$\|f\|_2 = \sqrt{2^l}\|\widehat{f}\|_2,$
Plancherel identity:	$\langle f, g\rangle = 2^l\langle \widehat{f}, \widehat{g}\rangle.$

Given $f, g : \mathbb{F}_2^l \longrightarrow \mathbb{R}$, their *convolution* is a map $(f * g) : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ given by

$$(f * g)(k) = \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} f(k) g(x + k).$$

The *convolution theorem* states that

$$\widehat{f * g} = \widehat{f} \odot \widehat{g},$$

where $\odot$ denotes the component-wise product of real vectors. Given g, we can compute $f * g$ by applying the fast Walsh transform on g, multiplying by $\widehat{f}$ component-wise, and applying the fast Walsh transform again.

Given $f : \mathbb{F}_2^l \longrightarrow \mathbb{R}$, if we assume that $x \in \mathbb{F}_2^l$ is uniformly distributed, then $f(x)$ is a real random variable whose mean and variance are:

$$\begin{aligned}
\mathrm{E}[f(x)] &= \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} f(x) = \widehat{f}(0), \\
\mathrm{Var}(f(x)) &= \mathrm{E}[f(x)^2] - \mathrm{E}[f(x)]^2 = \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} f(x)^2 - \widehat{f}(0)^2 \\
&= \sum_{u \in \mathbb{F}_2^l \setminus \{0\}} \widehat{f}(u)^2 = \|f\|_2^2 - \widehat{f}(0)^2, \\
\mathrm{Cov}(f(x), g(x)) &= \mathrm{E}[f(x)g(x)] - \mathrm{E}[f(x)]\mathrm{E}[g(x)] = \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} f(x)g(x) - \widehat{f}(0)\widehat{g}(0) \\
&= \langle f, g \rangle - \widehat{f}(0)\widehat{g}(0) = 2^l \langle \widehat{f}, \widehat{g} \rangle - \widehat{f}(0)\widehat{g}(0)
\end{aligned}$$

If f is balanced, then $\mathrm{E}[f(x)] = \widehat{f}(0) = 0$. For traditional balanced functions, since $\|f\|_2 = 1$, we conclude that $\mathrm{Var}(f(x)) = 1$ and $\mathrm{Cov}(f(x), g(x)) = \mathrm{cor}(f, g)$.

2.3 Vectorial Boolean Functions

Given a vectorial Boolean function $f : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2^m$, its correlation matrix or Walsh spectrum is a $2^l \times 2^m$ matrix containing the spectra of all of its components $f_v : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2$, $f_v(x) = \langle v, f(x) \rangle$:

$$\widehat{f}(u, v) = \widehat{f_v}(u) = \frac{1}{2^l} \sum_{x \in \mathbb{F}_2^l} (-1)^{\langle u, x \rangle \oplus \langle v, f(x) \rangle}.$$

The spectrum of a composition $f = f_r \circ \cdots \circ f_1$ is matrix product of the spectra:

$$\widehat{f} = \widehat{f_r} \times \cdots \times \widehat{f_1}, \qquad \widehat{f}(u, v) = \sum_{u_1} \cdots \sum_{u_{r-1}} \widehat{f_1}(u, u_1) \cdots \widehat{f_r}(u_{r-1}, v).$$

The correlation matrix of a map f consisting of the parallel application of several functions $f_1, \ldots, f_r$ is the Kronecker product of the correlation matrices:

$$\widehat{f} = \widehat{f_1} \otimes \cdots \otimes \widehat{f_r}, \qquad \widehat{f}(u_1 | \ldots | u_r, v_1 | \ldots | v_r) = \widehat{f_1}(u_1, v_1) \cdots \widehat{f_r}(u_r, v_r).$$

2.4 Linear Appproximations

Let $E_K : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ be a block cipher with key K. A *linear approximation* ν is a pair of masks $\alpha, \beta \in \mathbb{F}_2^n$. The evaluation of the linear approximation is the XOR of the inner product of the input and output masks by the plaintext and ciphertext, respectively: $\nu(p, c) = \langle \alpha, p \rangle \oplus \langle \beta, c \rangle$. Its *correlation* is:

$$\mathrm{cor}_K(\alpha, \beta) = \frac{1}{2^n} \sum_{p \in \mathbb{F}_2^n} (-1)^{\langle \alpha, p \rangle \oplus \langle \beta, E_K(p) \rangle} = \widehat{E_K}(\alpha, \beta).$$

The correlation is key-dependent, and thus follows some statistical distribution over the keyspace. The average is usually zero due to positive and negative correlations cancelling each other, so we use the *expected linear potential* [42]:

$$\mathrm{ELP}(\alpha, \beta) = \mathrm{E}\left[\mathrm{cor}_K(\alpha, \beta)^2\right] = \mathrm{Var}\left(\mathrm{cor}_K(\alpha, \beta)\right) + \mathrm{E}\left[\mathrm{cor}_K(\alpha, \beta)\right]^2.$$

Computing the ELP of a given approximation is generally a difficult problem. For *key-alternating block ciphers*, which feature multiple rounds which consist of a round subkey addition and a round function F, a *linear trail* [26] is defined as a particular sequence of linear approximations of the round function ($\alpha = \alpha_0, \alpha_1, \ldots, \alpha_r = \beta$). The correlation of the approximation for a given key is the sum of the *correlation contributions* of all its linear trails.

$$\mathrm{cor}_K(\alpha, \beta) = (-1)^{\langle \alpha_0, K_0 \rangle \oplus \cdots \oplus \langle \alpha_r, K_r \rangle} \prod_{i=1}^{r} \widehat{F}(\alpha_{i-1}, \alpha_i).$$

For some keys, the signs of these contributions cancel each other, and for others they add up to a larger value. Under some key schedule assumptions [42],

$$\mathrm{ELP}(\alpha, \beta) = \sum_{\alpha_1 \in \mathbb{F}_2^n} \cdots \sum_{\alpha_{r-1} \in \mathbb{F}_2^n} \prod_{i=1}^{r} \widehat{F}(\alpha_{i-1}, \alpha_i)^2.$$

We note that finding a collection of high-correlation trails is often enough to obtain an accurate lower bound for the ELP. A noise component 2^{-n} which accounts for any unknown trails [14] is often included too.

2.5 Key Recovery Linear Attack Scenario

Since this work focuses on the key recovery step of linear attacks, we focus on the functions which relate the plaintext, the ciphertext, the key guess, and the value of a linear approximation through the following abstraction:

Definition 1 (Attack scenario) *Consider that as part of a linear key recovery attack, the adversary has to compute the experimental correlation of a linear approximation for a given number of key guesses. Let $x \in \mathbb{F}_2^l$ denote the concatenation of the segments of the plaintext and ciphertext which influence the value of this linear approximation. We also consider that there is an external key guess*

$k_{\text{ext}} \in \mathbb{F}_2^l$ which is xored to this text segment, and an internal key guess k_{int} which is not. Let $f_0(x)$ be a separate plaintext-and-ciphertext-only term which is not influenced by the key. The linear approximation can be written as

$$\nu(p, c, k_{\text{ext}}, k_{\text{int}}) = f_0(p, c) \cdot f(x \oplus k_{\text{ext}}, k_{\text{int}}). \tag{1}$$

This function is called the key recovery map*. The attacker is given a list $\mathcal{D}$ of N plaintext-ciphertext pairs $(p, c = E_K(p))$ which are generated with a secret key K. By examining the key schedule, the attacker can construct a list $\mathcal{K}$ of L valid key guesses $(k_{\text{int}}, k_{\text{ext}})$, and a list $\mathcal{K}_{\text{int}}$ of size L_{int} with the permitted values of k_{int}. The aim of the attacker is to compute the experimental correlations*

$$\widetilde{\text{cor}}(k_{\text{ext}}, k_{\text{int}}) = \frac{1}{N} \sum_{(p,c)\in\mathcal{D}} f_0(p, c) \cdot f(x(p, c) \oplus k_{\text{ext}}, k_{\text{int}}) \tag{2}$$

for all $(k_{\text{ext}}, k_{\text{int}}) \in \mathcal{K}$, which can either be used directly or processed further.

There are several algorithms which compute the experimental correlations.

Matsui's Algorithm 2 [37]. Initialise an array which will store $\widetilde{\text{cor}}$ to zero. Then, for each plaintext-ciphertext pair, each of the entries is either incremented or decremented individually. The total time complexity is thus $\mathcal{O}(NL)$.

Algorithm 2 with distillation [38]. Algorithm 2 evaluates f multiple times for the same input. Matsui proposed an improved version with two stages:

1. **Distillation phase:** A table $\mathbf{a}$ of size 2^l is initialised to zero. For each plaintext-ciphertext pair, we increment or decrement one position to obtain

$$\mathbf{a}[x] = \sum_{\substack{(p,c)\in\mathcal{D} \\ x(p,c)=x}} f_0(p, c).$$

   The distillation table contains all the relevant information about the data sample, and can be constructed in $\mathcal{O}(N)$ time.
2. **Analysis phase:** For each key guess $(k_{\text{ext}}, k_{\text{int}}) \in \mathcal{K}$, we have:

$$\widetilde{\text{cor}}(k_{\text{ext}}, k_{\text{int}}) = \frac{1}{N} \sum_{x\in\mathbb{F}_2^l} f(x + k_{\text{ext}}, k_{\text{int}}) \cdot \mathbf{a}[x].$$

   Each key guess takes 2^l operations, so the total cost is $\mathcal{O}\left(2^l L\right)$.

The total time complexity is $\mathcal{O}(N) + \mathcal{O}\left(2^l L\right)$.

Fast Walsh transform attack [24]. A further attack algorithm was introduced by Collard et al.. We first note that the array $\mathbf{a}$ can be interpreted as a function $\mathbf{a} : \mathbb{F}_2^l \longrightarrow \mathbb{R}$. We momentarily fix the value of k_{int}, and define $\widetilde{\text{cor}}_{k_{\text{int}}} : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ as $\widetilde{\text{cor}}_{k_{\text{int}}}(k_{\text{ext}}) = \widetilde{\text{cor}}(k_{\text{ext}}, k_{\text{int}})$ and $f_{k_{\text{int}}} : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ as $f_{k_{\text{int}}}(x) = f(x, k_{\text{int}})$. Under this notation, $\widetilde{\text{cor}}_{k_{\text{int}}}$ is the convolution of $f_{k_{\text{int}}}$ and $\mathbf{a}$:

$$\widetilde{\text{cor}}_{k_{\text{int}}}(k_{\text{ext}}) = \frac{1}{N} \sum_{x \in \mathbb{F}_2^l} f_{k_{\text{int}}}(x + k_{\text{ext}}) \cdot \mathbf{a}[x] = \frac{1}{N} \left(f_{k_{\text{int}}} * \mathbf{a} \right)(k_{\text{ext}}).$$

This suggests that it can be evaluated efficiently using the convolution theorem:

1. Construct the distillation table $\mathbf{a}$ as in the previous attack algorithm.
2. Evaluate $\widehat{f_{k_{\text{int}}}}$ and $\widehat{\mathbf{a}}$ using the fast Walsh transform algorithm [25].
3. Multiply the previous vectors component-wise.
4. Apply the fast Walsh transform again to obtain $N 2^{-l} \widetilde{\text{cor}}_{k_{\text{int}}}$.

Except for the computation of $\widehat{\mathbf{a}}$, each of the steps 2 to 4 has to be repeated for each of the L_{int} guesses of k_{int}. The cost of each fast Walsh transform is $l2^l$ additions. The time complexity of the attack is thus $\mathcal{O}(N) + \mathcal{O}\left(L_{\text{int}} l 2^l\right)$.

Pruning-based attacks. Several improvements to this algorithm make use of pruning techniques on the fast Walsh transform [31, 30], that is, of optimised Walsh transform algorithms which can be used when the nonzero inputs or the desired outputs are restricted. A brief description of the attack algorithm of [30] can be found in Appendix A. In summary, it was shown that when the support of the Walsh spectrum $\widehat{f_{k_{\text{int}}}}$ is covered by an affine subspace of dimension d, the complexity can be reduced to $\mathcal{O}(N) + \mathcal{O}\left(L_{\text{int}} d 2^d\right)$. By using the linearity of the convolution, this technique can also be applied when the Walsh spectrum lies on the union of T such subspaces, at a cost of $\mathcal{O}(TN) + \mathcal{O}\left(\sum_i L_{\text{int},i} d_i 2^{d_i}\right) + \mathcal{O}(TL)$. Since the pruned fast Walsh transform can accommodate restrictions in both the input and the output, this complexity can be enhanced further by also accounting for the structure of the plaintext material and the key guesses.

2.6 Distribution of the Experimental Correlation

The probability of success of a linear key recovery attack depends on the statistical distribution of the key recovery statistic for both correct and incorrect key guesses. This paper follows the framework which is described in [15]. A sampling correction coefficient B is considered, which is 1 in the classical known plaintext scenario and $\frac{2^n - N}{2^n - 1}$ if the plaintexts are assumed to be distinct.

The wrong key recovery statistic is normally distributed (Theorem 2 in [15]):

$$\widetilde{\text{cor}}_W \sim \mathcal{N}\left(0, \frac{B}{N} + 2^{-n}\right). \tag{3}$$

We assume that these are statistically independent for different wrong key guesses.

For the right key recovery statistic, we consider two possibilities:

– If the approximation has no dominant linear trails, its correlation follows a normal distribution (usually with mean $c = 0$). Per Theorem 5 in [15]:

$$\widetilde{\mathrm{cor}}_R \sim \mathcal{N}\left(c, \frac{B}{N} + \mathrm{ELP} - c^2\right). \tag{4}$$

– If there is a single dominant trail, the keyspace can be separated into two disjoint parts of equal size so that (Theorem 4 in [15]):

$$\widetilde{\mathrm{cor}}_R \sim \mathcal{N}\left(\pm c, \frac{B}{N} + \mathrm{ELP} - c^2\right), \tag{5}$$

in each of the subsets ($+c$ in one and $-c$ in the other). The overall distribution of the statistic is thus a bimodal distribution with two peaks.

These distributions can be used to deduce the probability of success of the attack (see Section 2 of [15]). For the purposes of this paper, we instead focus on how to compensate the data complexity of a modified linear attack so that the resulting probability distributions match those of the original attack.

3 Approximating the Key Recovery Map

There are several examples in the literature in which the key recovery map f is substituted for another map g which "approximates" it in such a way that the impact on the data complexity can be predicted. For example, f may be substituted for another Boolean function g which is highly correlated with f (as in [2, 11, 5] and others), and the correlation is incorporated into the correlation of the distinguisher and used to determine the new data complexity. We can also consider that g is a copy of f which rejects some specific inputs (as in [39, 30, 18]), which amounts to sieving the plaintexts for each key guess. Assuming the correlation of the linear approximation is the same within the remaining pairs, the data complexity is compensated by the inverse of the proportion of rejected plaintexts. In all of these applications, g is chosen so that the key recovery becomes less costly, for example by having a smaller effective input space.

In [30], the rejected inputs are chosen specifically with the aim of reducing the dimension of the support of the Walsh spectrum. However, in order to nullify one coefficient of the Walsh spectrum, the whole spectrum has to be modified. In particular, forcing several Walsh coefficients to be zero often means rejecting all the inputs. This is the motivation for the question of whether it is possible to modify the Walsh spectrum directly, even if the resulting key recovery map is not a Boolean function in the traditional sense.

We introduce a generalisation of these existing techniques in which f is replaced by an arbitrary approximation g, as well as a statistical analysis of the effect on the attack's data complexity. One specific instance is *Walsh spectrum puncturing*, which consists of removing "inconvenient" coefficients from the Walsh spectrum of f to obtain g. This is motivated by the fact that the attack complexity of [30] is highly dependant on the structure of the support of $\widehat{f}$.

3.1 Effect on the Data Complexity

Let $f : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2$ be the key recovery map of a linear attack for a specific internal key guess (more generally, f can be real-valued). We also consider the approximating pseudoboolean function $g : \mathbb{F}_2^l \longrightarrow \mathbb{R}$. For simplicity, we assume that both f and g are balanced, that is, $\mathrm{E}[f(x)] = \widehat{f}(0) = \mathrm{E}[g(x)] = \widehat{g}(0) = 0$. By projecting g orthogonally onto f, we obtain the following decomposition:

$$g = \frac{\langle f, g\rangle}{\|f\|_2^2} f + g^{\perp}, \tag{6}$$

where $g^{\perp}$ is orthogonal to f, that is, both components are uncorrelated as random variables. We note that the orthogonality of both components also means that

$$\|g\|_2^2 = \frac{\langle f, g\rangle^2}{\|f\|_2^2} + \|g^{\perp}\|_2^2,$$

from which we deduce that the variance of $g^{\perp}$ is $\|g^{\perp}\|_2^2 = \|g\|_2^2 - \langle f, g\rangle^2 / \|f\|_2^2$.

We denote the alternative key recovery statistic which uses g instead of f by $\widetilde{\mathrm{cor}}^g(k)$. This statistic can also be separated into two orthogonal components, one of which is a scaled copy of the original key recovery statistic:

$$\widetilde{\mathrm{cor}}^g(k) = \frac{1}{N} \sum_{(p,c)\in\mathcal{D}} g(x(p,c) \oplus k) = \frac{\langle f, g\rangle}{\|f\|_2^2} \widetilde{\mathrm{cor}}^f(k) + \widetilde{\mathrm{cor}}^{g^{\perp}}(k).$$

Assuming the statistical distribution of $g^{\perp}$ under the attack sample is the same as for a uniformly-distributed input, we can prove the following:

Theorem 2 *the distributions of the right-key and wrong-key key recovery statistics using the approximation g of the key recovery map f (both assumed balanced) can be approximated by the normal distributions*

$$\widetilde{\mathrm{cor}}_R^g \sim \mathcal{N}\left(\frac{\langle f, g\rangle}{\|f\|_2^2} c, \|g\|_2^2 \frac{B}{N} + \frac{\langle f, g\rangle^2}{\|f\|_2^2} (\mathrm{ELP} - c^2)\right), \tag{7}$$

$$\widetilde{\mathrm{cor}}_W^g \sim \mathcal{N}\left(0, \|g\|_2^2 \frac{B}{N} + \frac{\langle f, g\rangle^2}{\|f\|_2^2} 2^{-n}\right). \tag{8}$$

Proof. Since the experimental correlation statistic is a (scaled) sum of equally-distributed independent random variables, we can assume that it is normally distributed, and we only have to determine its expected value and variance:

$$\begin{aligned} \mathrm{E}_{\mathcal{D},K}\left[\widetilde{\mathrm{cor}}_R^g\right] &= \mathrm{E}_{\mathcal{D},K}\left[\frac{\langle f, g\rangle}{\|f\|_2^2} \widetilde{\mathrm{cor}}_R^f\right] + \mathrm{E}_{\mathcal{D},K}\left[\widetilde{\mathrm{cor}}_R^{g^{\perp}}\right] \\ &\simeq \frac{\langle f, g\rangle}{\|f\|_2^2} c + \mathrm{E}_K\left[\mathrm{E}_{\mathcal{D}}\left[\widetilde{\mathrm{cor}}_R^{g^{\perp}}\right]\right] = \frac{\langle f, g\rangle}{\|f\|_2^2} c, \end{aligned}$$

assuming that $\mathrm{E}_{\mathcal{D}}\left[\widetilde{\mathrm{cor}}_R^{g^{\perp}}\right] = \mathrm{E}\left[g^{\perp}(x)\right] = 0$.

$$\begin{aligned}\mathrm{Var}_{\mathcal{D},K}\left(\widetilde{\mathrm{cor}}_R^g\right) &= \mathrm{Var}_{\mathcal{D},K}\left(\frac{\langle f,g\rangle}{\|f\|_2^2}\widetilde{\mathrm{cor}}_R^f\right) + \mathrm{Var}_K\left(\mathrm{E}_{\mathcal{D}}\left[\widetilde{\mathrm{cor}}_R^{g^\perp}\right]\right)\\ &\quad + \mathrm{E}_K\left[\mathrm{Var}_{\mathcal{D}}\left(\widetilde{\mathrm{cor}}_R^{g^\perp}\right)\right] - 2\mathrm{Cov}_{\mathcal{D},K}\left(\frac{\langle f,g\rangle}{\|f\|_2^2}\widetilde{\mathrm{cor}}_R^f, \widetilde{\mathrm{cor}}_R^{g^\perp}\right)\\ &\simeq \frac{\langle f,g\rangle^2}{\|f\|_2^2}\left(\frac{B}{N} + \mathrm{ELP} - c^2\right) + \|g^\perp\|_2^2\frac{B}{N}\\ &= \|g\|_2^2\frac{B}{N} + \frac{\langle f,g\rangle^2}{\|f\|_2^2}(\mathrm{ELP} - c^2),\end{aligned}$$

assuming $\mathrm{Cov}_{\mathcal{D},K}\left(\widetilde{\mathrm{cor}}_R^f, \widetilde{\mathrm{cor}}_R^{g^\perp}\right) = \mathrm{Cov}\left(f(x), g^\perp(x)\right) = 0$. To deduce $\mathrm{Var}_{\mathcal{D},K}\left(\widetilde{\mathrm{cor}}_R^f\right) = \|f\|_2^2\left(B/N + \mathrm{ELP} - c^2\right)$ and $\mathrm{Var}_{\mathcal{D}}\left(\widetilde{\mathrm{cor}}_R^{g^\perp}\right) = \|g^\perp\|_2^2\frac{B}{N}$, we require a similar assumption and the central limit theorem (in the distinct known plaintext case, we need to use a variant of the central limit theorem which accounts for sampling without replacement in finite populations, which is discussed in Appendix B).

The wrong key case can be treated similarly:

$$\mathrm{E}_{\mathcal{D},K}\left[\widetilde{\mathrm{cor}}_W^g\right] \simeq \frac{\langle f,g\rangle}{\|f\|_2^2}\mathrm{E}_{\mathcal{D},K}\left[\widetilde{\mathrm{cor}}_W\right] + \mathrm{E}_K\left[\mathrm{E}_{\mathcal{D}}\left[\widetilde{\mathrm{cor}}_W^{g^\perp}\right]\right] = 0.$$

$$\mathrm{Var}_{\mathcal{D},K}\left(\widetilde{\mathrm{cor}}_W^g\right) \simeq \frac{\langle f,g\rangle^2}{\|f\|_2^2}\left(\frac{B}{N} + 2^{-n}\right) + \|g^\perp\|_2^2\frac{B}{N} = \|g\|_2^2\frac{B}{N} + \frac{\langle f,g\rangle^2}{\|f\|_2^2}2^{-n}. \qquad \square$$

We believe the assumption that $\widehat{\mathrm{cor}}^{g^\perp}$ behaves the same in the data as for a uniform input sample is reasonable because in a realistic attack scenario, as the odds of a random balanced Boolean function being biased in the data are low, the only exception being the key recovery maps of linear approximations. However, we could theoretically find g which approximates the key recovery maps of more than one approximation. Describing this scenario and how it may be exploited in cryptanalysis remains an open problem.

This result can be applied directly in the formulas in Section 2 of [15]. However, there is a handy way of *compensating* the data complexity:

Corollary 3 *The success probability of a linear attack which substitutes the balanced key recovery map f for the balanced approximation g remains the same as long as the corrected data sample size N/B is increased by a factor $1/\rho^2$, where*

$$\rho = \frac{|\langle f,g\rangle|}{\|f\|_2 \cdot \|g\|_2} = \frac{|\mathrm{Cov}(f(x),g(x))|}{\sqrt{\mathrm{Var}(f(x))}\sqrt{\mathrm{Var}(g(x))}}, \tag{9}$$

which is the Pearson correlation coefficient of $f(x)$ and $g(x)$.

Proof. We first consider the case in which the correlation of the linear approximation is normally distributed over the keyspace. Let N/B be the corrected sample size for the base attack. The expected value of $\widetilde{\mathrm{cor}}^f$ is c, and

its variance is $\|f\|_2^2(B/N + \mathrm{ELP} - c^2)$ for the right key case. For the wrong key case the mean is 0 and the variance is $\|f\|_2^2(B/N + 2^{-n})$. Let N^*/B^* be the corrected sample for the attack using g. The expected value and variance of $\widetilde{\mathrm{cor}}^g$ are $\frac{\langle f,g\rangle}{\|f\|_2^2}c$ and $\|g\|_2^2\frac{B^*}{N^*} + \frac{\langle f,g\rangle^2}{\|f\|_2^2}(\mathrm{ELP} - c^2)$ for the right key case and 0 and $\|g\|_2^2\frac{B^*}{N^*} + \frac{\langle f,g\rangle^2}{\|f\|_2^2}2^{-n}$ in the wrong key case. If we take $N^*/B^* = (N/B)/\rho^2$, this variance is $\frac{\langle f,g\rangle^2}{\|f\|_2^2}\left(B/N + \mathrm{ELP} - c^2\right)$ for the right key case and $\frac{\langle f,g\rangle^2}{\|f\|_2^2}\left(B/N + 2^{-n}\right)$ for the wrong key. This means $\widetilde{\mathrm{cor}}^g$ has the same distribution as $\frac{\langle f,g\rangle}{\|f\|_2^2}\widetilde{\mathrm{cor}}^f$ in both the right and wrong key cases. Since substituting the key recovery statistic for a multiple has no effect on the success probability, we conclude that it is the same for both attacks.

The case in which a single dominant trail exists and the correlation distribution is bimodal remains. The keyspace consists of two disjoint parts, and the right key experimental correlation statistic is normally distributed in both. By swapping the sign in one of the parts, both distributions become identical. The squared statistic is thus distributed as if both parts were identical, and the previous reasoning still applies. We note that the case in which a small number of dominant trails exists is not covered by these arguments. □

In the known plaintext scenario, the data complexity N just increases by $1/\rho^2$. In the distinct known plaintext scenario, we must consider that B decreases with N. In order to compensate by increasing the data complexity, the original data complexity N must be increased to N^* so that

$$\frac{(2^n - N^*)N}{(2^n - N)N^*} = \rho^2.$$

We can confirm that Corollary 3 generalises existing techniques:

Boolean function substitution. If g is also a Boolean function, then $\|f\|_2 = \|g\|_2 = 1$, which means that $\rho^2 = \langle f, g\rangle^2 = \mathrm{cor}(f, g)^2$, and the data complexity must be increased by a factor equal to the square of the correlation of f and g.

Plaintext rejection. If g is a copy of f which rejects some of the inputs (that is, $g(x) \in \{f(x), 0\}$ for all x), then $\langle f, g\rangle = \|g\|_2^2 = \frac{1}{2^l}\left|x \in \mathbb{F}_2^l : g(x) \neq 0\right|$, and the increase in data complexity $1/\rho^2$ is the inverse of the fraction of inputs of f which are not rejected by g.

We now provide a brief additional justification for the result using the generalised linear cryptanalysis framework of [6]. The functions f and g define one-dimensional subspaces $U = \mathrm{span}\,\{f\}, V = \mathrm{span}\,\{g\}$ of $\mathbb{F}_2^n$. They define a linear approximation map over the identity $\langle V, U\rangle_{\mathrm{id}}$, whose principal correlation is ρ. When this map is appended to the original approximation using the piling-up lemma, a new approximation for the full cipher is obtained whose correlation is multiplied by ρ, and the data complexity has to be increased by $1/\rho^2$.

3.2 Walsh Spectrum Puncturing

In [30], it is shown that the structure of the support the Walsh spectrum of the key recovery map plays a key role in the time complexity. This suggests a simple way to construct an approximation g of f: take some nonzero Walsh coefficients of f, corresponding to a subset $\mathscr{P} \subseteq \mathbb{F}_2^n$, and set them to zero to obtain g:

Definition 4 (Walsh spectrum puncturing) *Let $f : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2$ be a boolean function and $\widehat{f}$ its Walsh spectrum. A* puncture set *is any subset $\mathscr{P} \subseteq \mathbb{F}_2^l$. To simplify the analysis, we assume that f is balanced ($\widehat{f}(0) = 0$), and $0 \notin \mathscr{P}$. We define the* punctured function of f according to $\mathscr{P}$ *as the pseudoboolean function $g = f - \sum_{u \in \mathscr{P}} \widehat{f}(u)\mathbf{h}_u$. The* puncture coefficient ε *is defined as*

$$\varepsilon = \mathrm{Var}(f - g) = \|f - g\|_2^2 = \sum_{u \in \mathscr{P}} \widehat{f}(u)^2. \tag{10}$$

The proportion of f which remains, $1 - \varepsilon$, is called puncturing correlation*:*

$$1 - \varepsilon = \sum_{u \notin \mathscr{P}} \widehat{f}(u)^2 = \langle f, g \rangle = \mathrm{cor}(f, g).$$

We note that, since $\widehat{f}(0) = 0$, we also have $\widehat{g}(0) = 0$, so g is also balanced.

Corollary 5 *Let $f : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2$ be the balanced key recovery map of a linear attack, and let g be a punctured version with puncture coefficient ε. Substituting f for g in the attack and increasing the (corrected) data sample by a factor $\frac{1}{1-\varepsilon}$ yields an attack with the same success probability and advantage.*

Proof. According to Corollary 3, the data complexity must be increased by

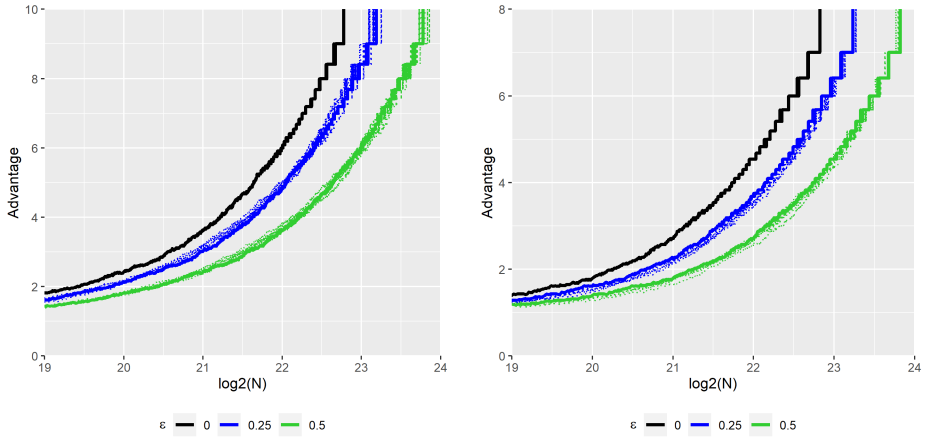
$$\frac{1}{\rho^2} = \frac{\|f\|_2^2 \cdot \|g\|_2^2}{\langle f, g \rangle^2} = \frac{1 \cdot (1 - \varepsilon)}{(1 - \varepsilon)^2} = \frac{1}{1 - \varepsilon}. \qquad \square$$

We note that the increase in data complexity is inversely proportional to the puncturing correlation instead of its square, as may be suggested by intuition.

3.3 Experimental Verification

We perform some experiments to verify the accuracy of the assumptions.

Normally distributed correlation. We consider an attack on 8-round PRESENT (Figure 8 in Appendix D) using a 6-round linear approximation between rounds 1 and 6. The input mask (before SboxLayer in round 1) is `00000000 00A00000`, and that the output mask (before Sboxlayer in round 7) is `00000000 00200020`. It is known that many similar linear trails exist [31], so the approximation's correlation is normally-distributed and centered at zero. Several scenarios with puncture coefficients $\varepsilon = 0.25$ and 0.5 were considered, where the spectrum of one of the four active Sboxes is punctured. They included restricting the spectrum to a hyperplane, removing a single coefficient, or removing a random subset.



(a) 8-round PRESENT. (b) 10-round GIFT-128.

Fig. 1: The results of the puncturing experiments.

Bimodal correlation. We consider 10-round GIFT-128 (Figure 9 in Appendix D) and a 5-round linear trail with correlation 2^{-10} extracted from Figure 4 of [45]. The trail covers rounds 3 to 7 and has input mask `00000000 00000000 00000000 11000000` and output mask `00000000 05000000 00000000 05000000`. There are no additional high-correlation trails. We consider three key recovery rounds on the input side and two on the output side, and puncture the spectra of the four active Sboxes in the inner key recovery rounds 2 and 8.

The attacks were performed for data saple sizes between 2^{19} and 2^{24}, and the experimental correlation of the correct key was compared with that of 2^{10} random wrong keys (2^8 for the GIFT-128 attacks), thus detecting advantage ([44]) values of up to 10 or 8 bits. Each of the punctured attacks was run 5000 times at each data complexity value. The median of the achieved advantages approximates the probability 50% advantage. The advantages are plotted against the data complexity in Figure 1. The black line represents the base attack without puncturing. The dotted blue lines represent the $\varepsilon = 0.25$ experiments and the dotted green lines represent the $\varepsilon = 0.5$ experiments. The predictions corresponding to a data complexity increase from the base case by $1/(1-\varepsilon)$ are shown as continuous lines.

The results indicate that although there is some variability between the results of different scenarios at a given ε, they all follow the model predictions closely. The variability can be attributed to several factors, the most likely of which is the fact that a punctured key recovery map may approximate the key recovery maps of additional linear approximations.

3.4 Relationship to Multiple and Multidimensional Attacks

This subsection briefly discusses how puncturing the Walsh spectrum of the key recovery map compares to other techniques which appear superficially similar, specifically multiple [13] and multidimensional [32, 34, 33] linear cryptanalysis. This is motivated by the observation that puncturing the spectrum of the key recovery map to a single coefficient $\widehat{f}(u)$ essentially amounts to appending one round to the approximation with masks β and u. Although the effective key guess collapses to dimension zero, the key recovery statistic can still be used to distinguish the cipher from a random permutation. Since the expected increase in data complexity is inversely proportional to $\widehat{f}(u)^2$, the data complexity is the same that is predicted by the piling-up lemma. This suggests (punctured) key recovery attacks can be interpreted as using several linear approximations at the same time, each one corresponding to a single coefficient of the Walsh spectrum.

Multiple linear cryptanalysis. Multiple linear cryptanalysis [35, 13], it can use an arbitrary set of linear approximations, and can be applied in both Algorithm 1 and Algorithm 2-type attacks, with Algorithm 2 being the most common. Unlike puncturing, it uses a χ^2 statistic which is not optimised to the joint distribution of the correlations. As a result, the expected data complexity is around $\sqrt{l}/C$ [33], where l is the number of approximations and C is the sum of their squared correlations. It also requires the assumption that the approximations are statistically independent. If the approximations match a punctured key recovery attack, the data complexity of that attack attack would be around $1/C$.

Multidimensional linear cryptanalysis. Introduced by Hermelin et al. [32–34], a multidimensional approximation is a vector space of classical linear approximations, which are not assumed to be independent. In addition to the χ^2 statistic with around $\sqrt{l}/C$ data complexity, the LLR statistic is available, with data complexity $1/C$. It takes the joint distribution and sign of the correlations into account, and the data complexity is similar to a punctured key recovery attack. Since the joint correlation distribution has to be known, in many cases it requires key guessing to determine this distribution. This suggests that puncturing may be interpreted as a hybrid approach in which *some* key material is guessed to determine some partial information about the correlation distribution. We note that in many cases, the way multidimensional approximations are constructed consists of selecting subspaces of the Walsh spectrum of the round function (see for example [22]), which is similar to puncturing. However, multidimensional linear cryptanalysis is limited to vector spaces of linear approximations, while in puncturing the choice of remaining coefficients is arbitrary.

4 Puncturing Walsh Spectra

In the previous section, a theoretical framework which predicts the effect of Walsh spectrum puncturing on the data complexity of a key recovery linear

attack has been laid out. This section deals with the puncturing step itself. Subsection 4.1 provides some intuitive results about puncturing common operations such as the composition and the XOR of Boolean functions. Detailed proofs can be found in Appendix C. Subsection 4.2 features a discussion of several ways of puncturing the spectrum for typical cipher designs.

4.1 Some Useful Results

We first focus on puncturing a composition of functions, which corresponds to a key recovery scenario covering multiple rounds. When puncturing the last function of the composition, the same ρ^2 applies to the whole composition.

Proposition 6 *Let $f_1 : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2^r$ and $f_2 : \mathbb{F}_2^r \longrightarrow \mathbb{R}$ be two Boolean functions, and let $f = f_2 \circ f_1$ be their composition. We also assume that the components of f_1 are all balanced. Let $g_2 : \mathbb{F}_2^r \longrightarrow \mathbb{R}$ be a map which approximates f_2 with Pearson corrrelation coefficient ρ. Then $g = g_2 \circ f_1$ is an approximation of f, and the Pearson correlation coefficient is also ρ.*

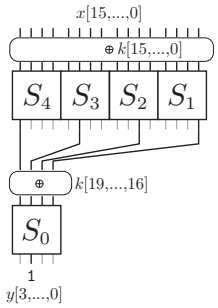
Remark. Puncturing the components of f_1 is also possible, but it requires an abstract definition of the composition of vectorial pseudoboolean functions as the inverse Walsh transform of the matrix product of their Walsh spectra.

Next, we look at puncturing the XOR of several functions (product of real-valued functions). We prove the result in the case in which both functions have the same input domain because it requires the most strict assumptions, but it also holds when the functions have (partially) disjoint input domains.

Proposition 7 *Let $f_1, f_2 : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ be two balanced pseudoboolean functions, and let $f : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2, f = f_1 \cdot f_2$. Let $g_1, g_2 : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ be balanced functions which approximate f_1 and f_2 with correlation coefficients ρ_1 and ρ_2, respectively. Then $g : \mathbb{F}_2^l \longrightarrow \mathbb{R}$, $g = g_1 \cdot g_2$ is an approximation of f, and the compensation factor is $\rho_1\rho_2$, under the assumption that*

$$\begin{aligned}&\mathrm{Cov}(f_1, f_2) = \mathrm{Cov}(f_1^2, f_2^2) = \mathrm{Cov}(g_1, g_2)\\ =&\mathrm{Cov}(g_1^2, g_2^2) = \mathrm{Cov}(f_1, g_2) = \mathrm{Cov}(f_2, g_1) = 0.\end{aligned}$$

We would also like a more general result which permits "step-by-step approximation" which would permit starting from the key recovery map f and taking successive approximations $g_1, g_2, \ldots$ where each function approximates the previous one. Unfortunately, the correlation is not a distance. Indeed, taking $f, g, h : \mathbb{F}_2 \to \mathbb{R}$ with $f = (1, 1)$, $g = (1, 0)$, and $h = (1, -1)$, the Pearson correlation coefficient between f and g and between g and h is $1/\sqrt{2}$, but f and h are incorrelated. However, in the scope of normal applications, there are many instances in which the puncturing coefficients can be multiplied, such as in Proposition 7. This means that we can often puncture the whole key recovery map by puncturing the spectra of its components, such as Sboxes.



	x																ε
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
S_0	1	F	8	3	C	0	B	6	2	5	4	A	9	E	7	D	
f	1	-1	1	1	-1	1	1	-1	1	-1	-1	1	1	-1	-1	-1	
$\widehat{f}$	0	4	0	4	4	0	-4	8	4	0	-4	-8	0	4	0	4	
$\widehat{g_1}$	0	4	0	4	4	0	-4	8	0	0	0	0	0	0	0	0	0.5
g_1	1	-1	0	1	0	0	0	-1	1	-1	0	1	0	0	0	-1	
$\widehat{g_2}$	0	4	0	0	0	0	0	8	0	0	0	-8	0	4	0	0	0.375
g_2	0.5	-0.5	0.5	-0.5	-1	1	1	-1	1	-1	-1	1	0.5	-0.5	0.5	-0.5	
$\widehat{g_3}$	0	0	0	0	0	0	0	8	0	0	0	-8	0	0	0	0	0.5
g_3	0	0	0	0	-1	1	1	-1	1	-1	-1	1	0	0	0	0	
$\widehat{g_4}$	0	4	0	4	4	0	-4	8	4	0	-4	-8	0	4	0	0	2^{-4}
g_4	0.75	-0.75	1.25	0.75	-0.75	0.75	0.75	-0.75	1.25	-1.25	-1.25	1.25	0.75	-0.75	-0.75	-1.25	

Fig. 2: Example of puncturing (the coefficients are multiplied by 16 for clarity).

4.2 Puncturing Strategies

Generic Hyperplane Puncturing. Given a Walsh transform-based linear attack with data complexity $N \leq 2^{n-1}$ and a key recovery input space of dimension l (so the complexity of the transforms is $l2^l$ additions), we can choose any hyperplane of $\mathbb{F}_2^l$ and puncture the spectrum of f to either the hyperplane itself or its complement. Since the sum of the puncture coefficients of both options is 1, one must be equal to or smaller than 1/2. This means it's always possible to construct a pruned transform attack with data complexity at most $2N$ with complexity $\mathcal{O}\left((l-1)2^{l-1}\right)$. This is a generic technique which can be used to rapidly create time-data trade-offs in a wide range of attacks.

Bit Puncturing. Puncturing can be used to find optimal ways to model the cost of ignoring parts of the key recovery map input space. For example, forcefully making one input bit of an Sbox inactive to exclude part of the key guess means forcing a hyperplane of its Walsh spectrum to be zero. It is clear that simply puncturing these coefficients gives the optimal approximation of the Sbox, and the data complexity impact is easy to compute.

Example 1. Consider the key recovery map depicted in Figure 2. The key recovery map to compute y_2 requires all of x and a 20-bit key guess. In the FWT-based procedure, we compute the Walsh transform for every 4-bit internal key guess. The complexity of a Walsh transform attack is thus proportional to $2^4 \times 16 \times 2^{16}$.

We focus on S_0 and let f denote $f(x) = (-1)^{\langle \mathtt{4}, S_0(x) \rangle}$, and $\widehat{f}$ its Walsh spectrum. Using bit puncturing, we exclude the MSB of the input of S_0. We obtain the punctured spectrum $\widehat{g_1}$, and we can compute the associated function g_1. We observe that the MSB is indeed irrelevant, as $g_1(x) = g_1(x \oplus \mathtt{8})$ for all x. In this case g_1 can be described as a traditional Boolean function which rejects some inputs, but this is not always the case. The puncture coefficient is $\varepsilon = 4 \times 2^{-4} + 2^{-2} = 2^{-1}$, i.e. the data must be doubled. We obtain a better time complexity than hyperplane puncturing because it effectively makes S_4 inactive,

thus making the key bits k_{15}, k_{14}, k_{13}, k_{12} and k_{19} unnecesary, and reducing the complexity of the Walsh transforms to $2^3 \times 12 \times 2^{12}$.

More advanced bit puncturing is also possible in the case of ciphers with more complicated linear layers. For example, we may decide that one Sbox will become inactive, and transforming this condition through the linear layer leads to restrictions on the spectra of the Sboxes of the next round. This technique is used in the Serpent attacks of Section 5. In the GIFT attacks of Section 6, the spectrum of the GIFT super Sbox is punctured instead of the spectrum of the Sbox itself. In this case, we don't obtain a traditional Boolean function which rejects some inputs, but a function taking multiple different real values.

LAT Subspace Puncturing. When applying hyperplane puncturing, it is often possible to choose a hyperplane for which ε will be significantly smaller than $1/2$, or to choose a subspace of smaller dimension with $\varepsilon = 1/2$. Quite often, these subspaces can be found by examining the Walsh spectra of the Sbox(es). For example, with 4-bit Sboxes, there often exists an affine subspace of dimension 1 (that is, two coefficients) which concentrates half of the 2-norm of the map.

Example 2. We return to Figure 2. First, we consider puncturing the coefficients in the positions $\{\mathtt{3}, \mathtt{4}, \mathtt{6}, \mathtt{8}, \mathtt{A}, \mathtt{F}\}$ to obtain $\widehat{g_2}$ and the corresponding function g_2. The puncture coefficient is $\varepsilon = 6 \times 2^{-4} = 6/16$, so the data complexity is increased by a factor of $1/(1 - 6/16) = 1.6$. All remaining nonzero coefficients lie in the affine subspace $\mathtt{1} + \operatorname{span}\{\mathtt{6}, \mathtt{A}\}$. Since the dimension is 2, only a 2-bit internal key guess is enough. Thus, the complexity of the Walsh transforms is reduced to $2^2 \times 16 \times 2^{16}$.

Example 3. Puncturing all coefficients of value $\pm 2^{-2}$, we obtain $\widehat{g_3}$ with puncture coefficient $\varepsilon = 8 \times 2^{-4} = 2^{-1}$, which doubles the data. The remaining nonzero coefficients are positions $\mathtt{7}$ and $\mathtt{B}$, which form an affine subspace of dimension 1, i.e., $\mathtt{7} + \operatorname{span}\{\mathtt{C}\}$. The complexity of the Walsh transforms is $2^1 \times 16 \times 2^{16}$.

To explain the reduction of the internal key guess, we show why using g_3 reduces it to 1 bit. Let $a = (a_3, a_2, a_1, a_0)$ be the input to S_0 before xoring the key $k = (k_{19}, k_{18}, k_{17}, k_{16})$. We guess the bit $\langle \mathtt{C}, k \rangle$. When $\langle \mathtt{C}, a \oplus k \rangle = 0$, the input of S_0 can be $\mathtt{00}{*}{*}$ or $\mathtt{11}{*}{*}$, where $*$ are arbitrary. Looking at g_3, the outputs are always 0, so this data is rejected. When $\langle \mathtt{C}, a \oplus k \rangle = 1$, the input of S_0 can be $\mathtt{10}{*}{*}$ or $\mathtt{01}{*}{*}$, and we have $g_3 = (-1)^{\langle 7, a \oplus k \rangle}$. In other words, $\langle 7, k \rangle$ just flips the sign of the correlation, which is unnecessary in many attacks.

We can use LAT subspace puncturing more generally than bit puncturing because it doesn't require taking the outermost key recovery map into consideration. This technique is used in the Serpent attacks of Section 5.

Hamming Weight Puncturing. The plaintext-ciphertext pair rejection technique of [30] is used in cases in which the key recovery map is of the form

$$f(x_0, x_1, \ldots, x_d) = f_2(f_{10}(x_0), f_{11}(x_1), \ldots, f_{1d}(x_d)),$$

which is frequent on attacks on Sbox-based ciphers with bit permutations as linear layers. A subset of inputs of f_2 is selected so that the function which rejects these inputs, f_2^*, verifies $\widehat{f_2^*}(\mathtt{11...1}) = 0$. As a result, the support of the Walsh spectrum of the modified key recovery map can be covered with d subspaces of smaller dimension. However, rejecting these inputs of f_2 modifies its whole Walsh spectrum. Using puncturing, we can simply remove the coefficient $\widehat{f_2}(\mathtt{11...1})$, which has a smaller impact on the data complexity. Furthermore, additional Walsh coefficients can be targeted, such as the ones of Hamming weight $d-1$, to cover the Walsh spectrum with even smaller subspaces. An example of this strategy is the attack on the DES of Section 7.

Example 4. We puncture the coefficient of Hamming weight four (mask `F`) to obtain $\widehat{g_4}$. The corresponding pseudoboolean function is g_4. The puncture coefficient ε is 2^{-4}, and as a result the data complexity increases by a factor of $1/0.9375$. The key recovery map then decomposes into four components with supports of dimension 12. Therefore, the complexity of the Walsh transforms is reduced to $4 \times 2^3 \times 12 \times 2^{12}$.

Generic Puncturing. Finally, it is possible to study the possible propagations of the input and output masks of the linear approximation to obtain a list of the Walsh coefficients of the key recovery map which are larger than a bigger threshold. Then, which Walsh coefficients will be used in the attack can be decided as an optimization problem (for example, trying to obtain the largest possible ρ while keeping the number of active key bits below a certain threshold). This is the approach used in the NOEKEON attack of Section 8.

5 Application to Serpent

Serpent is a 128-bit block cipher and one of the AES competition finalists [7, 8]. Appendix E.1 contains the full specification. It is the subject of substantial cryptanalysis, such as linear [9, 10, 24, 23], multidimensional-linear [32, 34, 41], and differential-linear [10, 36, 18] attacks, which are summarised in Table 1.

On the Key Recovery Attack against 11-Round Serpent-128. Before describing the 12-round attack, we start with an 11-round attack. Figure 3 shows the high-level structure, which uses the 9-round linear trail with correlation 2^{-57} reported in [23] (see Appendix E.2). We also searched for other linear trails with the same input/output mask to evaluate the ELP, but found no such trails with correlation higher than 2^{-64}. Therefore, we estimate the ELP as $2^{-114} + 2^{-128}$, where 2^{-128} is the noise component.

We append one key recovery round to both the plaintext and ciphertext sides, leading to a $1 + 9 + 1 = 11$-round attack. 15 Sboxes and 12 Sboxes are active in the first and last rounds, respectively. This attack structure is identical to the previous attack [24], where the Walsh transform complexity was 108×2^{108}.

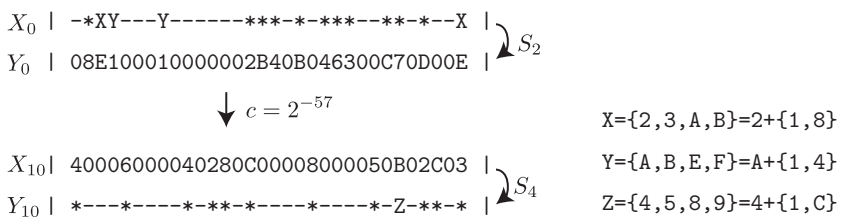


Fig. 3: Key recovery attack against 11-round Serpent

We first apply Walsh transform pruning [30]. The spectra of $\langle \mathtt{E}, S_2 \rangle$, $\langle \mathtt{1}, S_2 \rangle$ and $\langle \mathtt{B}, S_4^{-1} \rangle$ contain only four nonzero coefficients, and they lie in subspaces of dimension 2. The spectrum therefore occupies an $(52 + 46) = 98$-dimensional affine space[1]. The complexity of the Walsh transforms is reduced to 98×2^{98}.

Puncturing Walsh coefficients of the first or last-round Sboxes can reduce the time complexity in return for a slight data increase. However, the dominant complexity of the 11-round attack is the data collection. Therefore, we must reduce the data complexity to improve the 11-round attack. It is possible by switching some active Sboxes of the approximation to punctured key recovery. We adopt such an approach in the application to GIFT and Noekeon. Here, we focus on a simpler and more attractive case with a longer key length.

5.1 Improved Key Recovery Attack against 12-Round Serpent-256

To attack 12-round Serpent-256, we add an additional round of key recovery to the ciphertext side of the 11-round attack. Figure 4 shows these last two key recovery rounds. On the left is the original key recovery map. Since the 11th Sbox in the 12th round is not involved, $128 - 4 = 124$ ciphertext bits are used. Similarly to the 11-round attack, 52 plaintext bits, 52 first-round subkey bits and 46 11th-round subkey bits are involved in the key recovery.

On the right of Figure 4, we use bit puncturing. First, we focus on Sbox 19 at round 11. The input mask is $\mathtt{8}$, and the Walsh spectrum of $\langle \mathtt{8}, S_4^{-1} \rangle$ is

$$(0, 0, 0, 8, 4, -4, 4, 4, 0, 0, 0, -8, 4, -4, 4, 4).$$

When the MSB is ignored, via the linear layer, Sbox 26 in the last round becomes inactive. Therefore, we puncture the five coefficients which involve the MSB. The key guess is reduced by 5 bits, and the data complexity is doubled. Next, we try to exclude the 6th Sbox in the 12th round from the key recovery map. To do it, we puncture the following coefficients in the 11th round: five coefficients whose mask LSB is 1 in the 20th Sbox, four coefficients whose mask MSB is 0 in the 31st Sbox, and four coefficients whose 2nd mask MSB is 0 in the 3rd Sbox.

[1] The authors of [40] make a similar observation and decrease the number of recovered bits from 108 to 98. However, without pruning techniques [30], the complexity of the transforms of their attack is still 108×2^{108}.

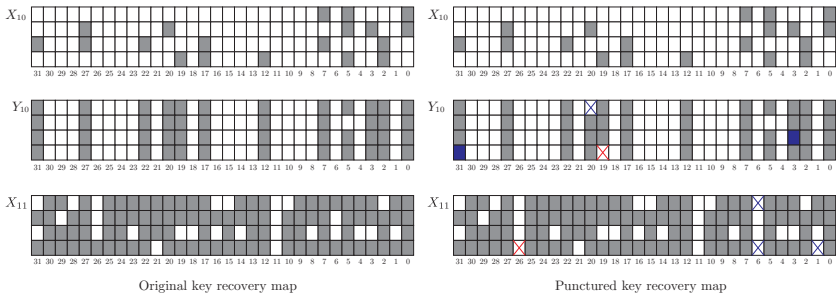


Fig. 4: Ciphertext side of the key recovery attack against 12-round Serpent

The puncturing correlations are 2^{-1}, $3/4$ and $3/4$. Therefore, the key guess is reduced by 7 bits at $2 \times 4/3 \times 4/3 \approx 2^{1.83}$ higher data.

In summary, puncturing reduces the number of active ciphertext bits from 124 to 116 and the key guess by 12 bits. On the other hand, the data complexity is increased by a factor $\rho^{-2} = 2^{2.83}$.

Attack Procedure. We have the following attack procedure using the FWT.

1. Store N known plaintext-ciphertext pairs.
2. Guess the 52 active subkey bits in the 1st round.
   (a) Prepare a distillation table, $\mathbf{a}$, of 2^{116} elements.
   (b) Compute the input parity of the 9-round linear approximation. According to the input parity, we increment or decrement the entry of the distillation table $\mathbf{a}$ indexed by the 116-bit truncated ciphertext.
   (c) Evaluate $\widehat{\mathbf{a}}$ using the FWT.
   (d) Guess the $4 \times 7 + 3 \times 4 + 2 = 42$-bit internal key.
      i. Compute the (punctured) key recovery map $g_{k_{\text{int}}} : \mathbb{F}_2^{112} \longrightarrow \mathbb{R}$.
      ii. Evaluate $\widehat{g_{k_{\text{int}}}}$ by using the FWT.
      iii. Multiply $\widehat{\mathbf{a}}$ with $\widehat{g_{k_{\text{int}}}}$ component-wise.
      iv. Apply the FWT to the resulting table.

The attack procedure above evaluates the experimental correlation of every $52+42+116 = 210$-bit key guess with a time complexity of

$$N + 2^{52} \cdot \big(N + 116 \cdot 2^{116}\texttt{ADD} + 2^{42} \cdot (2^{116}\texttt{PD} + 116 \cdot 2^{116}\texttt{ADD} + 2^{116}\texttt{MUL} + 116 \cdot 2^{116}\texttt{ADD})\big),$$

where `PD`, `ADD`, and `MUL` denote the costs of a 2-round decryption, an addition, and a multiplication, respectively.

With $N\rho^2 = 2^{122.33}$, the success probability with advantage $a = 210$ is higher than 81%. Therefore, we use $N = 2^{125.16}$ known plaintexts. The correct guess ranks among the few highest correlations, and auxiliary techniques recover the rest of the key bits. Assuming that `ADD` and `MUL` are faster than one round function and one encryption, respectively, the time complexity is at most $\frac{2}{12}2^{210} + \frac{116}{12}2^{210} + 2^{210} + \frac{116}{12}2^{210} \approx 2^{214.36}$. The dominant part of the memory complexity is storing the data, $2^{125.16}$.

Further puncturing results in a time-data trade-off. The above attack punctures four coefficients in the 3rd and 31st Sboxes. Puncturing a further four coefficients, leaving only coefficients of value $\pm 2^{-1}$, results in LAT subspace puncturing with $\rho^2 = 2^{-4}$. This variant of the attack has time and data complexities $2^{210.36}$ and $2^{126.30}$, respectively.

5.2 Improved Key Recovery Attack against 12-Round Serpent-192

We next show, to the best of our knowledge, the first key recovery attack on 12-round Serpent-192. Overall it's is almost the same as the attack on 12-round Serpent-256, but we use LAT subspace puncturing to reduce the time complexity further. Returning to the right part of Figure 4, there are 12 active Sboxes in the 11th round. The 5th Sbox is special because a 2-bit guess is enough to determine the parity. We use the same puncturing as above for the 19th and 20th Sbox because it reduces the size of involved ciphertext bits. For the other 9 Sboxes, we puncture all Walsh coefficients with $\pm 2^{-2}$. As a result, we can reduce the number of involved ciphertext bits from 124 to 116, and the $1+1+3\times 9+8 = 37$-bit guess by increasing the data by a factor of 2^{11}. The new attack procedure evaluates the correlation of every $52 + 17 + 116 = 185$-bit guess with a time complexity of

$$N+2^{52}\times\left(N+116\times 2^{116}\mathtt{ADD}+2^{17}\times(2^{116}\mathtt{PD}+116\times 2^{116}\mathtt{ADD}+2^{116}\mathtt{MUL}+116\times 2^{116}\mathtt{ADD})\right).$$

To save some more time, we can use $2^{17+116} = 2^{133}$ memory registers and precompute $g_{k_{\text{int}}}$ and $\widehat{g_{k_{\text{int}}}}$ before guessing the first round subkey. This precomputation reduces the time complexity to around $2^{185} + \frac{116}{12} \times 2^{185} \approx 2^{188.42}$.

With $N = 2^{127.5}$ known plaintexts, the success probability with advantage $a = 3.00$ is higher than 80%. This means that we must keep 2^{185-3} candidates for the 185-bit subkey. If, like [18], we assume that an a-bit advantage on the key guess leads to a an a-bit advantage on the master key without any complexity overhead, the final attack complexity is $2^{188.42} + 2^{189} \approx 2^{189.74}$. In reality, such a conversion is nontrivial due to the nonlinear key schedule. We analyzed the key schedule and found how to convert the partially recovered key to the full master key, although the memory complexity is 2^{185-3}. We show the technique in Appendix E.3.

6 Application to GIFT-128

GIFT is a lightweight block cipher introduced in [4] by Banik et al. There are two versions of GIFT, and we focus on the 128-bit block version, GIFT-128. Please refer to Appendix F.1 for a detailed specification. Similarly to existing attacks [47, 45, 46], we discuss attacks in the general and COFB [21, 3] settings. Please note that we use a more traditional step-by-step key recovery algorithm instead of the FWT. The reason is that the GIFT-128 state is 128 bits, but the round subkey is only 64 bits, which makes step-by-step guessing a better fit.

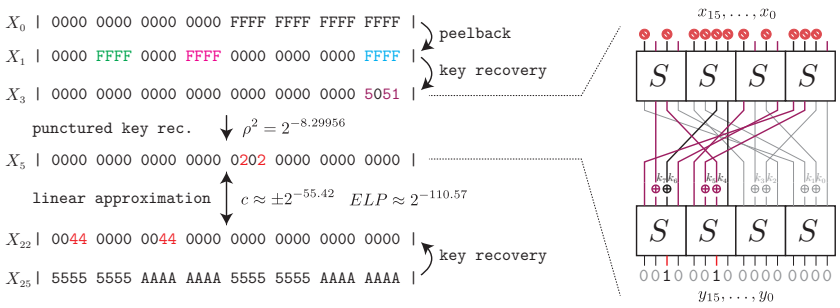


Fig. 5: Overview of the 25-round attack.

Table 2: Punctured Walsh spectrum, where each value is multiplied by 2^{16}.

$k_7 \| k_5 \| k_4$	type	Non-zero coefficients of the spectrum						ρ^2
		1011	1051	4011	4051	5011	5051	
000	A	512	512	2048	2048	−1536	−1536	$2^{-8.30}$
001	B	−1536	−1536	−2048	−2048	512	512	$2^{-8.30}$
010	B	1536	1536	2048	2048	−512	−512	$2^{-8.30}$
011	A	−512	−512	−2048	−2048	1536	1536	$2^{-8.30}$
100	A	−512	−512	−2048	−2048	1536	1536	$2^{-8.30}$
101	B	1536	1536	2048	2048	−512	−512	$2^{-8.30}$
110	B	−1536	−1536	−2048	−2048	512	512	$2^{-8.30}$
111	A	512	512	2048	2048	−1536	−1536	$2^{-8.30}$

6.1 Application to GIFT-128 in the General Setting

Linear cryptanalysis of GIFT-128 has been discussed in [47, 45, 46]. To the best of our knowledge, the best results were reported in [46], which attacks 25 rounds using a 19-round linear approximation.

We propose an improved version with lower data complexity, a high-level overview of which is shown in Figure 5. We switch an active super Sbox of the approximation (between X_3 and X_5) to key recovery and apply bit puncturing on it. Since there is no whitening key before the first Sbox layer, we peel back X_0 to X_1. The key recovery involves 48 bits of X_1 and 64 bits of X_{25}.

Correlation and ELP of the 17-Round Linear Approximation. The correlation of the (shortened) 17-round linear trail is 2^{-56}. We find another linear trail with the same input and output masks and correlation 2^{-57}. However, it involves exactly the same secret key bits so that the correlations of both trails always have the same sign. Considering this situation, we estimated that $ELP \approx 2^{-110.57}$. Furthermore, the correlation distribution is bimodal with peaks at $c \approx \pm 2^{-55.42}$. Please refer to Appendix F.2 for details.

Punctured Super Sbox. We puncture the super Sbox to approximate the parity at X_5 from X_3. The right of Figure 5 represents the active super Sbox. For simplicity, we use $(x_{15}, \ldots, x_0)$, $(y_{15}, \ldots, y_0)$, and $(k_7, \ldots, k_0)$ as the input, output and key of this super Sbox, respectively. The initial goal is to compute $y_{13} \oplus y_9$. To make adding three more key recovery rounds feasible, we use bit puncturing and remove the 11 input bits indexed by $\{15, 13, 11, 10, 9, 8, 7, 5, 3, 2, 1\}$, which also excludes k_6. Table 2 summarizes the Walsh spectrum after puncturing. The puncturing correlation ρ^2 generally depends on the internal key, but here it takes the same value for all key guesses. Moreover, up to a sign swap, there are only two different Walsh spectra, A and B in the Table. The same spectrum always returns the same absolute experimental correlation. Therefore, guessing $k_4 \oplus k_5$ instead of the three bits is enough. In summary, the puncturing increases the data complexity by $2^{8.30}$, but the number of active input bits is reduced from 16 to 5, and the internal key guess is reduced from 4 to 1 bits.

Overview of Results. We use a step-by-step key recovery procedure on the top and bottom three rounds. We first collect the data and prepare a distillation table. Step-by-step round subkey guesses are used to slowly reduce the size of the distillation tables, until a table indexed by the 5-bit input of the punctured super Sbox for each key guess is obtained. The (approximate) experimental correlation is obtained after guessing an additional keybit internal to the super Sbox. Refer to Appendix F.2 for the detailed procedure.

The time complexity of the main attack procedure is $N + 2^{117.40}$. When $N\rho^2 = 2^{114.72}$ with $a = 4.98$-bit advantage, the success probability is higher than 80%. Then, the required data complexity is $N = 2^{114.72} \times 2^{8.30} = 2^{123.02}$ known plaintexts. The total time complexity is $2^{123.02} + (2^{123.02} + 2^{117.40}) + 2^{128-4.98} \approx 2^{124.61}$. The attack requires 2^{112} memory.

6.2 Application to GIFT-128 on the COFB Setting

Linear cryptanalysis is not the best attack strategy against GIFT-128, as differential attacks cover more rounds [47]. Nevertheless, we believe improving linear attacks is meaningful because of the weaker assumption, i.e., known plaintext instead of chosen plaintext. In practice, in many modes of operation it is impossible to choose the input of the underlying block cipher. Linear cryptanalysis is applicable even if GIFT-128 is used on such modes [47, 45, 46].

We consider the COFB setting, where the collectable data is up to the birthday bound. Moreover, we cannot observe the top half of the plaintext because a secret block-dependent mask is XORed. The best existing attack targets 16 rounds using a 10-round linear approximation with 2^{-29} correlation [46].

We improve the attack from 16 to 17 rounds. We first construct an 11-round linear trail, but its correlation is 2^{-34}, which makes it unapplicable because of the birthday bound on the data complexity. Therefore, we switch the first and the last two rounds of the linear approximation to key recovery and apply bit puncturing on these rounds. Unlike in the 25-round attack on the general setting, bit puncturing is used in both the plaintext and ciphertext sides. The

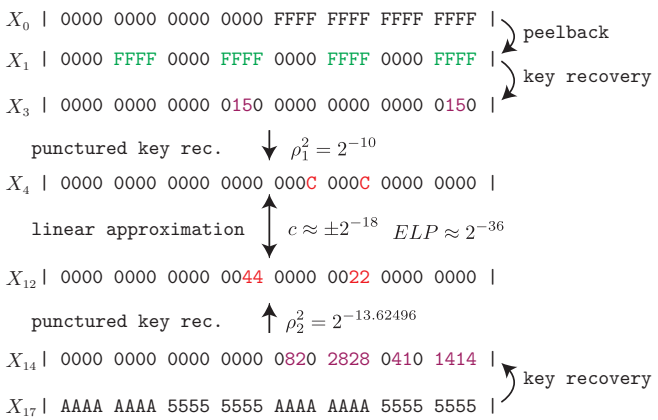


Fig. 6: Overview of the 17-round attack against GIFT-128 on the COFB setting.

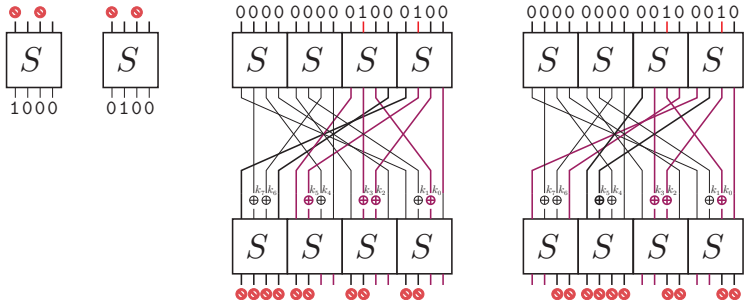


Fig. 7: Puncturing for the attack against GIFT-128 on the COFB setting.

puncturing correlations are $\rho_1^2 = 2^{-10}$ and $\rho_2^2 = 2^{-13.62}$ for the plaintext and ciphertext sides, respectively. Figure 6 shows a high-level overview, and Fig. 7 shows the behaviour of the bit puncturing. Refer to Appendix F.3 for a detailed analysis.

The time complexity of the main attack procedure is $N \times 2^{58} + 2^{115.09}$. Using $2^{62.10}$ known plaintexts (the same data complexity as the existing 16-round attack), and with $a = 2.96$-bit advantage, the success probability is higher than 80%. The total time complexity is $2^{62.10} + (2^{62.10+58} + 2^{115.09}) + 2^{128-2.96} \approx 2^{125.09}$. The main analysis requires a table of size 2^{50}, but since we must store the N plaintext-ciphertext pairs, the memory complexity is N.

7 Application to the Data Encryption Standard

This section describes a variant of the attack on the DES [1] of [30], which uses a 13-round linear approximation which is extended by one key recovery round on the plaintext side and two on the ciphertext side. In rounds 1 and 15, only

Table 3: Part of the Walsh spectrum of S_5, $\widehat{S_5}(\cdot, \mathtt{F})$, highlighting the entries which cover the spectrum in [30] as well as the ones used in the punctured version.

00	0	08	8	10	-40	18	-8	20	0	28	0	30	8	38	0
01	0	09	-8	11	8	19	-8	21	0	29	0	31	8	39	0
02	-8	0A	0	12	0	1A	0	22	-24	2A	8	32	0	3A	-8
03	-8	0B	0	13	0	1B	0	23	-8	2B	8	33	0	3B	8
04	0	0C	-8	14	0	1C	0	24	0	2C	0	34	0	3C	8
05	8	0D	0	15	8	1D	8	25	-8	2D	-8	35	8	3D	0
06	0	0E	-8	16	0	1E	0	26	0	2E	0	36	0	3E	8
07	-8	0F	0	17	8	1F	-8	27	-8	2F	-8	37	-8	3F	0

S_5 is active, while there are six active Sboxes in round 16. It leverages several properties to improve the complexity, such as the bits which are duplicated by the expansion function and the key schedule.

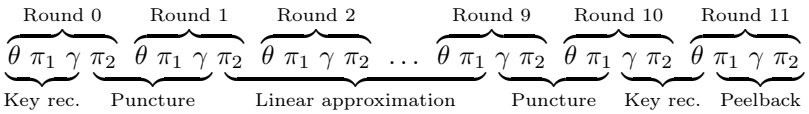
The Walsh spectrum of S_5 in round 15 is studied carefully, as shown in Table 3. The Walsh coefficient associated to the all-ones mask 3F is zero. The key recovery map is thus the (arithmetic) sum of five components, each one corresponding to one of the nonzero coefficients with mask of Hamming weight 5. In each component, one of the six active Sboxes in round 16 becomes effectively inactive. The correlation calculation is also separated into five parts where this inactive Sbox can be used to reduce the complexity.

In our variant, the coefficients corresponding to input masks of Hamming weight 5 are punctured. According to Proposition 6, the resulting puncturing coefficient is $\varepsilon = 0.0781$, and the data complexity increases by a factor of $1.085 \simeq 2^{0.117}$ to $2^{41.62}$. The remaining nonzero coefficients are covered by eight masks of Hamming weight 4 and one mask of Hamming weight 3. For each one of these masks, we now have two instead of one inactive Sbox in round 16. The time and memory complexities are obtained in Appendix G, and are $2^{41.76}$ equivalent encryptions and $2^{34.54}$ registers.

We note that through a very small increase in data complexity by a factor of $2^{0.12}$, we are able to reduce the memory complexity by a much larger factor of $2^{4.21}$. This is especially interesting because the memory complexity decreases from 3.3TB to 186.1GB, which makes the attack significantly more practical.

8 Application to Noekeon

This section describes an improvement on the linear attack on 12-round Noekeon in [19]. The idea is to exclude the first and last rounds from the known 9-round linear approximation. We then apply the puncturing technique to these excluded rounds. The following is a summary of our attack structure.



The linear approximation determines the input mask to γ (the nonlinear layer) in Round 9. We enumerated the Walsh spectrum coefficients of γ which activate at most 12 columns on the peeled-back ciphertext. There are only two such coefficients, and the puncture correlation is $1 - \varepsilon = 2^{-9.68}$. There may be up to $12 \times 4 = 48$ active ciphertext bits, but the dimension of the support is 35.

We next focus on γ in Round 1. We adopted a computer-aided *generic puncturing*. We enumerated all non-zero Walsh spectrum coefficients for which the size of the active plaintext bits is reasonably small. Specifically, we use 460 (out of 10^5) non-zero Walsh spectrum coefficients and puncture the rest. As a result, the puncture correlation is $\rho_1^2 = 2^{-6}$, and it involves $20 \times 4 = 80$ bits of plaintext.

The key recovery is performed using the algorithm of [30]. As a result, the cost of the analysis phase is $2^{118.52}$ for the distillation phase, $2^{116.38}$ for the first FWT, and $2^{112.33}$ for the second FWT. For further detail, refer to Appendix H.

When $N\rho^2 = 2^{103.86}$ with a 8.45-bit advantage, the success probability is higher than 80%. Therefore, the required data complexity is $N = 2^{103.86} \times 2^6 \times 2^{9.68} = 2^{119.54}$. The final time complexity is

$$2^{119.55} + 0.2 \cdot (2^{118.52} + 2^{116.38} + 2^{112.33}) + 2^{128-8.45} \approx 2^{120.63},$$

where we inherit the same constant factor 0.2 as the cost of `ADD` from [19]. The memory complexity is dominated by the distillation table of 2^{115} registers.

9 Conclusion

We have introduced a model which successfully generalises all previous techniques of key recovery map approximation, and which provides a simple formula which describes the data complexity of the modified linear attack. This new model can be applied to Walsh spectrum puncturing, which allows for larger time complexity improvements at a lower data complexity penalty than the existing techniques. Puncturing can be applied in a variety of scenarios, as shown by the applications to Serpent, GIFT-128, the DES and Noekeon. In particular, we have described, to the best of our knowledge, the first attack on 12-round Serpent with 192-bit key. We consider the following open problems:

- *Simultaneous puncturing:* Developing a generalisation of the model so that the new key recovery map can approximate the value of more than one linear approximation may lead to more accurate results, as well as enabling more powerful key recovery attacks.
- *Relationship to distinguishers:* Understanding the relationship between punctured key recovery and known distinguishers such as multidimensional linear cryptanalysis with LLR may deepen our understanding of both puncturing as well as these techniques.
- *Finding the optimal key recovery procedure with puncturing:* For each application, we found an adequate puncturing strategy by hand. Since there is a wide variety of puncturing strategies, we may not have found the optimal strategy. Whether we can develop an automatic tool that can handle this variety is an open problem.

References

1. Data Encryption Standard (DES). Federal Information Processing Standards Publication 46-3, U.S. Department of Commerce, National Institute of Standards and Technology (1977, reaffirmed 1988,1993,1999, withdrawn 2005)
2. Aumasson, J., Fischer, S., Khazaei, S., Meier, W., Rechberger, C.: New features of latin dances: Analysis of salsa, chacha, and rumba. In: Nyberg, K. (ed.) Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5086, pp. 470–488. Springer (2008), https://doi.org/10.1007/978-3-540-71039-4_30
3. Banik, S., Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT-COFB. IACR Cryptol. ePrint Arch. p. 738 (2020), https://eprint.iacr.org/2020/738
4. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Cryptographic Hardware and Embedded Systems - CHES 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 321–345. Springer (2017)
5. Beierle, C., Broll, M., Canale, F., David, N., Flórez-Gutiérrez, A., Leander, G., Naya-Plasencia, M., Todo, Y.: Improved differential-linear attacks with applications to ARX ciphers. Journal of Cryptology 35(4), 29 (2022)
6. Beyne, T.: A geometric approach to linear cryptanalysis. In: Advances in Cryptology - ASIACRYPT 2021, Prooceedings. Lecture Notes in Computer Science, vol. 13090, pp. 36–66. Springer (2021)
7. Biham, E., Anderson, R.J., Knudsen, L.R.: Serpent: A new block cipher proposal. In: Fast Software Encryption 1998, Proceedings. Lecture Notes in Computer Science, vol. 1372, pp. 222–238. Springer (1998)
8. Biham, E., Anderson, R.J., Knudsen, L.R.: Serpent: A proposal for the Advanced Encryption Standard. AES competition (1998)
9. Biham, E., Dunkelman, O., Keller, N.: Linear cryptanalysis of reduced round serpent. In: Fast Software Encryption, 8th International Workshop, FSE 2001, Revised Papers. Lecture Notes in Computer Science, vol. 2355, pp. 16–27. Springer (2001)
10. Biham, E., Dunkelman, O., Keller, N.: Differential-linear cryptanalysis of serpent. In: Fast Software Encryption, 10th International Workshop, Revised Papers. Lecture Notes in Computer Science, vol. 2887, pp. 9–21. Springer (2003)
11. Biham, E., Perle, S.: Conditional linear cryptanalysis - Cryptanalysis of DES with less than 2^{42} complexity. IACR Transactions on Symmetric Cryptology 2018(3), 215–264 (2018)
12. Biham, E., Shamir, A.: Differential cryptanalysis of the full 16-round DES. In: Advances in Cryptology - CRYPTO '92, Proceedings. Lecture Notes in Computer Science, vol. 740, pp. 487–496. Springer (1992)
13. Biryukov, A., Cannière, C.D., Quisquater, M.: On multiple linear approximations. In: Advances in Cryptology - CRYPTO 2004, Proceedings. Lecture Notes in Computer Science, vol. 3152, pp. 1–22. Springer (2004)
14. Blondeau, C., Nyberg, K.: Improved parameter estimates for correlation and capacity deviates in linear cryptanalysis. IACR Transactions on Symmetric Cryptology 2016(2), 162–191 (2016)
15. Blondeau, C., Nyberg, K.: Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. Designs, Codes and Cryptography 82(1-2), 319–349 (2017)

16. Bogdanov, A., Vejre, P.S.: Linear cryptanalysis of DES with asymmetries. In: Advances in Cryptology - ASIACRYPT 2017, Proceedings. Lecture Notes in Computer Science, vol. 10624, pp. 187–216. Springer (2017)
17. Bohrnstedt, G.W., Goldberger, A.S.: On the exact covariance of products of random variables. Journal of the American Statistical Association 64(328), 1439–1442 (1969)
18. Broll, M., Canale, F., David, N., Flórez-Gutiérrez, A., Leander, G., Naya-Plasencia, M., Todo, Y.: New attacks from old distinguishers - Improved attacks on serpent. In: Topics in Cryptology - CT-RSA 2022, Proceedings. Lecture Notes in Computer Science, vol. 13161, pp. 484–510. Springer (2022)
19. Broll, M., Canale, F., Flórez-Gutiérrez, A., Leander, G., Naya-Plasencia, M.: Generic framework for key-guessing improvements. In: Advances in Cryptology - ASIACRYPT 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 453–483. Springer (2021)
20. Carlet, C.: Boolean Functions for Cryptography and Coding Theory. Cambridge University Press (2021)
21. Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M.: Blockcipher-based authenticated encryption: How small can we go? In: Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 277–298. Springer (2017)
22. Cho, J.Y.: Linear cryptanalysis of reduced-round PRESENT. In: Topics in Cryptology - CT-RSA 2010, Proceedings. Lecture Notes in Computer Science, vol. 5985, pp. 302–317. Springer (2010)
23. Collard, B., Standaert, F., Quisquater, J.: Improved and multiple linear cryptanalysis of reduced round Serpent. In: Information Security and Cryptology, Inscrypt 2007, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4990, pp. 51–65. Springer (2007)
24. Collard, B., Standaert, F., Quisquater, J.: Improving the time complexity of Matsui's linear cryptanalysis. In: Information Security and Cryptology - ICISC 2007, Proceedings. Lecture Notes in Computer Science, vol. 4817, pp. 77–88. Springer (2007)
25. Cooley, J., Tukey, J.: An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation 19, 297–301 (01 1965)
26. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Fast Software Encryption 1994, Proceedings. Lecture Notes in Computer Science, vol. 1008, pp. 275–285. Springer (1994)
27. Daemen, J., Peeters, M., Van Assche, G., Rijmen, V.: The NOEKEON block cipher. Proposal to the NESSIE Project (2000)
28. Erdös, P., Rényi, A.: On the central limit theorem for samples from a finite population. Publications of the Mathematical Institute of the Hungarian Academy of Sciences 4(1), 49–57 (1959)
29. Ethier, S.N.: The Doctrine of Chances: Probabilistic Aspects of Gambling. Probability and its Applications, Springer (2010)
30. Flórez-Gutiérrez, A.: Optimising linear key recovery attacks with affine Walsh transform pruning. In: Advances in Cryptology - ASIACRYPT 2022, Proceedings. Lecture Notes in Computer Science, vol. 13794. Springer (2022)
31. Flórez-Gutiérrez, A., Naya-Plasencia, M.: Improving key-recovery in linear attacks: Application to 28-round PRESENT. In: Advances in Cryptology - EUROCRYPT 2020, Proceedings. Lecture Notes in Computer Science, vol. 12105, pp. 221–249. Springer (2020)

32. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional linear cryptanalysis of reduced round Serpent. In: Australasian Conference on Information Security and Privacy, ACISP 2008, Proceedings. pp. 203–215 (2008)
33. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional extension of Matsui's Algorithm 2. In: Fast Software Encryption, FSE 2009, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5665, pp. 209–227. Springer (2009)
34. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional linear cryptanalysis. Journal of Cryptology 32(1), 1–34 (2019)
35. Kaliski, B.J.S., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 26–39. Springer (1994), https://doi.org/10.1007/3-540-48658-5_4
36. Liu, M., Lu, X., Lin, D.: Differential-linear cryptanalysis from an algebraic perspective. In: Advances in Cryptology - CRYPTO 2021, Virtual Event, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 247–277. Springer (2021)
37. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Advances in Cryptology - EUROCRYPT '93, Proceedings. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993)
38. Matsui, M.: The first experimental cryptanalysis of the Data Encryption Standard. In: Advances in Cryptology - CRYPTO '94, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 1–11. Springer (1994)
39. Matsui, M., Yamagishi, A.: A new method for known plaintext attack of FEAL cipher. In: Advances in Cryptology - EUROCRYPT '92, Proceedings. Lecture Notes in Computer Science, vol. 658, pp. 81–91. Springer (1992)
40. McLaughlin, J., Clark, J.A.: Filtered nonlinear cryptanalysis of reduced-round Serpent, and the wrong-key randomization hypothesis. In: IMA International Conference on Cryptography and Coding, IMACC 2013, Proceedings. Lecture Notes in Computer Science, vol. 8308, pp. 120–140. Springer (2013)
41. Nguyen, P.H., Wu, H., Wang, H.: Improving the algorithm 2 in multidimensional linear cryptanalysis. In: Information Security and Privacy - 16th Australasian Conference, ACISP 2011, Melbourne, Australia, July 11-13, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6812, pp. 61–74. Springer (2011)
42. Nyberg, K.: Linear approximation of block ciphers. In: Advances in Cryptology - EUROCRYPT '94, Proceedings. Lecture Notes in Computer Science, vol. 950, pp. 439–444. Springer (1994)
43. O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press (2014)
44. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. Journal of Cryptology 21(1), 131–147 (2008)
45. Sun, L., Wang, W., Wang, M.: Linear cryptanalyses of three AEADs with GIFT-128 as underlying primitives. IACR Transactions on Symmetric Cryptology 2021(2), 199–221 (2021)
46. Sun, L., Wang, W., Wang, M.: Addendum to linear cryptanalyses of three AEADs with GIFT-128 as underlying primitives. IACR Transactions on Symmetric Cryptology 2022(1), 212–219 (2022)
47. Zong, R., Dong, X., Chen, H., Luo, Y., Wang, S., Li, Z.: Towards key-recovery-attack friendly distinguishers: Application to GIFT-128. IACR Transactions on Symmetric Cryptology 2021(1), 156–184 (2021)

A Exploiting Affine Subspaces in the Walsh spectrum

In [30], a modified version of the fast Walsh transform algorithm was proposed which can be applied when it is known that the support of the input function f is contained in an affine subspace of $\mathbb{F}_2^l$, and/or that we only desire to query inputs of $\widehat{f}$ which lie in an affine subspace of $\mathbb{F}_2^l$. This algorithm can be used to leverage affine subspace structures in the Walsh spectrum, the data and the key. We will now describe an attack algorithm for a basic case (which only acknowledges the structure of the Walsh spectrum of the key recovery map) which does not require the use of the pruned version of the fast Walsh transform, although it is used implicitly by referencing Lemma 8 (Lemma 6 in [30]).

Given a subset $Y \subseteq \mathbb{F}_2^l$, we say that x is orthogonal to Y and write $x \perp Y$ if $x \perp y$ for all $y \in Y$. If $X \subseteq \mathbb{F}_2^l$ is another subset, we say that X and Y are orthogonal to each other and write $X \perp Y$ if $x \perp y$ for all $x \in X, y \in Y$. Given a vector subspace $U \subseteq \mathbb{F}_2^l$ of dimension d, the set of all vectors orthogonal to U conforms another vector subspace which is denoted $U^\perp$, and its dimension is always $l - d$. Note that, unlike with real vector spaces, the intersection of U and $U^\perp$ is not necessarily $\{0\}$, and their sum does not necessarily span $\mathbb{F}_2^l$.

Given a vector subspace $U \subseteq \mathbb{F}_2^l$ of dimension d, we say that $x, y \in \mathbb{F}_2^l$ belong to the same coset of U if and only if $x + y \in U$. In other words, we identify all the elements of U with the vector 0, and we consider that any pair of vectors which differs by an element of U is the same. The coset of $x \in \mathbb{F}_2^l$ is denoted by $x + U$. When we do not want to choose a specific representative of a coset, we will denote them by capital letters like X. The sum of the cosets $x + U$ and $y + U$ can be defined as $x + y + U$, which is the same coset independently of the choice of the representatives x, y. This gives the cosets a vector space structure. This quotient space is denoted $\mathbb{F}_2^l/U$, and its dimension is $l - d$.

The following auxiliary lemma is a specific case of Lemma 6 in [30]:

Lemma 8 *Let U be a vector subspace of $\mathbb{F}_2^l$ of dimension d, and let $V = \mathbb{F}_2^l/U^\perp$ be the quotient space of $\mathbb{F}_2^l$ by $U^\perp$. Given $X \in \mathbb{F}_2^l/U^\perp$ and $y \in U$, the inner product $\langle X, y\rangle$ is uniquely defined. Furthermore, we can choose basis $(X_1, \ldots, X_d)$ of $\mathbb{F}_2^l/U^\perp$ and $(y_1, \ldots, y_d)$ of U so that $X_i \perp y_j$ if $i \neq j$ and $X_i \not\perp y_j$ if $i = j$.*

We start by expressing the value of the experimental correlation:

$$\begin{aligned}\widetilde{\mathrm{cor}}_{k_{\mathrm{int}}}(k_{\mathrm{ext}}) &= \frac{1}{N}\sum_{x\in\mathbb{F}_2^l} f_{k_{\mathrm{int}}}(x + k_{\mathrm{ext}}) \cdot \mathbf{a}[x] = \frac{1}{N}\left(f_{k_{\mathrm{int}}} * \mathbf{a}\right)(k_{\mathrm{ext}}) \\ &= \frac{1}{N}\frac{1}{2^l}\sum_{u\in\mathbb{F}_2^l} (-1)^{\langle k_{\mathrm{ext}},u\rangle}\widehat{f_{k_{\mathrm{int}}}}(u)\sum_{x\in\mathbb{F}_2^l}(-1)^{\langle u,x\rangle}\mathbf{a}[x].\end{aligned}$$

We assume that the support of $\widehat{f_{k_{\mathrm{int}}}}$ is contained in the affine subspace $u_0 + U$, where $u_0 \in \mathbb{F}_2^l$ and $U \subseteq \mathbb{F}_2^l$ is a vector subspace of $\mathbb{F}_2^l$ of dimension d. Then

$$\widetilde{\mathrm{cor}}_{k_{\mathrm{int}}}(k_{\mathrm{ext}}) = \frac{1}{N}\frac{1}{2^l}\sum_{u\in U}(-1)^{\langle k_{\mathrm{ext}},u_0+u\rangle}\widehat{f_{k_{\mathrm{int}}}}(u_0+u)\sum_{x\in\mathbb{F}_2^l}(-1)^{\langle u_0+u,x\rangle}\mathbf{a}[x]$$
$$= \frac{1}{N}\frac{1}{2^l}(-1)^{\langle k_{\mathrm{ext}},u_0\rangle}\sum_{u\in U}(-1)^{\langle k_{\mathrm{ext}},u\rangle}\widehat{f_{k_{\mathrm{int}}}}(u_0+u)\sum_{x\in\mathbb{F}_2^l}(-1)^{\langle u,x\rangle\oplus\langle u_0,x\rangle}\mathbf{a}[x].$$

We note that for any given $u \in U$, $\langle u, x\rangle$ is constant for all x belonging to the same coset of $U^\perp$ in $\mathbb{F}_2^l$. This means we can perform the calculation as follows:

$$\widetilde{\mathrm{cor}}_{k_{\mathrm{int}}}(k_{\mathrm{ext}}) = \frac{1}{N}\frac{1}{2^l}(-1)^{\langle k_{\mathrm{ext}},u_0\rangle}\sum_{u\in U}(-1)^{\langle k_{\mathrm{ext}},u\rangle}\widehat{f_{k_{\mathrm{int}}}}(u_0+u)$$
$$\sum_{X\in\mathbb{F}_2^l/U^\perp}(-1)^{\langle u,X\rangle}\sum_{x\in X}(-1)^{\langle u_0,x\rangle}\mathbf{a}[x]$$

This means that in the distillation phase, instead of $\mathbf{a}$, which has 2^l entries, we can build a smaller vector of size 2^d containing the values of $\sum_{x\in X}(-1)^{\langle u_0,x\rangle}\mathbf{a}[x]$ for each $X \in \mathbb{F}_2^l/U^\perp$. Thanks to Lemma 8, we know how to construct a basis of $X \in \mathbb{F}_2^l/U^\perp$ so that $\sum_{X\in\mathbb{F}_2^l/U^\perp}(-1)^{\langle u,x\rangle}\sum_{x\in X}(-1)^{\langle u_0,x\rangle}\mathbf{a}[x]$ can be computed by applying the fast Walsh transform algorithm on this vector of size 2^d.

We next notice that $\langle k_{\mathrm{ext}}, u\rangle$ is similarly constant for all k_{ext} in each coset of $U^\perp$. As a result, all the key guesses in this subset have the same experimental correlation save for a potential sign flip which is given by $\langle k_{\mathrm{ext}}, u_0\rangle$. As before, Lemma 8 ensures that the correlation values in all the cosets can be computed through a Walsh transform of size 2^d.

In summary, the attack proceeds as follows:

1. Choose basis of $\mathbb{F}_2^l/U^\perp$ and U as in Lemma 8. This can be done using an iterative algorithm [30], they can often be found by inspection. These basis are used as indices for the arrays used in the attack.
2. **Distillation phase:** The distillation table is of size 2^d and contains the values $\sum_{x\in X}(-1)^{\langle u_0,x\rangle}\mathbf{a}[x]$ for all $X \in \mathbb{F}_2^l/U^\perp$.
3. **Analysis phase:**
   - Apply the fast Walsh transform on the previous vector.
   - Multiply this vector elementwise by the Walsh spectrum $\widehat{f_{k_{\mathrm{int}}}}(u_0+U)$.
   - Apply the fast Walsh transform again and divide by $N2^l$.
4. **Key guess query phase:** We obtain valid key guesses from the key schedule. For each one, we obtain $\widetilde{\mathrm{cor}}_{k_{\mathrm{int}}}(k_{\mathrm{ext}})$ by multiplying the entry of the correlation vector corresponding to the coset of k_{ext} in $\mathbb{F}_2^l/U^\perp$ by $(-1)^{\langle k_{\mathrm{ext}},u_0\rangle}$.

In practice, we often divide the Walsh spectrum of $f_{k_{\mathrm{int}}}$ into several components whose supports are included in affine subspaces of sifnificatively small dimension, which can sometimes be deduced from the construction of the cipher. In addition, there are cases in which these subspaces are independent of the choice of k_{int}, which means that the distillation phase can still be performed for all of the internal key guesses at the same time. In this case, and if we assume we use T subspaces of dimension no larger than d, the total cost of the attack is $\mathcal{O}(TN) + \mathcal{O}\left(TL_{\mathrm{int}}d2^d\right) + \mathcal{O}(TL)$.

B On Finite Population Sampling without Replacement

When proving Theorem 2 for the distinct known plaintext case, we need to approximate the distribution of the experimental correlation of several functions when they are sampled without replacement from a finite population. In classical key recovery attacks like the ones discussed in [15], the function only takes two values and the hypergeometric distribution can be used. However, with puncturing, more than two values may appear. We use the following version of the central limit theorem for finite population sampling which can be found as Theorem A.2.13 in [29], and which is a version of a result from [28].

Theorem 9 (Central limit theorem for a finite population)
For each $n \geq 1$, let the random vector $(X_{n,1}, X_{n,2}, \ldots, X_{n,2^n})$ have the discrete uniform distribution over all $2^n!$ permutations of the 2^n (not necessarily distinct but not all equal) real numbers $x_{n,1}, x_{n,2}, \ldots, x_{n,2^n}$. Let

$$\mu_n = \mathrm{E}\left[X_{n,1}\right] = \frac{1}{2^n}\sum_{i=1}^{2^n} x_{n,i}$$

and

$$\sigma_n^2 = \mathrm{Var}\left(X_{n,1}\right) = \frac{1}{2^n}\sum_{i=1}^{2^n}(x_{n,i} - \mu_n)^2.$$

Assume that

$$\max_{1\leq i\leq 2^n} \frac{|x_{n,i} - \mu_n|}{\sqrt{2^n\sigma_n^2}} \longrightarrow 0 \text{ as } n \to \infty.$$

Then, with $S_{n,N} = X_{n,1} + X_{n,2} + \cdots + X_{n,N}$, we have:

$$\frac{S_{n,N} - N\mu_n}{\sqrt{NB\sigma_n^2}} \xrightarrow{d} \mathcal{N}(0,1),$$

where $B = \frac{2^n - N}{2^n - 1}$, and assuming $n, N \to \infty$ in such a way that $\frac{N}{2^n} \to \alpha \in (0,1)$.

This theorem provides informal justification for the following approximation:

Corollary 10 *Let $f : \mathbb{F}_2^n \longrightarrow \mathbb{R}$ be a pseudoboolean function, and let*

$$S = \frac{1}{N}\sum_{x\in\mathcal{D}} f(x) \tag{11}$$

be a random variable, where $\mathcal{D}$ is a uniformly sampled subset of $\mathbb{F}_2^n$ with $|\mathcal{D}| = N$. Then, under certain assumptions, we can approximate the distribution of S by

$$S \sim \mathcal{N}\left(\widehat{f}(0), \frac{B}{N}\|f\|_2^2\right).$$

C Proofs of the Results of Subsection 4.1

Proposition 6 *Let $f_1 : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2^r$ and $f_2 : \mathbb{F}_2^r \longrightarrow \mathbb{R}$ be two Boolean functions, and let $f = f_2 \circ f_1$ be their composition. We also assume that the components of f_1 are all balanced. Let $g_2 : \mathbb{F}_2^r \longrightarrow \mathbb{R}$ be a map which approximates f_2 with Pearson corrrelation coefficient ρ. Then $g = g_2 \circ f_1$ is an approximation of f, and the Pearson correlation coefficient is also ρ.*

Proof. We start by looking at the inner product $\langle f, g\rangle$:

$$\begin{aligned}
\langle f, g\rangle &= \sum_{u\in\mathbb{F}_2^l} \widehat{f}(u)\widehat{g}(u) = \sum_{u\in\mathbb{F}_2^l} \left(\sum_{v_1\in\mathbb{F}_2^r} \widehat{f_1}(u,v_1)\widehat{f_2}(v_1)\right)\left(\sum_{v_2\in\mathbb{F}_2^r} \widehat{f_1}(u,v_2)\widehat{g_2}(v_2)\right)\\
&= \sum_{u\in\mathbb{F}_2^l}\sum_{v_1\in\mathbb{F}_2^r}\sum_{v_2\in\mathbb{F}_2^r} \widehat{f_1}(u,v_1)\widehat{f_2}(v_1)\widehat{f_1}(u,v_2)\widehat{g_2}(v_2)\\
&= \sum_{v_1\in\mathbb{F}_2^r}\sum_{v_2\in\mathbb{F}_2^r} \widehat{f_2}(v_1)\widehat{g_2}(v_2)\sum_{u\in\mathbb{F}_2^l} \widehat{f_1}(u,v_1)\widehat{f_1}(u,v_2)\\
&= \sum_{v_1\in\mathbb{F}_2^r}\sum_{v_2\in\mathbb{F}_2^r} \widehat{f_2}(v_1)\widehat{g_2}(v_2)\frac{1}{2^l}\sum_{x\in\mathbb{F}_2^l} \langle f_1(x), v_1\rangle \oplus \langle f_1(x), v_2\rangle\\
&= \sum_{v_1\in\mathbb{F}_2^r}\sum_{v_2\in\mathbb{F}_2^r} \widehat{f_2}(v_1)\widehat{g_2}(v_2)\frac{1}{2^l}\sum_{x\in\mathbb{F}_2^l} \langle f_1(x), v_1\oplus v_2\rangle\\
&= \sum_{v\in\mathbb{F}_2^r} \widehat{f_2}(v)\widehat{g_2}(v) = \langle f_2, g_2\rangle.
\end{aligned}$$

By substituting f for g (that is, substituting f_2 for g_2) in the previous calculation, we similarly deduce that $\|g\|_2^2 = \|g_2\|_2^2$. Together with the fact that $\|f\|_2 = \|f_2\|_2 = 1$ because both are Boolean functions, we deduce

$$\frac{|\langle f, g\rangle|}{\|f\|_2\|g\|_2} = \frac{|\langle f_2, g_2\rangle|}{\|f_2\|_2\|g_2\|_2} = \rho. \qquad \square$$

Proposition 7 *Let $f_1, f_2 : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ be two balanced pseudoboolean functions, and let $f : \mathbb{F}_2^l \longrightarrow \mathbb{F}_2, f = f_1 \cdot f_2$. Let $g_1, g_2 : \mathbb{F}_2^l \longrightarrow \mathbb{R}$ be balancedfunctions which approximate f_1 and f_2 with correlation coefficients ρ_1 and ρ_2, respectively. Then $g : \mathbb{F}_2^l \longrightarrow \mathbb{R}$, $g = g_1 \cdot g_2$ is an approximation of f, and the compensation factor is $\rho_1\rho_2$, under the assumption that*

$$\begin{aligned}
&\mathrm{Cov}(f_1, f_2) = \mathrm{Cov}(f_1^2, f_2^2) = \mathrm{Cov}(g_1, g_2)\\
=&\mathrm{Cov}(g_1^2, g_2^2) = \mathrm{Cov}(f_1, g_2) = \mathrm{Cov}(f_2, g_1) = 0.
\end{aligned} \tag{12}$$

Proof. In order to determine the Pearson correlation coefficient of f and g, we need to compute the variances of f and g as well as their covariance. Since these

are all written as the product of functions, we use the formulas from [17]. We start with the variance of f:

$$\begin{aligned}\mathrm{Var}(f) = \mathrm{Var}(f_1 \cdot f_2) &= \mathrm{Cov}(f_1^2, f_2^2) \\ &+ \left(\ \mathrm{Var}(f_1) + \mathrm{Exp}(f_1)^2\right)\left(\ \mathrm{Var}(f_2) + \mathrm{Exp}(f_2)^2\right) \\ &- \left(\mathrm{Cov}(f_1, f_2) + \mathrm{Exp}(f_1)\mathrm{Exp}(f_2)\right)^2 \\ &= \mathrm{Var}(f_1)\mathrm{Var}(f_2).\end{aligned}$$

Similarly, we can prove:

$$\mathrm{Var}(g) = \mathrm{Var}(g_1)\mathrm{Var}(g_2).$$

Finally, we look at the covariance:

$$\begin{aligned}\mathrm{Cov}(f, g) &= \mathrm{Cov}(f_1 \cdot f_2, g_1 \cdot g_2) \\ &= \mathrm{Exp}(f_1)\mathrm{Exp}(g_1)\mathrm{Cov}(f_2, g_2) + \mathrm{Exp}(f_1)\mathrm{Exp}(g_2)\mathrm{Cov}(f_2, g_1) \\ &+ \mathrm{Exp}(f_2)\mathrm{Exp}(g_1)\mathrm{Cov}(f_1, g_2) + \mathrm{Exp}(f_2)\mathrm{Exp}(g_2)\mathrm{Cov}(f_1, g_1) \\ &+ \mathrm{Cov}(f_1, g_1)\mathrm{Cov}(f_2, g_2) + \mathrm{Cov}(f_1, g_2)\mathrm{Cov}(f_2, g_1) \\ &= \mathrm{Cov}(f_1, g_1)\mathrm{Cov}(f_2, g_2).\end{aligned}$$

From these expressions, we can deduce:

$$\begin{aligned}\rho &= \frac{|\langle f, g\rangle|}{\|f\|_2 \cdot \|g\|_2} = \frac{|\langle f_1, g_1\rangle \cdot \langle f_2, g_2\rangle|}{\|f_1\|_2 \cdot \|f_2\|_2 \cdot \|g_1\|_2 \cdot \|g_2\|_2} \\ &= \frac{|\langle f_1, g_1\rangle|}{\|f_1\|_2 \cdot \|g_1\|_2} \cdot \frac{|\langle f_2, g_2\rangle|}{\|f_2\|_2 \cdot \|g_2\|_2} = \rho_1 \cdot \rho_2. \qquad \square\end{aligned}$$

D Diagrams of the Attacks of the Experiments of Subsection 3.3

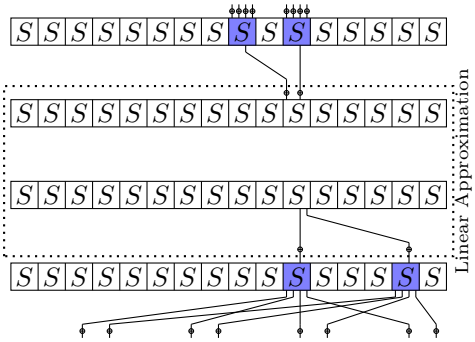


Fig. 8: The attack on 8-round PRESENT used in the experiment.

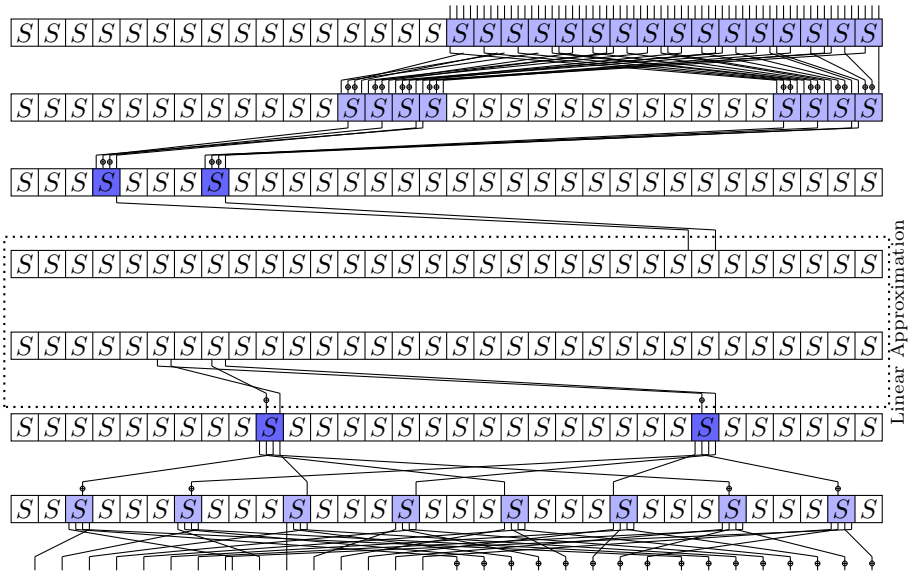


Fig. 9: The attack on 10-round GIFT-128 used in the experiment.

E Supplementary Material for Linear Cryptanalysis against Serpent

E.1 Specification of Serpent

Serpent is a block cipher which was introduced in [7] by Anderson, Biham and Knudsen. In response to feedback on the original Serpent, the authors submitted a revised version to the AES competition [8], and it was selected as one of the finalists. We show the specification of the AES candidate.

Serpent is a 128-bit block cipher adopting a substitution-permutation network (SPN). It accepts 128, 192 or 256-bit keys. The encryption consists of a round function which is iterated 32 times. The round function consists of three layers: a key XORing, an Sbox layer, and a linear layer. The round function uses different Sboxes for each round. Specifically, it uses the following eight Sboxes.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x)$	3	8	F	1	A	6	5	B	E	D	4	2	7	0	9	C
$S_1(x)$	F	C	2	7	9	0	5	A	1	B	E	8	6	D	3	4
$S_2(x)$	8	6	7	9	3	C	A	F	D	1	E	4	0	B	5	2
$S_3(x)$	0	F	B	8	C	9	6	3	D	1	2	4	A	7	5	E
$S_4(x)$	1	F	8	3	C	0	B	6	2	5	4	A	9	E	7	D
$S_5(x)$	F	5	2	B	4	A	9	C	0	3	E	8	D	6	7	1
$S_6(x)$	7	2	C	5	8	4	6	B	E	9	1	F	D	3	A	0
$S_7(x)$	1	D	F	0	E	8	2	B	7	4	C	A	9	3	5	6

Serpent is a bit-slice implementation-friendly cipher, and we show the specification in a bit-slice manner. The 128-bit internal state X is represented by four 32-bit words which are denoted X_0, X_1, X_2 and X_3, with $X_j[i]$ being the i-th leftmost bit of word j. The 32 rounds are numbered 0 to 31. Each round consists of the following three steps:

- **Key XORing**. A 128-bit subkey, K_i, is XORed to the internal state X. Specifically, in round r, compute $X_i \oplus k_{4\times r+i}$ for $i = 0, 1, 2, 3$.
- **Sbox Layer**. At round r, the Sbox layer applies $S_{(r \bmod 8)}$ in a bit-slice manner, i.e., $(X_3, X_2, X_1, X_0) = S_{(r \bmod 8)}(X_3, X_2, X_1, X_0)$, where X_3 is the MSB, and X_0 is the LSB.
- **Linear transformation**. The four 32-bit words are mixed linearly as follows:

$$
\begin{aligned}
&X_0 \leftarrow X_0 \lll 13; \quad X_2 \leftarrow X_2 \lll 3\\
&X_1 \leftarrow X_1 \oplus X_0 \oplus X_2; \quad X_3 \leftarrow X_3 \oplus X_2 \oplus (X_0 \ll 3)\\
&X_1 \leftarrow X_1 \lll 1; \quad X_3 \leftarrow X_3 \lll 7\\
&X_0 \leftarrow X_0 \oplus X_1 \oplus X_3; \quad X_2 \leftarrow X_2 \oplus X_3 \oplus (X_1 \ll 7)\\
&X_0 \leftarrow X_0 \lll 5; \quad X_2 \leftarrow X_2 \lll 22
\end{aligned}
$$

Here $\ll j$ denotes a j-bit left shift and $\lll j$ denotes a j-bit left rotation. At the 31st round, this linear transformation is omitted. Instead, an additional subkey $(k_{128}, k_{129}, k_{130}, k_{131})$ is XORed to the state.

Note that we do not show the initial permutation IP as specified in [7, 8],because it can be omitted in the bit-slice implementation.

Key Schedule. The original Serpent [7] and the AES-competition version [8] have different key schedules. Here, we show the key schedule of the AES candidate [8].

The key schedule accepts a secret key of length 128, 192, or 256 bits. When the length is 128 or 192, the key is padded to 256 bits by adding constant bits. Then, the 256-bit key is written as eight 32-bit words, $(w_{-8}, w_{-7}, \ldots, w_{-1})$ and expanded to an intermediate key, called *prekey*, $w_0, \ldots, w_{131}$ as follows:

$$w_i = (w_{i-8} \oplus w_{i-5} \oplus w_{i-1} \oplus \varphi \oplus i) \lll 11,$$

where $\varphi =$ `0x9e3779b9`. We then build the sequence k_i from w_i using the Sboxes:

$$\begin{aligned}
\{k_0, k_1, k_2, k_3\} &= S_3(w_0, w_1, w_2, w_3) \\
\{k_4, k_5, k_6, k_7\} &= S_2(w_4, w_5, w_6, w_7) \\
&\cdots \\
\{k_{124}, k_{125}, k_{126}, k_{127}\} &= S_4(w_{124}, w_{125}, w_{126}, w_{127}) \\
\{k_{128}, k_{129}, k_{130}, k_{131}\} &= S_3(w_{128}, w_{129}, w_{130}, w_{131})
\end{aligned}$$

E.2 Detail of the 9-Round Linear Approximation by Collard et al.

Collard et al. showed a 9-round linear trail with a correlation of 2^{-57}. Table 4 shows the detail of this 9-round trail. Note that the trail starts from S_3, covering, for example, from the output of round 3 to the input of round 13.

E.3 From Partial Subkey Recovery to Partial Master Key Recovery

Many modern block ciphers adopt relatively simple key schedules. For example, GIFT uses a bit-permutated master key as the subkey. Therefore, a guess of l independent bits of the subkeys can be converted to a guess of l bits of the master key, and an advantage of a bits in the former becomes an advantage of a bits in the latter. The key schedule of Serpent, however, is more complicated, and contains a well-diffused linear layer and a nonlinear layer. Therefore, the transition from a subkey guess to a master key guess is nontrivial.

The key schedule first expands the master key to the prekey sequence linearly. Therefore, an a-bit prekey advantage is an a-bit master key advantage.

Hereinafter, we discuss the case of our key recovery attack against 12-round Serpent-192 (see Fig. 10). Our attack guesses the following subkey bits:

- 52 bits of the subkey $(k_8, k_9, k_{10}, k_{11})$, where we guess all 4 bits of 11 columns but guess only 2 bits of 4 other columns.

Table 4: 9-Round Linear Approximation by Collard et al.

Y00	08E100010000002B40B046300C70D00E	
X01(S3)	0E00F000BF0C00000A00000DE00CC00D	2^{-14}
Y01	0800400091050000010000004100A2004	
X02(S4)	08000000000000000000000000A00040	2^{-6}
Y02	04000000000000000000000000400080	
X03(S5)	04000000000000000000000000000020	2^{-4}
Y03	04000000000000000000000000000080	
X04(S6)	00000000000000000000000800000000	2^{-2}
Y04	00000000000000000000000100000000	
X05(S7)	0000010000A0000000000000000000000	2^{-4}
Y05	00000100001000000000000000000000	
X06(S0)	00000000000000000000010000B0000A00	2^{-5}
Y06	00000000000000000000010000010000100	
X07(S1)	010000B0000B0000A000000000000000	2^{-6}
Y07	01000010000100001000000000000000	
X08(S2)	000A000000000010000B0000B0000B0	2^{-5}
Y08	00010000000000050000100001000010	
X09(S3)	00B0000B000030000B0200E000000100	2^{-11}
Y09	00400004000010000508002000000E00	
X10(S4)	40006000040280C00008000050B02C03	
Total		2^{-57}

- 17 bits of the subkey $(ek_{48}, ek_{49}, ek_{50}, ek_{51}) = L^{-1}(k_{48}, k_{49}, k_{50}, k_{51})$.
- 116 bits of the subkey $(k_{52}, k_{53}, k_{54}, k_{55})$, where we guess all 4 bits in 29 columns. Note that we assume the last round of 12-round Serpent does not contain the linear layer, as does the last round of full Serpent.

In total, we guess 185 subkey bits. With an a-bit advantage, we have 2^{185-a} candidates for this 185-bit segment, one of which will correspond to the correct secret key with high probability.

There are $185-25 = 160$ bits which conform full 4-bit outputs an Sbox of the key schedule nonlinear layer. Therefore, it is easy to compute the corresponding 160 prekey bits by applying the inverse Sbox (blue bits in Figure 10). On the other hand, the remaining 25 bits involve more than 25 prekey bits.

To get an a-bit prekey advantage, we first choose 192 bits of the prekey, always including 16 bits of $(w_8, w_9, w_{10}, w_{11})$ and 112 bits of $(w_{48}, w_{49}, w_{50}, w_{51})$ which are shown as slashed cells in Figure 10. These are enough to deduce the $2 \times 4 + 17 = 25$ problematic subkey bits. The other $192 - 16 - 112 = 64$ bits are arbitrary linearly independent bits such that the full 192 bits determine the full prekey sequence.

Let w_g be the 160 prekey bits shown in blue in Figure 10. Let k_g be the chosen 192 prekey bits, where the last 128 bits are the slashed cells. We now

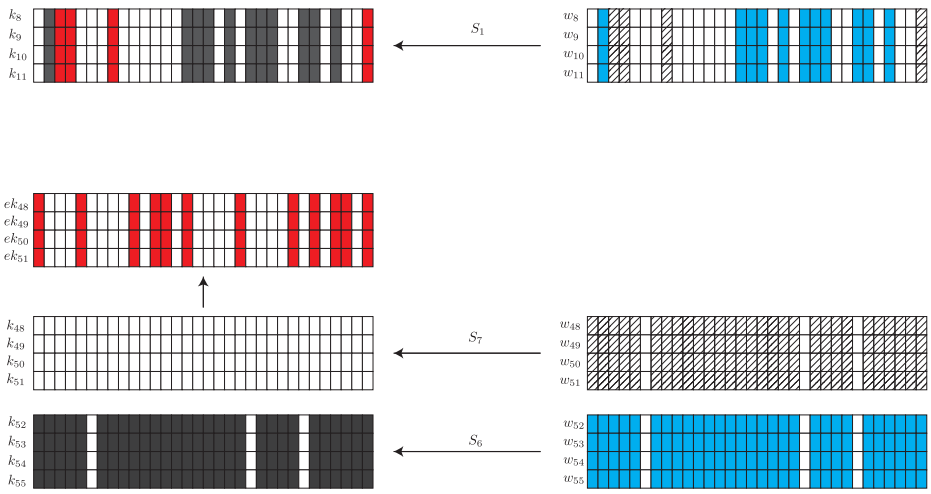


Fig. 10: Analysis of the key schedule of Serpent

construct a linear system $w_g = A \times k_g$, where A is a 160×192 binary matrix. We then get $U \times w_g = (U \times A) \times k_g$, where $(U \times A)$ is an upper triangular matrix.

We now get an a-bit prekey advantage with the following procedure.

1. For each of the 2^{185-a} candidates, compute the 160 blue prekey bits w_g and store the 25 remaining subkey bits and $U \times w_g$. The time complexity is 2^{185-a} and it requires 2^{185-a} memory.
2. We guess the last $(192 - a)$ bits of k_g and compute the 25-bit prekey and the last $(160 - a)$ bits of $U \times w_g$. Looking up the tables above, we get one (on average) solution for the top $(160 - a)$ bits of $U \times w_g$. Then, we compute the master key and run a trial encryption. The time complexity is 2^{192-a}.

The procedure above allows us to get an a-bit prekey (master key) advantage from the a-bit subkey advantage without a large time complexity overhead. Unfortunately, the memory complexity is significantly increased, and reducing it is left as an open problem.

F Supplementary Material for Linear Cryptanalysis against GIFT-128

F.1 Specification of GIFT-128

GIFT is a lightweight block cipher which was introduced in [4] by Banik et al. There are two versions of GIFT; GIFT-64 has a 64-bit block length, and GIFT-128 has a 128-bit block length. Both GIFT-64 and GIFT-128 accept a 128-bit secret key. GIFT is one of the most well-known lightweight block ciphers. Some NIST-LWC candidates use GIFT-128 as an underlying primitive, e.g., GIFT-COFB, which is one of the finalists of the NIST LWC.

The 128-bit internal state S can be represented bit-wise, $S = b_{127}\|b_{126}\|\cdots\|b_0$, or nibble-wise, $S = w_{31}\|w_{30}\|\cdots\|w_0$. GIFT-128 is a 40-round SPN cipher, and the round function consists of the following three steps:

– **SubCells**. The Sbox S is applied to every nibble of the cipher state, i.e., $w_i \leftarrow S(w_i)$ for all $i \in \{0, 1, \ldots, 31\}$.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	1	a	4	c	6	f	3	9	2	d	b	7	5	0	8	e

– **PermBits**. The bit permutation $P(i)$ is applied, i.e., $b_{P(i)} \leftarrow b_i$ for all $i \in \{0, 1, \ldots, 127\}$.

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

– **AddRoundKey**. The round key and round constant are XORed with the internal state. The size of the round key is the half of the block length, i.e., 64 bits in GIFT-128. The round key RK is partitioned into two 32-bit words $RK = U\|V = u_{31}\|\cdots\|u_0\|v_{31}\|\cdots\|v_0$. The two words U and V are XORed to $\{b_{4i+2}\}$ and $\{b_{4i+1}\}$ of the internal state, respectively, i.e.,

$$b_{4i+2} \leftarrow b_{4i+2} \oplus u_i, \qquad b_{4i+1} \leftarrow b_{4i+1} \oplus v_i$$

for all $i \in \{0, 1, \ldots, 31\}$.
A single bit "1" and a 6-bit round constant $C = c_5\|c_4\|\cdots\|c_0$ are XORed into the internal state at bit position 127, 23, 19, 15, 11, 7, and 3, respectively.

Key Schedule. The key schedule accepts a 128-bit key $K = k_7\|k_6\|\cdots\|k_0$, where k_i is a 16-bit word. The round key is extracted from the secret key as follows:

$$RK = U\|V \leftarrow k_5\|k_4\|k_1\|k_0.$$

After extracting a round subkey, the key state is updated as follows:

$$k_7\|k_6\|\cdots\|k_1\|k_0 \leftarrow k_1 \ggg 2\|k_0 \ggg 12\|k_7\|k_6\|k_5\|k_4\|k_3\|k_2.$$

The round constants are generated using a 6-bit affine LFSR, whose state is denoted as $(c_5, c_4, c_3, c_2, c_1, c_0)$. Its update function is

$$(c_5, c_4, c_3, c_2, c_1, c_0) \leftarrow (c_4, c_3, c_2, c_1, c_0, c_5 \oplus c_4 \oplus 1).$$

The six bits are initialized to zero, and updated before use every round.

F.2 Detail of the 25-Round Linear Attack against GIFT-128

Correlation and ELP of Linear Approximation. Although we inherit a 19-round linear approximation proposed in [46], we revisit the approximation because of two reasons: First, we use a 17-round shortened approximation instead of the 19-round approximation, which means that we need to reevaluate the ELP. Second, the estimation of [46] ignores the interaction of the half-size key XORing and the linear hull of the approximation.

The input and output linear masks of the 17-round approximation are

0000 0000 0000 0000 0202 0000 0000 0000,
0044 0000 0022 0000 0000 0000 0000 0000,

respectively. We enumerated all linear trails with correlation larger than 2^{-64}. As a result, we found 98 linear trails, and Table 5 summarizes the distribution.

Table 6 shows the two linear trails with the highest correlation. The trails only differ in one active nibble of `X4` where one has mask `8` and the other has mask `9`. If GIFT-128 used a full 128-bit independent round subkey every round, then the correlations of both trails would be independent. However, GIFT-128 does not XOR any secret key to the MSB and LSB of each nibble. This means

Table 5: Distribution of 17-round linear trails.

$\log_2(\lvert c\rvert)$	-56	-57	-58	-59	-60	-61	-62	-63	-64
# trails	1	1	2	4	7	15	23	27	18

Table 6: Top two linear trails with the input/output linear mask restriction.

X0	00000000000000000202000000000000	2^{-4}
X1	0000A000000000000000A00000000000	2^{-2}
X2	00000000020002000000000000000000	2^{-2}
X3	00220000001100000000000000000000	2^{-6}
X4	80800000808000000000000000000000	2^{-4}
X5	50500000000000005050000000000000	2^{-4}
X6	00000000A000A000000000000A000A000	2^{-4}
X7	00000000002200220000000000000000	2^{-6}
X8	00990000000000000066000000000000	2^{-6}
X9	00000000C000C000000000000000000000	2^{-2}
X10	00000000000000000011000000000000	2^{-3}
X11	00000000000C000000000000000000000	2^{-1}
X12	00000000000200000000000000000000	2^{-1}
X13	00000000000000000020000000100000	2^{-3}
X14	00000000000080800000000000000000	2^{-2}
X15	00050000000000000005000000000000	2^{-4}
X16	00000000400040000000000000000000	2^{-2}
X17	00440000002200000000000000000000	
Total		2^{-56}

X0	00000000000000000202000000000000	2^{-4}
X1	0000A000000000000000A00000000000	2^{-2}
X2	00000000020002000000000000000000	2^{-2}
X3	00220000001100000000000000000000	2^{-6}
X4	90800000808000000000000000000000	2^{-5}
X5	50500000000000005050000000000000	2^{-4}
X6	00000000A000A000000000000A000A000	2^{-4}
X7	00000000002200220000000000000000	2^{-6}
X8	00990000000000000066000000000000	2^{-6}
X9	00000000C000C000000000000000000000	2^{-2}
X10	00000000000000000011000000000000	2^{-3}
X11	00000000000C000000000000000000000	2^{-1}
X12	00000000000200000000000000000000	2^{-1}
X13	00000000000000000020000000100000	2^{-3}
X14	00000000000080800000000000000000	2^{-2}
X15	00050000000000000005000000000000	2^{-4}
X16	00000000400040000000000000000000	2^{-2}
X17	00440000002200000000000000000000	
Total		2^{-57}

both trails involve the same subkey (and round constant), and their correlation contributions always have either the same or opposite sign. We look at the different part of both trails more carefully:

$$\texttt{0x0022} \xrightarrow{Sbox,2^{-2}\times 2^{-2}} \texttt{0x0088} \xrightarrow{bit\ perm.} \texttt{0x8800} \xrightarrow{Sbox,-2^{-1}\times -2^{-1}} \texttt{0x5500}$$

$$\texttt{0x0022} \xrightarrow{Sbox,2^{-4}\times 2^{-2}} \texttt{0x0098} \xrightarrow{bit\ perm.} \texttt{0x9800} \xrightarrow{Sbox,-2^{-2}\times -2^{-1}} \texttt{0x5500}$$

Fortunately for the attacker, the Sbox approximations $\texttt{0x2} \to \texttt{0x8}$ and $\texttt{0x2} \to \texttt{0x9}$ have the same sign, and the trails $\texttt{0x9} \to \texttt{0x5}$ and $\texttt{0x8} \to \texttt{0x5}$ also have the same sign. Therefore, both full trails will always have correlation contributions of the same sign, and the correlation of the linear approximation is enhanced to $\pm(2^{-56} + 2^{-57}) \approx \pm 2^{-55.42}$.

Such effects frequently happen in the enumerated 98 trials. Specifically, there are only eight different active key patterns, the resulting correlations are

$$\pm 2^{-55.42}, \quad \pm 2^{-56.83}, \quad \pm 2^{-57.79}, \quad \pm 2^{-58.42}, \quad \pm 2^{-62.42}, \quad \pm 2^{-62.42}, \quad 0, \quad 0.$$

Therefore, we estimate the ELP as the sum of the squares of the above correlations, which is $2^{-110.57}$.

Attack Procedure. Let $x^r = (x^r_{127}, \ldots, x^r_0)$ be the input of the rth round function, where x^r_0 and x^r_{127} denote the LSB and MSB of x^r, respectively. We

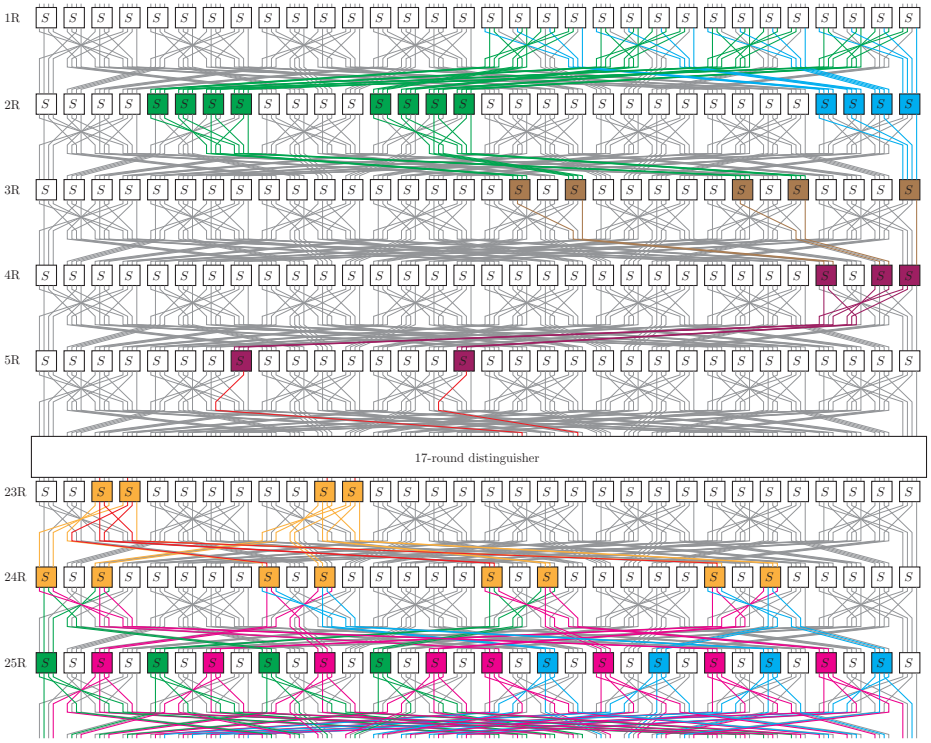


Fig. 11: key recovery map for the improved 25-round attack.

also use $x'^r = (x'^r_{127}, \ldots, x'^r_0) = P^{-1}(x^r)$. The round number r starts from 0, i.e., x^0 denotes the plaintext and x^{25} denotes the ciphertext. Figure 11 shows the key recovery map, and Fig. 12 summarizes the involved master-key bits.

1. We collect N plaintext-ciphertext pairs and store these pairs into a $(48 + 64) = 112$-bit table, according to $(x^1_{111}, \ldots, x^1_{96}, x^1_{79}, \ldots, x^1_{64}, x^1_{15}, \ldots, x^1_0)$ and x^{25}_i for all odd i.
2. We focus on the 19th and 27th Sboxes of the 2nd round and the 31st Sbox of the 25th round (part of Sboxes coloured green). It involves secret key bits indexed by 36, 44, 100, and 108. Then, the 12-bit input can be compressed to the 6-bit output. Therefore, after guessing the 4-bit key, we can compress the 112-bit table into a 106-bit table. The time complexity is $2^4 \times 2^{112} = 2^{116}$.
3. Similar to Step 2, we focus on the green Sboxes step by step. We guess key bits in the 1st and 25th round function. After guessing key bits indexed by

$$\{37, 38, 39, 45, 46, 47, 101, 102, 103, 109, 110, 111\},$$

we have 88-bit tables for each guess of a 16-bit key. The time complexity is $2^{4+4+106} + 2^{8+4+100} + 2^{12+4+94} \approx 2^{114.39}$.

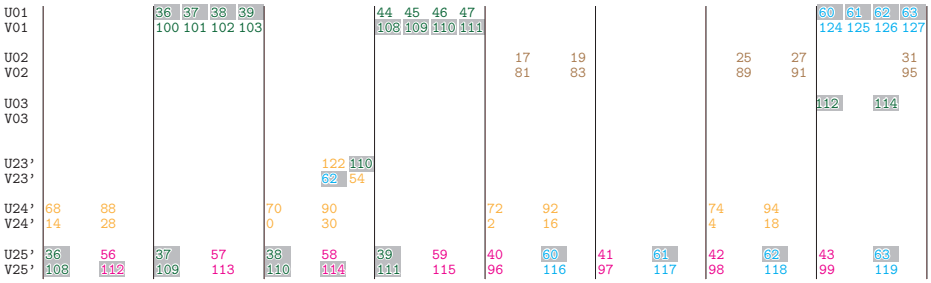


Fig. 12: Summary of involved master-key bits

4. We next focus on Sboxes coloured blue. We guess a 12-bit secret key indexed by
$$\{60, 61, 62, 63, 124, 125, 126, 127, 116, 117, 118, 119\}.$$
Then, we have 68-bit tables for each guess of a 28-bit key. The time complexity is $2^{16} \times 2^{12} \times 2^{88} = 2^{116}$.
5. We focus on Sboxes coloured pink. We guess a 16-bit secret key index by
$$\{56, 57, 58, 59, 112, 113, 114, 115, 40, 41, 42, 43, 96, 97, 98, 99\}.$$
Then, we have 52-bit tables for each guess of a 44-bit key. The time complexity is $2^{28} \times 2^{16} \times 2^{68} = 2^{112}$.
6. We focus on Sboxes coloured yellow. We guess an 18-bit secret key index by
$$\{68, 88, 70, 90, 72, 92, 74, 94, 14, 28, 0, 30, 2, 16, 4, 18, 122, 54\}.$$
Then, we can compute the output parity of the 17-round approximation and construct 20-bit tables for each guess of the 62-bit key. The time complexity is $2^{44} \times 2^{18} \times 2^{52} = 2^{114}$.
7. We focus on Sboxed coloured brown. We guess a 10-bit secret key in the 2nd round and compress the 20-bit table into a 5-bit table. The time complexity is $2^{62} \times 2^{10} \times 2^{20} = 2^{92}$.
8. We now have 5-bit tables for each guess of the 72-bit key. These 5 bits are the input of the punctured super Sbox. We compute the (punctured) correlation by additionally guessing the 1-bit secret key involved in the punctured super Sbox. The time complexity is $2^{72} \times 2^{1} \times 2^{5} = 2^{78}$.

The time complexity of the main attack procedure is

$$N + 2^{116} + 2^{114.39} + 2^{116} + 2^{112} + 2^{114} + 2^{92} + 2^{78} \approx N + 2^{117.40}.$$

When $N\rho^2 = 2^{114.72}$ is used with $a = 4.98$-bit advantage, the success probability is higher than 80%. Therefore, the required data complexity is $N = 2^{114.72} \times 2^{8.30} = 2^{123.02}$ KP. Thus, the total time complexity is

$$\underbrace{2^{123.02}}_{\text{data collection}} + \underbrace{2^{123.02} + 2^{117.40}}_{\text{cost of main analysis}} + \underbrace{2^{128-4.98}}_{\text{exhaustive search}} \approx 2^{124.61}.$$

Step 1 requires 2^{112} table. Thus, the memory complexity is 2^{112}.

F.3 Detail of the 17-Round Linear Attack against GIFT-128 on the COFB Setting

Table 7: 11-round linear trail.

X0	00000000000001100000000000000110	2^{-6}
X1	0000000000000000000C000C00000000	2^{-2}
X2	00000000000000000000000000001100	2^{-3}
X3	0000000000000000000000000000000C	2^{-1}
X4	00000000000000000000000200000000	2^{-1}
X5	00000000000000000000020000000100	2^{-3}
X6	00000000000000000000080800000000	2^{-2}
X7	00000000000005000000000000000500	2^{-4}
X8	00000000000000000000000000040004	2^{-2}
X9	00000000000004400000022000000000	2^{-5}
X10	00000900000C0000000606000003000	2^{-5}
X11	00000000000000000800202000101410	
Total		2^{-34}

Linear Approximation. The existing attack uses the 10-round linear approximation and appends a 3-round key recovery to both plaintext and ciphertext sides. We first search for a suitable 11-round linear trail satisfying the following conditions:

- When we add a 3-round key recovery to the plaintext side, it only involves the last half block. This condition is necessary for the attack on the COFB setting.
- When we add a 3-round key recovery to the ciphertext side, it involves only half the size of the block length.
- The correlation of the linear trail is as high as possible.

As a result, we found a new 11-round linear trail shown in Table 7.

The correlation of this trail is too low to lead to a valid attack with the birthday query limitation. Therefore, we remove the first round and the last two rounds from the linear trail and use the following 8-round linear approximation instead.

$$\texttt{0000000000000000000C000C00000000}$$
$$\rightarrow\texttt{00000000000000044000002200000000}$$

The correlation of the extracted 8-round linear trail is $\pm 2^{-18}$. To estimate the correlation and ELP of this linear approximation, we searched for other linear trails under restricting the input and output linear masks. Then, the second best linear trail has a correlation of $\pm 2^{-55}$, which is significantly less than 2^{-18}. Therefore, we simply estimate the correlation and ELP of the linear approximation is $\pm 2^{-18}$ and 2^{-36}, respectively.

Punctured Key Recovery. We apply the bit puncturing to both plaintext and ciphertext sides. Specifically, the bit puncturing is applied to the GIFT Sbox for the plaintext side. On the other hand, it is applied to the GIFT super Sbox for the ciphertext side. Figure 7 shows the punctured Sboxes and super Sboxes.

Plaintext Side. There are four active Sboxes in the bit puncturing of the plaintext side. Specifically, we want to compute $\langle \mathtt{8}, S \rangle$ and $\langle \mathtt{4}, S \rangle$. Considering the feasibility of adding a 3-round key recovery further, we cannot use the MSB and the 2nd LSB. Therefore, we puncture these two bits.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	1	A	4	C	6	F	3	9	2	D	B	7	5	0	8	E
$f_{\mathtt{8}} = (-1)^{\langle \mathtt{8}, S \rangle}$	1	-1	1	-1	1	-1	1	-1	1	-1	-1	1	1	1	-1	-1
$\hat{f}_{\mathtt{8}}$	0	8	4	4	0	0	-4	4	0	8	-4	-4	0	0	4	-4
$\hat{g}_{\mathtt{8}}$	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$g_{\mathtt{8}}$	0.5	-0.5	0.5	-0.5	0.5	-0.5	0.5	-0.5	0.5	-0.5	0.5	-0.5	0.5	-0.5	0.5	-0.5
$f_4 = (-1)^{\langle \mathtt{4}, S \rangle}$	1	1	-1	-1	-1	-1	1	1	1	-1	1	-1	-1	1	1	-1
$\hat{f}_4$	0	4	0	-4	0	4	8	4	0	-4	0	4	0	-4	8	-4
$\hat{g}_4$	0	4	0	0	0	4	0	0	0	0	0	0	0	0	0	0
g_4	0.5	-0.5	0.5	-0.5	0	0	0	0	0.5	-0.5	0.5	-0.5	0	0	0	0

Table 8: Bit puncturing for plaintext side

Table 8 summarizes the bit puncturing for the plaintext side. Note that $g_{\mathtt{8}}$ is equivalent to just the extension of the linear trail rather than the key recovery. It uses only x_0 to compute y_3 with puncturing correlation 2^{-2}. On the other hand, g_4 uses two non-zero Walsh spectrum coefficients. It rejects half data, and the puncturing correlation is 2^{-3}.

Combined four punctured Sbox, the puncturing correlation is $\rho_1^2 = 2^{-2-2-3-3} = 2^{-10}$.

Ciphertext Side. Figure 7 shows two active super Sboxes, to which we apply the puncturing technique. Let $(x_{15}, \ldots, x_0)$ and $(y_{15}, \ldots, y_0)$ be the input and output of the super Sbox, and $(k_7, \ldots, k_0)$ denote the internal key.

In the one super Sbox, we compute $x_6 \oplus x_2$ from $(y_{15}, \ldots, y_0)$ with 10-bit puncturing. Computing the Walsh spectrum, we get a very simple result. For all involved internal key bits, only $\mathtt{0x0133}, \mathtt{0x0233}$ takes non-zero Walsh spectrum coefficients, and these values (multiplied by 2^{16}) take $(4096, 4096)$ or $(-4096, -4096)$. Thus, the puncturing correlation is 2^{-7}, and we do not need to guess internal key bits.

In another super Sbox, we compute $x_5 \oplus x_1$ with 10-bit puncturing. The punctured Walsh spectrum contains 38 non-zero coefficients; each non-zero value

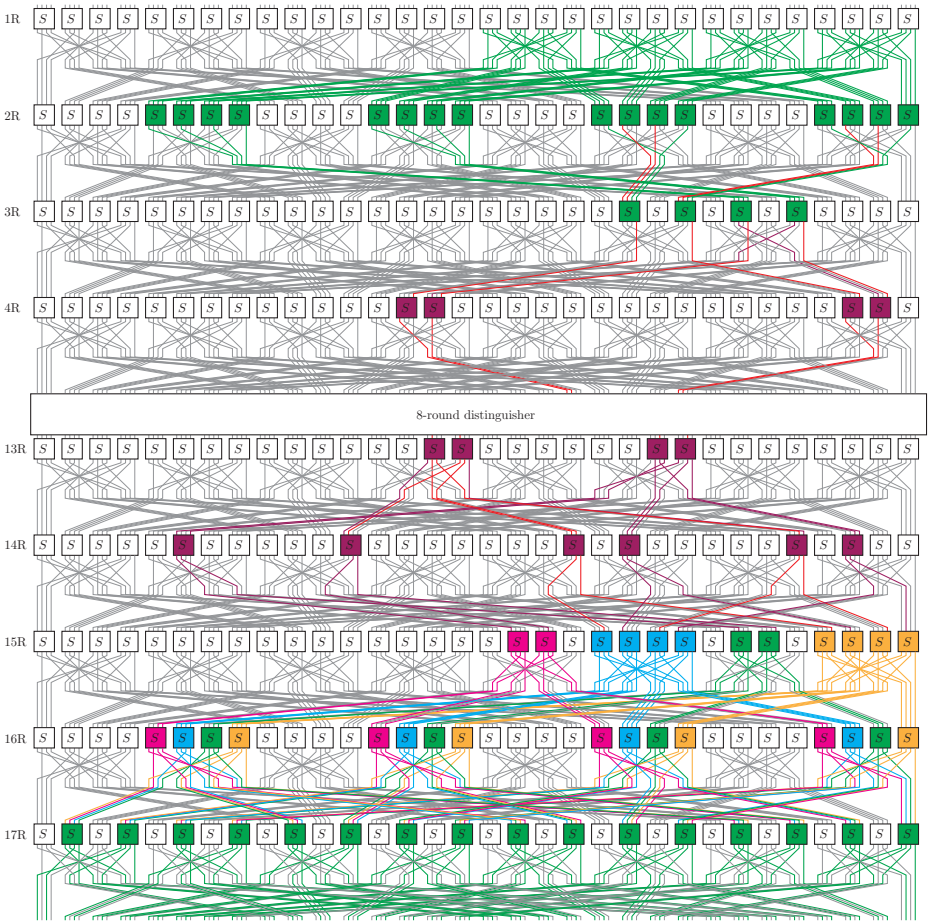


Fig. 13: key recovery map for the improved 17-round attack.

depends on the involved 3-bit internal key. The puncturing correlation is $2^{-6.62}$ for all internal key bits, which is larger than the second example.

Combined two punctured super Sbox, the puncturing correlation $\rho_2^2 = 2^{-7-6.62} = 2^{-13.62}$.

Attack Procedure. Let $x^r = (x^r_{127}, \ldots, x^r_0)$ be the input of the rth round function, where x^r_0 and x^r_{127} denote the LSB and MSB of x^r, respectively. We also use $x'^r = (x'^r_{127}, \ldots, x'^r_0) = P^{-1}(x^r)$. The round number r starts from 0, i.e., x^0 denotes a plaintext and x^{25} denotes a ciphertext. Figure 13 shows the key recovery map, and Fig. 14 summarizes the involved master-key bits.

1. We collect N ($< 2^{64}$) plaintext-ciphertext pairs and store them.
2. We guess involved key bits in the 1st, 2nd, 3rd, and the last rounds. Many key guesses are overlapped, and in total, a 54-bit guess is enough. Then, we can

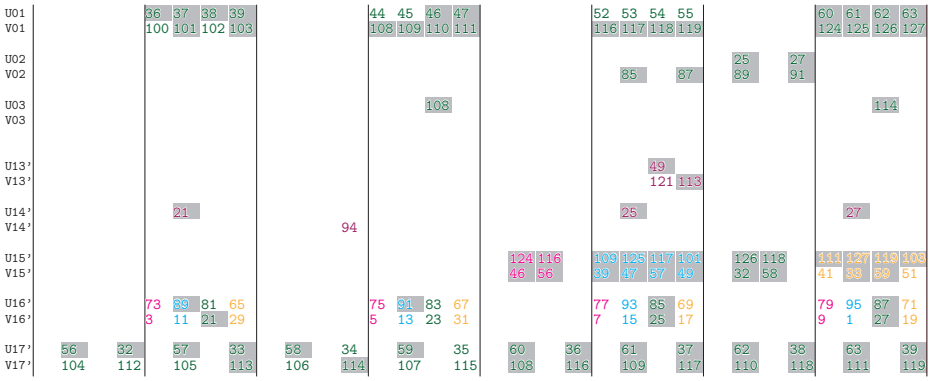


Fig. 14: Summary of involved master-key bits

compute the (punctured) input parity of the 9-round approximation and the output of the 16th-round function. Then, the punctured input parity takes $\pm 2^{-4}$ or 0. So, we normalize it by multiplying 2^4. When we combine three cases and generate a unified distillation table, each entry of the distillation table is incremented, decremented, or not updated depending on the normalized parity. We further guess the 81st, 21st, 83rd, and 23rd key bits and evaluate four and two Sboxes in the 16th and 15th round functions, respectively (Sboxes coloured green). As a result, we construct $16 \times 3 + 2 = 50$-bit table. The time complexity is $N \times 2^{58}$.

3. We guess key bits indexed by $\{11, 13, 93, 15, 95, 1, 49\}$. Then, we can evaluate four Sboxes in the 16th round and four Sboxes in the 15th round (Sboxes coloured blue). We can compress the 50-bit table into a 38-bit table. The time complexity is $2^{58+7} \times 2^{50} = 2^{115}$.
4. We guess key bits indexed by $\{73, 3, 75, 5, 77, 7, 79, 9\}$. Then, we can evaluate four Sboxes in the 16th round and two Sboxes in the 15th round (Sboxes coloured pink). We can compress the 38-bit table into a 24-bit table. The time complexity is $2^{65+8} \times 2^{38} = 2^{111}$.
5. We guess key bits indexed by $\{65, 29, 67, 31, 69, 17, 71, 19, 41, 51\}$. Then, we can evaluate four Sboxes in the 16th round and four Sboxes in the 15th round (Sboxes coloured yellow). We can compress the 24-bit table into a 12-bit table. The time complexity is $2^{73+10} \times 2^{24} = 2^{107}$.
6. We further guess a key bit indexed by 94. Then, we have the 12-bit table, which outputs the punctured super Sboxes. The involved key bits in the punctured super Sbox are 49, 113, and 121. We already guessed 49 and 113. Thus, we additionally guess the key bit indexed by 49 and compute the (punctured) correlation. The time complexity is $2^{83+1+1} \times 2^{12} = 2^{97}$.

The time complexity of the main attack procedure is

$$N \times 2^{58} + 2^{115} + 2^{111} + 2^{107} + 2^{104} \approx N \times 2^{58} + 2^{115.09}.$$

The most critical part of the memory complexity is Step 2, and we need 2^{108} table.

When we use $2^{62.10}$ KP, that is the same data complexity as the existing 16-round attack, with $a = 2.96$-bit advantage, the success probability is higher than 80%. The total time complexity is

$$\underbrace{2^{62.10}}_{\text{data collection}} + \underbrace{2^{62.10+58} + 2^{115.09}}_{\text{cost of main analysis}} + \underbrace{2^{128-2.96}}_{\text{exhaustive search}} \approx 2^{125.09}.$$

The table size in the main analysis is at most 2^{50}, but we must sore N plaintext-ciphertext pairs. Therefore, the memory complexity is N.

G Supplementary Material for the Application to the Data Encryption Standard

G.1 Specification of the Data Encryption Standard

The Data Encryption Standard [1] takes a 64-bit plaintext and a 56-bit key. The cipher is a 16-round Feistel network whose state (L, R) has two 32-bit parts.

$(L_0, R_0) \leftarrow IP(P)$;
for $i \leftarrow 1$ **to** 16 **do**
  $L_i \leftarrow R_{i-1}$;
  $R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, K_i)$;
end
$C \leftarrow IP^{-1}(R_{16}, L_{16})$;

where IP is an initial permutation and the K_i are 48-bit round subkeys.

The round function f. The 32-bit input is expanded to a 48-bit string which is xored with the round subkey, and eight different 6-to-4-bit Sboxes $S_1, \ldots, S_8$ are applied to obtain a 32-bit string, whose bits are reordered again. The expansion function E, the Sboxes and the final permutation P can be found in [1].

The key schedule. The 56-bit key is expanded to sixteen 48-bit subkeys:

$(C_0, D_0) \leftarrow PC_1(K)$;
for $i \leftarrow 1$ **to** 16 **do**
  $C_i \leftarrow LS_{p(i)}(C_{i-1})$;
  $D_i \leftarrow LS_{p(i)}(D_{i-1})$;
  $K_i \leftarrow PC_2(C_i, D_i)$;
end

where C_i and D_i are 28 bits long, PC_1 and PC_2 are two permutated choices, LS_j is a j bit rotation to the left, and $p(i)$ is either 1 or 2.

G.2 Calculation of the Time and Memory Complexities

Let us compute the overall time and memory complexities of the attack. Since the attack algorithm is nearly identical to that of [30], we will just compute the new complexity of each step, and refer the reader to the original attack for further details. For each of the nine components that the punctured Walsh spectrum is split into, we must compute the dimensions of the input and output spaces for both the Walsh transform steps, as well as the dimension $t = \dim\left(U/(U \cap V^{\perp})\right) = \dim\left(V/(V \cap U^{\perp})\right)$, where U is the input space and V is the output space. This is the dimension which determines the actual time complexity of the pruned Walsh transform as per Proposition 7 of [30]. The results are shown in Table 9. The total number of necessary additions for the first set of Walsh transforms is $2^{39.26}$, and for the second set it is $2^{39.59}$.

The cost of each step of the attack is thus:

Table 9: The effective dimensions of the pruned Walsh transforms.

	First FWT			Second FWT		
S_5 Mask	Input	Output	Inner	Input	Output	Inner
0E	40	27	27	27	40	25
17	40	34	32	34	40	28
1D	40	34	30	34	40	28
27	40	34	32	34	40	28
2B	40	34	30	34	40	31
2D	40	34	32	34	40	28
35	40	34	32	34	40	28
3A	40	34	28	34	40	34
3C	40	34	30	34	40	32
Additions			$2^{39.26}$		$2^{39.59}$	

- The cost of the distillation phase is $9 \cdot N$ increments and decrements.
- The cost of the first set of Walsh transforms is $2^{39.26}$ additions.
- The cost of the Walsh spectrum multiplication step is bound by the total number of nonzero coefficients in the Walsh spectrum. This means this step can be carried out with $2^{25.36}$ products, most of which are bit shifts.
- The second set of Walsh transforms requires $2^{39.59}$ additions.
- Combining the information from the nine components takes $9 \cdot 2^{40}$ additions.
- The cost of the final search with an 8-bit advantage is 2^{40} trial encryptions.

Using the same cost comparisons of each of the operations in the attack to a full DES encryption as in [30], we deduce that the full time complexity of the attack in equivalent encryptions is is

$$\frac{1}{16} \cdot 9 \cdot 2^{41.62} + \frac{1}{16} \left(2^{39.26} + 2^{39.59} + 9 \cdot 2^{40}\right) + \frac{6}{16} \cdot 2^{25.36} + 2^{40} \simeq 2^{41.76}. \quad (13)$$

The required memory registers which are used in the attack are $2^{34.27}$ for the first set of pruned Walsh transforms and $2^{34.54}$ for the second. Since it is possible to perform the multiplication step in such a way that both sets of arrays don't need to be stored in full simultaneously, the total memory complexity of the attack is $2^{34.54}$.

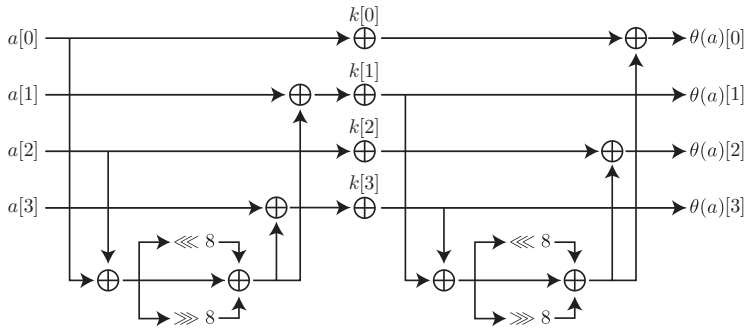
H Supplementary Material for the Application to Noekeon

H.1 Specification of Noekeon

Noekeon [27] is the NESSIE-proposal 128-bit block cipher accepting a 128-bit secret key. The designers recommend 16 rounds.

The internal state a consists of four 32-bit words, $a = (a[0], a[1], a[2], a[3])$. The round subkey is the same for all the rounds. Each round function consists of the following transformations:

1. A constant is XORed to $a[0]$.
2. A keyed linear transformation θ is applied to the state. Note that the transformation θ is involution if no key is added.



3. A shift operation π_1 is applied to the state.

$$\pi_1(a)[0] = a[0], \qquad \pi_1(a)[1] = a[1] \lll 1$$
$$\pi_1(a)[2] = a[2] \lll 5, \qquad \pi_1(a)[3] = a[3] \lll 2,$$

4. A non-linear function γ consisting of the parallel application of 4-bit Sbox is applied. Note that the γ is involution.

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
$S(x)$	0x7	0xA	0x2	0xC	0x4	0x8	0xF	0x0	0x5	0x9	0x1	0xE	0x3	0xD	0xB	0x6

5. Another shift operation π_2, which is the inverse of π_1 is applied to the state.

for convenience, we will denote $\hat{\theta} = \pi_1 \circ \theta \circ \pi_2$, which is also involution.

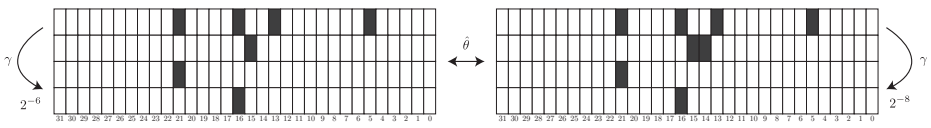
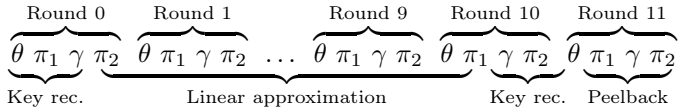


Fig. 15: Iterative linear trail of NOEKEON

H.2 Review of the Linear Attack Proposed in Asiacrypt 2021 [19]

The linear attack against the 12-round NOEKEON was shown in [19]. We first review the attack and then apply the punctured key recovery.

The authors of [19] showed the 2-round iterative linear trail with a correlation of 2^{-14}, and Fig. 15 shows the iterative trail. The trail is extended to the 9-round one to mount the 12-round attack with the following key recovery structure:



They focused on the 15th Sbox in Round 1 and applied their technique; the 3-bit input of the 15th Sbox is enough to compute the output parity with a probability of 1/2. Therefore, when N KPs are used, $N/2$ plaintext-ciphertext pairs are available. They also changed the linear approximation for the 15th Sbox in Round 9 to increase the correlation by the factor of 2^{-1}. As a result, the correlation increases from 2^{-62} to 2^{-59}. The size of the key guess is 124 bits. The time complexity for the FWT is $2^{124.29}$ `ADD` by using the technique in [31].

There is no discussion about the success probability and advantage in [19]. Therefore, we use our formula. When $N = 2^{122.35}$ KP is used, $2^{121.35}$ plaintext-ciphertext pairs are available, with a 5.65-bit advantage, the success probability is higher than 80%. The final time complexity is $2^{122.35}+0.2\times 2^{124.29}+2^{128-5.65} \approx 2^{123.82}$, where we inherit the same constant factor 0.2 from [19].

H.3 Improved Key Recovery Using Puncturing

The existing attack requires data that is equivalent to the time complexity of the FWT. Therefore, the straightforward puncturing, reducing the time complexity for the FWT in return for the data increase, cannot improve the attack. We need to switch some rounds of the linear trail into the key recovery.

We remove the first and last rounds from the original 9-round linear trail and get the 7-round trail whose correlation is 2^{-50}. In the key recovery, we add two rounds to the plaintext side and three rounds to the ciphertext side. Note that, similarly to [19], the key recovery rounds of the ciphertext side are regarded as two rounds by using the peelback. Therefore, our attack has the following key

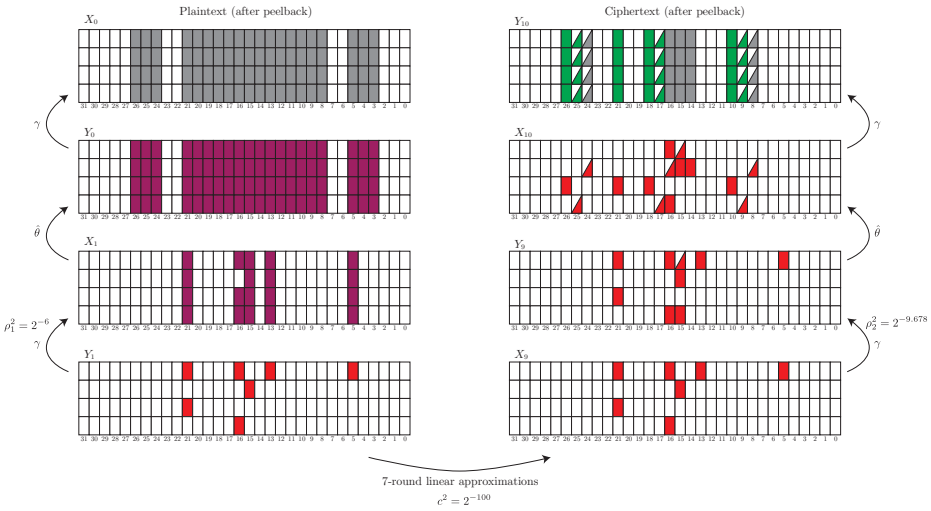


Fig. 16: Key recovery of 12-round NOEKEON with puncturing

recovery structure:

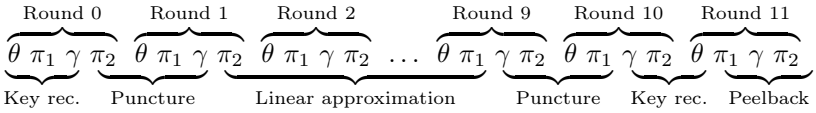


Figure 16 shows the (punctured) key recovery.

Two-round decryption is required on the ciphertext side. We use the punctured key recovery in Round 9. Specifically, we remove Walsh spectrum coefficients except for two coefficients, where active Sbox in Y_9 has either linear mask $(\texttt{0x5}, \texttt{0x9}, \texttt{0xA}, \texttt{0x1}, \texttt{0x1})$ or $(\texttt{0x5}, \texttt{0x9}, \texttt{0xB}, \texttt{0x1}, \texttt{0x1})$. We call the first coefficient type A and the second coefficient type B for simplicity. The type A has a correlation of 2^{-6}. It causes seven active Sboxes in Round 10. There are four active Sboxes whose dimension of the affine subspace is only two. Therefore, the dimension of the affine subspace is $2\times4+4\times3 = 20$. The type B has a correlation of 2^{-5}. It causes 12 active Sboxes in Round 10. There are seven active Sboxes whose dimension of the affine subspace is only two. Therefore, the dimension of the affine subspace is $2\times7+4\times5 = 34$. Note that the guessed bits in the type B include the guessed bits in the type A. Therefore, when we consider affine subspace merging these two types, the dimension is 35, and $\rho_2^2 = (2^{-10} + 2^{-12})^{-1} \approx 2^{-9.68}$.

We more widely apply the same idea to the plaintext side. There are 10^5 non-zero Walsh coefficients in X_1, and we puncture Walsh coefficients if it involves either of the 0th, 1st, 2nd, 6th, 7th, 22nd, 23rd, 27th, 28th, 29th, 30th, and 31st Sboxes in Round 0. As a result, 460 out of 10^5 non-zero Walsh coefficients remain. Then, the dimension of the involved plaintext is $20 \times 4 = 80$ after puncturing, and $\rho_1^2 = 2^{-6}$.

We must guess the 19-bit and 1-bit internal keys in Round 1 and 9, respectively. Therefore, the naive FWT-based key recovery requires higher complexity than 2^{128}. We use the Walsh spectrum decomposition [30] to reduce the complexity. Namely, we decompose the key recovery map into $2 \times 460 = 920$ maps, apply the FWT independently, and combine 920 results in the final guess. Note that we no longer need to guess the internal key bits in each decomposition because the impact is just a sign of the empirical correlation. More accurately, we take the impact into consideration in the final key guess step.

Figures 17, 18, 19, and 20 summarizes our decomposition. `mask of X1` denotes Walsh spectrum coefficients in X_1, and `mask of Y0` denotes corresponding coefficients in Y_0. `d1` denotes the dimension of the affine subspace in X_0. `comp.1` denotes the complexity for the 1st FWT, which is

$$(\texttt{d1} + 20) \times 2^{\texttt{d1}+20} + (\texttt{d1} + 34) \times 2^{\texttt{d1}+34}.$$

Sometimes, we have exactly the same support in different decompositions, e.g., the decomposition with `id` 4 and `id` 5 share the same support. Then, we do not need to apply the 1st FWT in `id` 5. By using this trick, we can reduce the number of decompositions for the 1st FWT from 460×2 to 331×2. `# mul` denotes the number of non-zero coefficients. It is also the number of required multiplications. We notice that each decomposed Walsh spectrum is very sparse. Therefore, the multiplication cost is negligible compared with other parts. `d2` and `d3` denote the dimension of involved key bits when the type A and type B are used on the ciphertext side, respectively. Note that $\texttt{d2} \leq \texttt{d1} + 20$ and $\texttt{d3} \leq \texttt{d1} + 34$ hold because the 10th subkey can be linearly computed from the 0th subkey. Finally, `comp.2` denotes the complexity for the 2nd FWT, which is

$$2^{\texttt{d1}+20} + (\texttt{d2} \times 2^{\texttt{d2}}) + 2^{\texttt{d1}+34} + (\texttt{d3} \times 2^{\texttt{d3}}).$$

Distillation Phase. We first store the information of N plaintext-ciphertext pairs into a distillation table with a size of $2^{80+35} = 2^{115}$. Then, we construct two distillation tables, T_A and T_B, where T_A is used for the type A and T_B is used for the type B. The size of T_A is $2^{80+20} = 2^{100}$, and the size of T_B is $2^{80+34} = 2^{114}$. From T_A, we construct 331 distillation tables for each decomposition. We further construct four distillation tables, $T_{B,1}$, $T_{B,2}$, $T_{B,3}$, and $T_{B,4}$, from the table T_B accordingly to the Walsh coefficieient for the 26th Sboxes in Round 0. We notice that there is only four cases, `0x0`, `0x4`, `0x8`, and `0xC`. As a result, the sizes of $T_{B,1}$, $T_{B,2}$, $T_{B,3}$, and $T_{B,4}$ are 2^{106}, 2^{110}, 2^{110}, and 2^{110}, respectively. From $T_{B,1}$, $T_{B,2}$, $T_{B,3}$, and $T_{B,4}$, we construct, 63, 109, 56, and 103 distillation tables for each decomposition, respectively. The complexity of the distillation phase is summarized as follows.

$$\begin{aligned} &N + (2^{115} + 4 \times 2^{114} + 331 \times 2^{100} + 63 \times 2^{106} + 109 \times 2^{110} + 56 \times 2^{110} + 103 \times 2^{110}) \\ &\approx N + 2^{118.52} \end{aligned}$$

First FWT. We apply the FWT to 662 decompositions. The complexity is the sum of `comp.1` and about $2^{116.38}$.

Componenet-Wise Multiplications. As already explained, the Walsh spectrum is very sparse. We regard the cost for the multiplications as negligible.

Second FWT. After multiplications, we have 920 tables, where each table size is $2^{\mathtt{d1}+20}$ or $2^{\mathtt{d1}+34}$. The dimension of the corresponding master key subspace is lower because the 10th subkey can be linearly computed from the 0th subkey. The complexity is the sum of `comp.2` and about $2^{112.33}$.

Final Result. As a result, when $N\rho^2 = 2^{103.86}$ with a 8.45-bit advantage, the probability is higher than 80%. Therefore, the required data complexity is $N = 2^{103.86} \times 2^6 \times 2^{9.68} = 2^{119.54}$ The final time complexity is

$$2^{119.55} + 0.2 \times (2^{118.52} + 2^{116.38} + 2^{112.33}) + 2^{128-8.45} \approx 2^{120.63},$$

where we inherit the same constant factor 0.2 as the cost of `ADD` from [19].

id	mask of X1	mask of Y0	id'	d1	comp.1	# mul	d2	d3	comp.2
1	5 4 C 1 1	- - - - - C 2 - - - C A 2 C 2 4 4 2 8 A 2 C 2 - - - 8 A 2 - - -	1	56	96.49	48.91	72	83	96.52
2	5 4 C 3 1	- - - - - C 6 - - - C B 2 C 6 4 4 2 A B 2 C 6 - - - 8 B 2 - - -	2	58	98.52	46.92	74	85	98.55
3	5 4 C B 1	- - - - - C 6 4 - - C B 3 C 6 - 4 2 2 B 3 C 6 4 - - 8 B 3 - - -	3	60	100.55	48.57	76	87	100.58
4	5 4 1 1 1	- - - - - 8 A 2 - - D 2 - 8 A 6 9 2 9 2 - 8 A 2 - - 9 2 - - - -	4	62	102.59	40.55	74	84	102.60
5	5 4 1 3 1	- - - - - 8 E 2 - - D 3 - 8 E 6 9 2 B 3 - 8 E 2 - - 9 3 - - - -	4	62		40.55	74	84	102.60
6	5 4 1 B 1	- - - - - 8 E 6 - - D 3 1 8 E 2 9 2 3 3 1 8 E 6 - - 9 3 1 - - -	6	74	114.75	40.83	85	94	114.77
7	5 4 5 1 1	- - - - - 8 A 2 - - D A 2 8 A 6 D 2 9 A 2 8 A 2 - - 9 A 2 - - -	6	74		40.83	85	94	114.77
8	5 4 5 3 1	- - - - - 8 E 2 - - D B 2 8 E 6 D 2 B B 2 8 E 2 - - 9 B 2 - - -	6	74		40.83	85	94	114.77
9	5 4 5 B 1	- - - - - 8 E 6 - - D B 3 8 E 2 D 2 3 B 3 8 E 6 - - 9 B 3 - - -	6	74		40.83	85	94	114.77
10	5 4 9 1 1	- - - - - C A 2 - - C 2 - C A 6 1 2 8 2 - C A 2 - - 8 2 - - - -	10	56	96.49	45.48	69	79	96.51
11	5 4 9 3 1	- - - - - C E 2 - - C 3 - C E 6 1 2 A 3 - C E 2 - - 8 3 - - - -	11	58	98.52	43.55	71	80	98.54
12	5 4 9 B 1	- - - - - C E 6 - - C 3 1 C E 2 1 2 2 3 1 C E 6 - - 8 3 1 - - -	12	70	110.70	43.69	82	90	110.71
13	5 4 D 1 1	- - - - - C A 2 - - C A 2 C A 6 5 2 8 A 2 C A 2 - - 8 A 2 - - -	13	68	108.67	45.61	80	89	108.69
14	5 4 D 3 1	- - - - - C E 2 - - C B 2 C E 6 5 2 A B 2 C E 2 - - 8 B 2 - - -	12	70		43.69	82	90	110.71
15	5 4 D B 1	- - - - - C E 6 - - C B 3 C E 2 5 2 2 B 3 C E 6 - - 8 B 3 - - -	12	70		43.69	82	90	110.71
16	5 9 B 1 1	- - - - - - 4 8 2 - - 4 - - 4 8 B 3 2 - - - 4 8 2 - - - - - - -	16	34	74.09	45.34	50	63	74.15
17	5 9 B 3 1	- - - - - - 4 C 2 - - 4 1 - 4 C B 3 2 2 1 - 4 C 2 - - - 1 - - -	17	50	90.39	45.48	64	75	90.41
18	5 9 B B 1	- - - - - - 4 C 6 - - 4 1 1 4 C F 3 2 A 1 1 4 C 6 - - - 1 1 - - -	18	62	102.59	45.61	75	85	102.60
19	5 9 2 1 1	- - - - - - - - - - 5 - - - - 9 A 2 1 - - - - - - - - 1 - - - -	19	24	63.86	36.54	40	54	63.96
20	5 9 2 3 1	- - - - - - - 4 - - - 5 1 - - 4 9 A 2 3 1 - - 4 - - - 1 1 - - -	20	42	82.25	39.85	56	68	82.27
21	5 9 2 B 1	- - - - - - - 4 4 - - 5 1 1 - 4 D A 2 B 1 1 - 4 4 - - 1 1 1 - - -	21	58	98.52	42.55	71	81	98.54
22	5 9 6 1 1	- - - - - - - - - - - 5 8 2 - - 9 E 2 1 8 2 - - - - - 1 8 2 - - -	22	42	82.25	41.25	57	71	82.31
23	5 9 6 3 1	- - - - - - - 4 - - - 5 9 2 - 4 9 E 2 3 9 2 - 4 - - - 1 9 2 - - -	23	54	94.46	40.28	67	79	94.48
24	5 9 6 B 1	- - - - - - - 4 4 - - 5 9 3 - 4 D E 2 B 9 3 - 4 4 - - 1 9 3 - - -	21	58		42.55	71	81	98.54
25	5 9 A 1 1	- - - - - - 4 - - - - 4 - - 4 - 9 2 2 - - - 4 - - - - - - - - -	25	20	59.75	39.16	36	50	59.86
26	5 9 A 3 1	- - - - - - 4 4 - - - 4 1 - 4 4 9 2 2 2 1 - 4 4 - - - - 1 - - -	26	42	82.25	44.88	57	69	82.28
27	5 9 A B 1	- - - - - - 4 4 4 - - 4 1 1 4 4 D 2 2 A 1 1 4 4 4 - - - 1 1 - - -	27	58	98.52	48.74	72	82	98.54
28	5 9 E 1 1	- - - - - - 4 - - - - 4 8 2 4 - 9 6 2 - 8 2 4 - - - - - 8 2 - - -	28	38	78.17	44.97	54	68	78.27
29	5 9 E 3 1	- - - - - - 4 4 - - - 4 9 2 4 4 9 6 2 2 9 2 4 4 - - - - 9 2 - - -	29	54	94.46	45.07	68	80	94.48
30	5 9 E B 1	- - - - - - 4 4 4 - - 4 9 3 4 4 D 6 2 A 9 3 4 4 4 - - - 9 3 - - -	27	58		48.74	72	82	98.54
31	8 4 C 8 8	- - - - - - 4 - - - - 1 A 2 4 - - C - 1 A 3 4 - - - - - 1 A 2 - - -	31	44	84.29	41.79	61	74	84.39
32	8 4 C 2 8	- - - - - - 4 4 4 - - 1 B 3 4 4 4 C - B B 2 4 4 4 - - 1 B 3 - - -	32	56	96.49	51.17	72	83	96.52
33	8 4 C A 8	- - - - - - 4 4 - - - 1 B 2 4 4 - C - 3 B 3 4 4 - - - 1 B 2 - - -	33	50	90.39	46.15	67	79	90.45
34	8 4 1 8 8	- - - - - - - 8 2 - - - 2 - - 8 2 1 - - 2 1 - 8 2 - - - 2 - - - -	34	38	78.17	39.75	57	69	78.35
35	8 4 1 2 8	- - - - - - - C 6 - - - 3 1 - C 6 1 - A 3 - - C 6 - - - 3 1 - - -	35	46	86.32	40.13	64	75	86.38
36	8 4 1 A 8	- - - - - - - C 2 - - - 3 - - C 2 1 - 2 3 1 - C 2 - - - 3 - - - -	36	42	82.25	39.85	60	71	82.31
37	8 4 5 8 8	- - - - - - - 8 2 - - - A 2 - 8 2 5 - - A 3 - 8 2 - - - A 2 - - -	37	46	86.32	40.13	64	76	86.42
38	8 4 5 2 8	- - - - - - - C 6 - - - B 3 - C 6 5 - A B 2 - C 6 - - - B 3 - - -	38	50	90.39	40.21	67	78	90.43
39	8 4 5 A 8	- - - - - - - C 2 - - - B 2 - C 2 5 - 2 B 3 - C 2 - - - B 2 - - -	38	50		40.21	67	78	90.43
40	8 4 9 8 8	- - - - - - 4 8 2 - - 1 2 - 4 8 2 9 - 1 2 1 4 8 2 - - 1 2 - - - -	40	56	96.49	44.23	71	81	96.51
41	8 4 9 2 8	- - - - - - 4 C 6 - - 1 3 1 4 C 6 9 - B 3 - 4 C 6 - - 1 3 1 - - -	41	60	100.55	44.48	74	85	100.57
42	8 4 9 A 8	- - - - - - 4 C 2 - - 1 3 - 4 C 2 9 - 3 3 1 4 C 2 - - 1 3 - - - -	42	56	96.49	44.23	70	81	96.51
43	8 4 D 8 8	- - - - - - 4 8 2 - - 1 A 2 4 8 2 D - 1 A 3 4 8 2 - - 1 A 2 - - -	43	64	104.61	44.50	78	88	104.63
44	8 4 D 2 8	- - - - - - 4 C 6 - - 1 B 3 4 C 6 D - B B 2 4 C 6 - - 1 B 3 - - -	44	64	104.61	44.50	77	88	104.63
45	8 4 D A 8	- - - - - - 4 C 2 - - 1 B 2 4 C 2 D - 3 B 3 4 C 2 - - 1 B 2 - - -	44	64		44.50	77	88	104.63
46	8 9 B 8 8	- - - - - - C A 2 - - 9 - - C A F B - 9 - 1 C A 2 - - 9 - - - - -	46	50	90.39	40.21	64	76	90.41
47	8 9 B 2 8	- - - - - - C E 6 - - 9 1 1 C E B B - 3 1 - C E 6 - - 9 1 1 - - -	47	66	106.64	40.73	79	88	106.66
48	8 9 B A 8	- - - - - - C E 2 - - 9 1 - C E F B - B 1 1 C E 2 - - 9 1 - - - -	48	62	102.59	40.55	75	84	102.60
49	8 9 2 8 8	- - - - - - 8 2 - - - 8 - - 8 2 D 2 - 8 - 1 8 2 - - - 8 - - - -	49	36	76.13	44.62	53	66	76.23
50	8 9 2 2 8	- - - - - - 8 6 4 - - 8 1 1 8 6 9 2 - 2 1 - 8 6 4 - - 8 1 1 - - -	50	58	98.52	46.62	75	85	98.55
51	8 9 2 A 8	- - - - - - 8 6 - - - 8 1 - 8 6 D 2 - A 1 1 8 6 - - - 8 1 - - -	51	50	90.39	42.96	67	78	90.43
52	8 9 6 8 8	- - - - - - 8 2 - - - 8 8 2 8 2 D 6 - 8 8 3 8 2 - - - 8 8 2 - - -	52	50	90.39	50.53	67	79	90.45
53	8 9 6 2 8	- - - - - - 8 6 4 - - 8 9 3 8 6 9 6 - 2 9 2 8 6 4 - - 8 9 3 - - -	53	62	102.59	46.63	78	88	102.61
54	8 9 6 A 8	- - - - - - 8 6 - - - 8 9 2 8 6 D 6 - A 9 3 8 6 - - - 8 9 2 - - -	54	58	98.52	43.13	74	85	98.55
55	8 9 A 8 8	- - - - - - C 2 - - - 9 - - C 2 D A - 9 - 1 C 2 - - - 9 - - - - -	55	42	82.25	39.85	57	70	82.29
56	8 9 A 2 8	- - - - - - C 6 4 - - 9 1 1 C 6 9 A - 3 1 - C 6 4 - - 9 1 1 - - -	56	62	102.59	43.25	76	86	102.60
57	8 9 A A 8	- - - - - - C 6 - - - 9 1 - C 6 D A - B 1 1 C 6 - - - 9 1 - - - -	57	54	94.46	40.28	68	79	94.48
58	8 9 E 8 8	- - - - - - C 2 - - - 9 8 2 C 2 D E - 9 8 3 C 2 - - - 9 8 2 - - -	58	56	96.49	45.48	70	82	96.51
59	8 9 E 2 8	- - - - - - C 6 4 - - 9 9 3 C 6 9 E - 3 9 2 C 6 4 - - - 9 9 3 - - -	59	66	106.64	43.28	79	89	106.66
60	8 9 E A 8	- - - - - - C 6 - - - 9 9 2 C 6 D E - B 9 3 C 6 - - - 9 9 2 - - -	60	62	102.59	40.55	75	86	102.60
61	2 4 C 8 2	- - - - - - 4 4 4 - - B B 3 4 - - C - 1 A 3 4 4 4 - - B B 3 - - -	61	52	92.43	47.19	69	81	92.48
62	2 4 C 2 2	- - - - - - 4 - - - - B A 2 4 4 4 C - B B 2 4 - - - - - B A 2 - - -	62	48	88.36	45.70	64	77	88.42
63	2 4 C 6 6	- - - - - - C 2 - - - B A 2 C 6 4 C - F B 2 4 - - - - F A 2 - - -	63	54	94.46	44.17	70	83	94.52
64	2 4 C A 2	- - - - - - 4 - 4 - - B A 3 4 4 - C - 3 B 3 4 - 4 - - B A 3 - - -	64	50	90.39	46.15	67	79	90.45
65	2 4 C E 6	- - - - - - C 2 4 - - B A 3 C 6 - C - 7 B 3 4 - 4 - - F A 3 - - -	65	56	96.49	45.11	73	85	96.55
66	2 4 1 8 2	- - - - - - - C 6 - - A 3 1 - 8 2 1 - - 2 1 - C 6 - - A 3 1 - - -	66	54	94.46	40.43	71	83	94.52
67	2 4 1 2 2	- - - - - - - 8 2 - - A 2 - - C 6 1 - A 3 - - 8 2 - - - A 2 - - -	67	46	86.32	40.13	62	73	86.35
68	2 4 1 6 6	- - - - - - 8 A 2 - - A 2 - 8 E 6 1 - E 3 - - 8 2 - - - E 2 - - -	68	54	94.46	40.48	70	80	94.48
69	2 4 1 A 2	- - - - - - - 8 6 - - A 2 1 - C 2 1 - 2 3 1 - 8 6 - - - A 2 1 - - -	69	58	98.52	40.49	73	84	98.54
70	2 4 1 E 6	- - - - - - 8 A 6 - - A 2 1 8 E 2 1 - 6 3 1 - 8 6 - - - E 2 1 - - -	70	66	106.64	40.77	81	91	106.66
71	2 4 5 8 2	- - - - - - - C 6 - - A B 3 - 8 2 5 - - A 3 - C 6 - - A B 3 - - -	66	54		40.43	71	83	94.52
72	2 4 5 2 2	- - - - - - - 8 2 - - A A 2 - C 6 5 - A B 2 - 8 2 - - - A A 2 - - -	69	58		40.49	73	84	98.54
73	2 4 5 6 6	- - - - - - 8 A 2 - - A A 2 8 E 6 5 - E B 2 - 8 2 - - E A 2 - - -	70	66		40.77	81	91	106.66
74	2 4 5 A 2	- - - - - - - 8 6 - - A A 3 - C 2 5 - 2 B 3 - 8 6 - - A A 3 - - -	69	58		40.49	73	84	98.54
75	2 4 5 E 6	- - - - - - 8 A 6 - - A A 3 8 E 2 5 - 6 B 3 - 8 6 - - E A 3 - - -	70	66		40.77	81	91	106.66
76	2 4 9 8 2	- - - - - - 4 C 6 - - B 3 1 4 8 2 9 - 1 2 1 4 C 6 - - B 3 1 - - -	76	64	104.61	44.50	78	89	104.63
77	2 4 9 2 2	- - - - - - 4 8 2 - - B 2 - 4 C 6 9 - B 3 - 4 8 2 - - B 2 - - - -	77	52	92.43	44.21	67	78	92.45
78	2 4 9 6 6	- - - - - - C A 2 - - B 2 - C E 6 9 - F 3 - 4 8 2 - - F 2 - - - -	78	56	96.49	42.44	71	80	96.51
79	2 4 9 A 2	- - - - - - 4 8 6 - - B 2 1 4 C 2 9 - 3 3 1 4 8 6 - - B 2 1 - - -	79	64	104.61	44.50	78	89	104.63
80	2 4 9 E 6	- - - - - - C A 6 - - B 2 1 C E 2 9 - 7 3 1 4 8 6 - - B 2 1 - - -	80	68	108.67	42.73	82	91	108.69
81	2 4 D 8 2	- - - - - - 4 C 6 - - B B 3 4 8 2 D - 1 A 3 4 C 6 - - B B 3 - - -	76	64		44.50	78	89	104.63
82	2 4 D 2 2	- - - - - - 4 8 2 - - B A 2 4 C 6 D - B B 2 4 8 2 - - B A 2 - - -	79	64		44.50	78	89	104.63
83	2 4 D 6 6	- - - - - - C A 2 - - B A 2 C E 6 D - F B 2 4 8 2 - - F A 2 - - -	80	68		42.73	82	91	108.69
84	2 4 D A 2	- - - - - - 4 8 6 - - B A 3 4 C 2 D - 3 B 3 4 8 6 - - B A 3 - - -	79	64		44.50	78	89	104.63
85	2 4 D E 6	- - - - - - C A 6 - - B A 3 C E 2 D - 7 B 3 4 8 6 - - F A 3 - - -	80	68		42.73	82	91	108.69
86	2 9 B 8 2	- - - - - - C E 6 - - 3 1 1 C A F B - 9 - 1 C E 6 - - 3 1 1 - - -	86	66	106.64	40.73	80	90	106.66
87	2 9 B 2 2	- - - - - - C A 2 - - 3 - - C E B B - 3 1 - C A 2 - - 3 - - - -	87	50	90.39	40.21	64	75	90.41
88	2 9 B 6 6	- - - - - - 4 8 2 - - 3 - - 4 C B B - 7 1 - C A 2 - - 7 - - - -	88	46	86.32	41.89	62	75	86.38
89	2 9 B A 2	- - - - - - C A 6 - - 3 - 1 C E F B - B 1 1 C A 6 - - 3 - 1 - - -	89	62	102.59	40.55	76	87	102.60
90	2 9 B E 6	- - - - - - 4 8 6 - - 3 - 1 4 C F B - F 1 1 C A 6 - - 7 - 1 - - -	90	58	98.52	42.25	74	87	98.58
91	2 9 2 8 2	- - - - - - 8 6 4 - - 2 1 1 8 2 D 2 - 8 - 1 8 6 4 - - 2 1 1 - - -	91	60	100.55	45.17	77	88	100.59
92	2 9 2 2 2	- - - - - - 8 2 - - - 2 - - 8 6 9 2 - 2 1 - 8 2 - - - 2 - - - -	92	42	82.25	39.85	58	71	82.31
93	2 9 2 6 6	- - - - - - - - - - 2 - - - 4 9 2 - 6 1 - 8 2 - - - 6 - - - -	93	32	72.04	39.09	49	63	72.23
94	2 9 2 A 2	- - - - - - 8 2 4 - - 2 - 1 8 6 D 2 - A 1 1 8 2 4 - - 2 - 1 - - -	94	58	98.52	43.13	74	86	98.56
95	2 9 2 E 6	- - - - - - - - 4 - - 2 - 1 - 4 D 2 - E 1 1 8 2 4 - - 6 - 1 - - -	95	48	88.36	41.84	65	78	88.46
96	2 9 6 8 2	- - - - - - 8 6 4 - - 2 9 3 8 2 D 6 - 8 8 3 8 6 4 - - 2 9 3 - - -	96	62	102.59	47.16	79	90	102.62
97	2 9 6 2 2	- - - - - - 8 2 - - - 2 8 2 8 6 9 6 - 2 9 2 8 2 - - - 2 8 2 - - -	97	58	98.52	43.55	73	85	98.55
98	2 9 6 6 6	- - - - - - - - - - - 2 8 2 - 4 9 6 - 6 9 2 8 2 - - - 6 8 2 - - -	98	48	88.36	42.70	64	78	88.46
99	2 9 6 A 2	- - - - - - 8 2 4 - - 2 8 3 8 6 D 6 - A 9 3 8 2 4 - - 2 8 3 - - -	99	62	102.59	46.79	77	88	102.61
100	2 9 6 E 6	- - - - - - - 4 - - 2 8 3 - 4 D 6 - E 9 3 8 2 4 - - 6 8 3 - - -	100	52	92.43	44.92	68	81	92.48
101	2 9 A 8 2	- - - - - - C 6 4 - - 3 1 1 C 2 D A - 9 - 1 C 6 4 - - 3 1 1 - - -	101	62	102.59	43.25	77	88	102.61
102	2 9 A 2 2	- - - - - - C 2 - - - 3 - - C 6 9 A - 3 1 - C 2 - - - 3 - - - -	102	42	82.25	39.85	57	70	82.29
103	2 9 A 6 6	- - - - - - 4 - - - - 3 - - 4 4 9 A - 7 1 - C 2 - - - 7 - - - -	103	36	76.13	40.68	52	66	76.23
104	2 9 A A 2	- - - - - - C 2 4 - - 3 - 1 C 6 D A - B 1 1 C 2 4 - - 3 - 1 - - -	104	58	98.52	43.13	73	85	98.55
105	2 9 A E 6	- - - - - - 4 - 4 - - 3 - 1 4 4 D A - F 1 1 C 2 4 - - 7 - 1 - - -	105	52	92.43	43.91	68	81	92.48
106	2 9 E 8 2	- - - - - - C 6 4 - - 3 9 3 C 2 D E - 9 8 3 C 6 4 - - 3 9 3 - - -	106	64	104.61	45.21	78	89	104.63
107	2 9 E 2 2	- - - - - - C 2 - - - 3 8 2 C 6 9 E - 3 9 2 C 2 - - - 3 8 2 - - -	107	58	98.52	43.55	72	84	98.54
108	2 9 E 6 6	- - - - - - 4 - - - - 3 8 2 4 4 9 E - 7 9 2 C 2 - - - 7 8 2 - - -	108	52	92.43	44.73	67	81	92.48
109	2 9 E A 2	- - - - - - C 2 4 - - 3 8 3 C 6 D E - B 9 3 C 2 4 - - 3 8 3 - - -	109	62	102.59	46.79	76	87	102.60
110	2 9 E E 6	- - - - - - 4 - 4 - - 3 8 3 4 4 D E - F 9 3 C 2 4 - - 7 8 3 - - -	110	56	96.49	47.48	71	84	96.53
111	3 4 C 1 3	- - - - - - 4 4 - - - A B 2 4 - 4 4 2 8 A 2 4 4 - - - A B 2 - - -	111	52	92.43	48.14	70	83	92.61
112	3 4 C 3 3	- - - - - - 4 - - - - A A 2 4 4 4 4 4 2 A B 2 4 - - - - A A 2 - - -	112	52	92.43	45.71	68	81	92.48
113	3 4 C 7 7	- - - - - - C 2 - - - A A 2 C 6 4 4 2 E B 2 4 - - - - E A 2 - - -	113	58	98.52	44.17	74	87	98.58
114	3 4 C B 3	- - - - - - 4 - 4 - - A A 3 4 4 - 4 2 2 B 3 4 - 4 - - A A 3 - - -	114	54	94.46	46.15	71	83	94.52
115	3 4 C F 7	- - - - - - C 2 4 - - A A 3 C 6 - 4 2 6 B 3 4 - 4 - - E A 3 - - -	115	60	100.55	45.12	77	89	100.61
116	3 4 1 1 3	- - - - - - C 2 - - B 3 - - 8 6 9 2 9 2 - - C 2 - - B 3 - - - -	116	50	90.39	40.21	64	75	90.41
117	3 4 1 3 3	- - - - - - 8 2 - - B 2 - - C 6 9 2 B 3 - - 8 2 - - B 2 - - - -	117	50	90.39	40.21	64	75	90.41
118	3 4 1 7 7	- - - - - - 8 A 2 - - B 2 - 8 E 6 9 2 F 3 - - 8 2 - - F 2 - - - -	118	58	98.52	40.54	72	82	98.54
119	3 4 1 B 3	- - - - - - 8 6 - - B 2 1 - C 2 9 2 3 3 1 - 8 6 - - B 2 1 - - -	119	62	102.59	40.55	75	85	102.60

Fig. 17: Walsh spectrum decomposition and attack complexity (1)

id	mask of X1	mask of Y0	id'	d1	comp.1	# mul	d2	d3	comp.2
120	3 4 1 F 7	- - - - - 8 A 6 - - B 2 1 8 E 2 9 2 7 3 1 - 8 6 - - F 2 1 - - -	120	70	110.70	40.82	83	92	110.71
121	3 4 5 1 3	- - - - - - - C 2 - - B B 2 - 8 6 D 2 9 A 2 - C 2 - - B B 2 - - -	121	62	102.59	40.55	75	85	102.60
122	3 4 5 3 3	- - - - - - 8 2 - - B A 2 - C 6 D 2 B B 2 - 8 2 - - B A 2 - - -	119	62		40.55	75	85	102.60
123	3 4 5 7 7	- - - - - 8 A 2 - - B A 2 8 E 6 D 2 F B 2 - 8 2 - - F A 2 - - -	120	70		40.82	83	92	110.71
124	3 4 5 B 3	- - - - - - 8 6 - - B A 3 - C 2 D 2 3 B 3 - 8 6 - - B A 3 - - -	119	62		40.55	75	85	102.60
125	3 4 5 F 7	- - - - - 8 A 6 - - B A 3 8 E 2 D 2 7 B 3 - 8 6 - - F A 3 - - -	120	70		40.82	83	92	110.71
126	3 4 9 1 3	- - - - - 4 C 2 - - A 3 - 4 8 6 1 2 8 2 - 4 C 2 - - A 3 - - - -	126	54	94.46	46.20	69	80	94.48
127	3 4 9 3 3	- - - - - 4 8 2 - - A 2 - 4 C 6 1 2 A 3 - 4 8 2 - - A 2 - - - -	127	56	96.49	44.23	69	80	96.51
128	3 4 9 7 7	- - - - - C A 2 - - A 2 - C E 6 1 2 E 3 - 4 8 2 - - E 2 - - - -	128	60	100.55	42.50	73	82	100.57
129	3 4 9 B 3	- - - - - 4 8 6 - - A 2 1 4 C 2 1 2 2 3 1 4 8 6 - - A 2 1 - - -	129	68	108.67	44.51	80	90	108.69
130	3 4 9 F 7	- - - - - C A 6 - - A 2 1 C E 2 1 2 6 3 1 4 8 6 - - E 2 1 - - -	130	72	112.73	42.78	84	92	112.74
131	3 4 D 1 3	- - - - - 4 C 2 - - A B 2 4 8 6 5 2 8 A 2 4 C 2 - - A B 2 - - -	131	66	106.64	46.47	80	90	106.66
132	3 4 D 3 3	- - - - - 4 8 2 - - A A 2 4 C 6 5 2 A B 2 4 8 2 - - A A 2 - - -	129	68		44.51	80	90	108.69
133	3 4 D 7 7	- - - - - C A 2 - - A A 2 C E 6 5 2 E B 2 4 8 2 - - E A 2 - - -	130	72		42.78	84	92	112.74
134	3 4 D B 3	- - - - - 4 8 6 - - A A 3 4 C 2 5 2 2 B 3 4 8 6 - - A A 3 - - -	129	68		44.51	80	90	108.69
135	3 4 D F 7	- - - - - C A 6 - - A A 3 C E 2 5 2 6 B 3 4 8 6 - - E A 3 - - -	130	72		42.78	84	92	112.74
136	3 9 B 1 3	- - - - - C E 2 - - 2 1 - C A B 3 2 - - - C E 2 - - 2 1 - - - -	136	54	94.46	40.43	68	78	94.48
137	3 9 B 3 3	- - - - - C A 2 - - 2 - - C E B 3 2 2 1 - C A 2 - - 2 - - - - -	137	54	94.46	40.28	67	77	94.48
138	3 9 B 7 7	- - - - - 4 8 2 - - 2 - - 4 C B 3 2 6 1 - C A 2 - - 6 - - - - -	138	50	90.39	41.92	65	77	90.42
139	3 9 B B 3	- - - - - C A 6 - - 2 - 1 C E F 3 2 A 1 1 C A 6 - - 2 - 1 - - -	139	66	106.64	40.61	79	88	106.66
140	3 9 B F 7	- - - - - 4 8 6 - - 2 - 1 4 C F 3 2 E 1 1 C A 6 - - 6 - 1 - - -	140	62	102.59	42.26	77	88	102.61
141	3 9 2 1 3	- - - - - 8 6 - - - 3 1 - 8 2 9 A 2 1 - - 8 6 - - - 3 1 - - - -	141	50	90.39	40.21	65	76	90.41
142	3 9 2 3 3	- - - - - 8 2 - - - 3 - - 8 6 9 A 2 3 1 - 8 2 - - - 3 - - - - -	142	46	86.32	39.94	61	73	86.35
143	3 9 2 7 7	- - - - - - - - - - 3 - - - 4 9 A 2 7 1 - 8 2 - - - 7 - - - - -	143	36	76.13	39.24	52	66	76.23
144	3 9 2 B 3	- - - - - 8 2 4 - - 3 - 1 8 6 D A 2 B 1 1 8 2 4 - - 3 - 1 - - -	144	62	102.59	43.17	77	87	102.60
145	3 9 2 F 7	- - - - - - - 4 - - 3 - 1 - 4 D A 2 F 1 1 8 2 4 - - 7 - 1 - - -	145	52	92.43	41.93	68	81	92.48
146	3 9 6 1 3	- - - - - 8 6 - - - 3 9 2 8 2 9 E 2 1 8 2 8 6 - - - 3 9 2 - - -	146	64	104.61	42.46	78	89	104.63
147	3 9 6 3 3	- - - - - 8 2 - - - 3 8 2 8 6 9 E 2 3 9 2 8 2 - - - 3 8 2 - - -	147	62	102.59	43.57	76	87	102.60
148	3 9 6 7 7	- - - - - - - 4 - - 3 8 2 - 4 9 E 2 7 9 2 8 2 - - - 7 8 2 - - -	148	52	92.43	42.75	67	81	92.48
149	3 9 6 B 3	- - - - - 8 2 4 - - 3 8 3 8 6 D E 2 B 9 3 8 2 4 - - 3 8 3 - - -	149	66	106.64	46.80	80	89	106.66
150	3 9 6 F 7	- - - - - - - 4 - - 3 8 3 - 4 D E 2 F 9 3 8 2 4 - - 7 8 3 - - -	150	56	96.49	44.96	71	83	96.52
151	3 9 A 1 3	- - - - - C 6 - - - 2 1 - C 2 9 2 2 - - - C 6 - - - 2 1 - - - -	151	46	86.32	40.13	61	73	86.35
152	3 9 A 3 3	- - - - - C 2 - - - 2 - - C 6 9 2 2 2 1 - C 2 - - - 2 - - - - -	152	46	86.32	39.94	60	72	86.34
153	3 9 A 7 7	- - - - - 4 - - - - 2 - - 4 4 9 2 2 6 1 - C 2 - - - 6 - - - - -	153	40	80.21	40.73	55	69	80.27
154	3 9 A B 3	- - - - - C 2 4 - - 2 - 1 C 6 D 2 2 A 1 1 C 2 4 - - 2 - 1 - - -	154	62	102.59	43.17	76	86	102.60
155	3 9 A F 7	- - - - - 4 - 4 - - 2 - 1 4 4 D 2 2 E 1 1 C 2 4 - - 6 - 1 - - -	155	56	96.49	43.93	71	84	96.53
156	3 9 E 1 3	- - - - - C 6 - - - 2 9 2 C 2 9 6 2 - 8 2 C 6 - - - 2 9 2 - - -	156	60	100.55	42.40	75	87	100.58
157	3 9 E 3 3	- - - - - C 2 - - - 2 8 2 C 6 9 6 2 2 9 2 C 2 - - - 2 8 2 - - -	157	62	102.59	43.57	75	86	102.60
158	3 9 E 7 7	- - - - - 4 - - - - 2 8 2 4 4 9 6 2 6 9 2 C 2 - - - 6 8 2 - - -	158	56	96.49	44.74	70	84	96.53
159	3 9 E B 3	- - - - - C 2 4 - - 2 8 3 C 6 D 6 2 A 9 3 C 2 4 - - 2 8 3 - - -	159	66	106.64	46.80	79	88	106.66
160	3 9 E F 7	- - - - - 4 - 4 - - 2 8 3 4 4 D 6 2 E 9 3 C 2 4 - - 6 8 3 - - -	160	60	100.55	47.49	74	86	100.57
161	6 4 C 8 2	- - - - - C 6 4 - - F B 3 C 2 - C - 1 A 3 C 6 4 - - B B 3 - - -	161	60	100.55	45.11	76	86	100.57
162	6 4 C 2 2	- - - - - C 2 - - - F A 2 C 6 4 C - B B 2 C 2 - - - B A 2 - - -	162	58	98.52	44.17	73	84	98.54
163	6 4 C 6 6	- - - - - 4 - - - - F A 2 4 4 4 C - F B 2 C 2 - - - F A 2 - - -	163	52	92.43	45.70	68	81	92.48
164	6 4 C A 2	- - - - - C 2 4 - - F A 3 C 6 - C - 3 B 3 C 2 4 - - B A 3 - - -	161	60		45.11	76	86	100.57
165	6 4 C E 6	- - - - - 4 - 4 - - F A 3 4 4 - C - 7 B 3 C 2 4 - - F A 3 - - -	165	54	94.46	46.15	71	83	94.52
166	6 4 1 8 2	- - - - - 8 E 6 - - E 3 1 8 A 2 1 - - 2 1 8 E 6 - - A 3 1 - - -	166	66	106.64	40.73	81	92	106.66
167	6 4 1 2 2	- - - - - 8 A 2 - - E 2 - 8 E 6 1 - A 3 - 8 A 2 - - A 2 - - - -	167	58	98.52	40.49	72	82	98.54
168	6 4 1 6 6	- - - - - - 8 2 - - E 2 - - C 6 1 - E 3 - 8 A 2 - - E 2 - - - -	168	50	90.39	40.15	66	77	90.42
169	6 4 1 A 2	- - - - - 8 A 6 - - E 2 1 8 E 2 1 - 2 3 1 8 A 6 - - A 2 1 - - -	169	70	110.70	40.78	83	93	110.71
170	6 4 1 E 6	- - - - - - 8 6 - - E 2 1 - C 2 1 - 6 3 1 8 A 6 - - E 2 1 - - -	170	62	102.59	40.51	77	88	102.61
171	6 4 5 8 2	- - - - - 8 E 6 - - E B 3 8 A 2 5 - - A 3 8 E 6 - - A B 3 - - -	166	66		40.73	81	92	106.66
172	6 4 5 2 2	- - - - - 8 A 2 - - E A 2 8 E 6 5 - A B 2 8 A 2 - - A A 2 - - -	169	70		40.78	83	93	110.71
173	6 4 5 6 6	- - - - - - 8 2 - - E A 2 - C 6 5 - E B 2 8 A 2 - - E A 2 - - -	170	62		40.51	77	88	102.61
174	6 4 5 A 2	- - - - - 8 A 6 - - E A 3 8 E 2 5 - 2 B 3 8 A 6 - - A A 3 - - -	169	70		40.78	83	93	110.71
175	6 4 5 E 6	- - - - - - 8 6 - - E A 3 - C 2 5 - 6 B 3 8 A 6 - - E A 3 - - -	170	62		40.51	77	88	102.61
176	6 4 9 8 2	- - - - - C E 6 - - F 3 1 C A 2 9 - 1 2 1 C E 6 - - B 3 1 - - -	176	70	110.70	40.78	82	91	110.71
177	6 4 9 2 2	- - - - - C A 2 - - F 2 - C E 6 9 - B 3 - C A 2 - - B 2 - - - -	177	58	98.52	40.49	71	80	98.54
178	6 4 9 6 6	- - - - - 4 8 2 - - F 2 - 4 C 6 9 - F 3 - C A 2 - - F 2 - - - -	178	54	94.46	42.23	69	80	94.48
179	6 4 9 A 2	- - - - - C A 6 - - F 2 1 C E 2 9 - 3 3 1 C A 6 - - B 2 1 - - -	176	70		40.78	82	91	110.71
180	6 4 9 E 6	- - - - - 4 8 6 - - F 2 1 4 C 2 9 - 7 3 1 C A 6 - - F 2 1 - - -	180	66	106.64	42.51	80	91	106.66
181	6 4 D 8 2	- - - - - C E 6 - - F B 3 C A 2 D - 1 A 3 C E 6 - - B B 3 - - -	176	70		40.78	82	91	110.71
182	6 4 D 2 2	- - - - - C A 2 - - F A 2 C E 6 D - B B 2 C A 2 - - B A 2 - - -	176	70		40.78	82	91	110.71
183	6 4 D 6 6	- - - - - 4 8 2 - - F A 2 4 C 6 D - F B 2 C A 2 - - F A 2 - - -	180	66		42.51	80	91	106.66
184	6 4 D A 2	- - - - - C A 6 - - F A 3 C E 2 D - 3 B 3 C A 6 - - B A 3 - - -	176	70		40.78	82	91	110.71
185	6 4 D E 6	- - - - - 4 8 6 - - F A 3 4 C 2 D - 7 B 3 C A 6 - - F A 3 - - -	180	66		42.51	80	91	106.66
186	6 9 B 8 2	- - - - - 4 C 6 - - 7 1 1 4 8 F B - 9 - 1 4 C 6 - - 3 1 1 - - -	186	60	100.55	44.48	76	88	100.59
187	6 9 B 2 2	- - - - - 4 8 2 - - 7 - - 4 C B B - 3 1 - 4 8 2 - - 3 - - - - -	187	44	84.29	43.88	60	73	84.34
188	6 9 B 6 6	- - - - - C A 2 - - 7 - - C E B B - 7 1 - 4 8 2 - - 7 - - - - -	188	48	88.36	42.16	64	75	88.39
189	6 9 B A 2	- - - - - 4 8 6 - - 7 - 1 4 C F B - B 1 1 4 8 6 - - 3 - 1 - - -	189	56	96.49	44.23	72	85	96.55
190	6 9 B E 6	- - - - - C A 6 - - 7 - 1 C E F B - F 1 1 4 8 6 - - 7 - 1 - - -	190	60	100.55	42.50	76	87	100.58
191	6 9 2 8 2	- - - - - - 4 4 - - 6 1 1 - - D 2 - 8 - 1 - 4 4 - - 2 1 1 - - -	191	46	86.32	42.72	64	77	86.50
192	6 9 2 2 2	- - - - - - - - - - 6 - - - 4 9 2 - 2 1 - - - - - - 2 - - - - -	192	26	65.91	37.16	43	57	66.09
193	6 9 2 6 6	- - - - - 8 2 - - - 6 - - - 8 6 9 2 - 6 1 - - - - - - 6 - - - - -	193	36	76.13	37.90	53	67	76.31
194	6 9 2 A 2	- - - - - - 4 - - 6 - 1 - 4 D 2 - A 1 1 - - 4 - - 2 - 1 - - -	194	42	82.25	39.85	59	72	82.35
195	6 9 2 E 6	- - - - - 8 2 4 - - 6 - 1 8 6 D 2 - E 1 1 - - 4 - - 6 - 1 - - -	195	52	92.43	41.14	69	82	92.53
196	6 9 6 8 2	- - - - - - 4 4 - - 6 9 3 - - D 6 - 8 8 3 - 4 4 - - 2 9 3 - - -	196	48	88.36	44.67	66	79	88.54
197	6 9 6 2 2	- - - - - - - - - - - 6 8 2 - 4 9 6 - 2 9 2 - - - - - 2 8 2 - - -	197	42	82.25	40.73	58	72	82.35
198	6 9 6 6 6	- - - - - 8 2 - - - 6 8 2 8 6 9 6 - 6 9 2 - - - - - 6 8 2 - - -	198	52	92.43	41.57	68	82	92.53
199	6 9 6 A 2	- - - - - - 4 - - - 6 8 3 - 4 D 6 - A 9 3 - - 4 - - 2 8 3 - - -	199	46	86.32	42.92	62	75	86.38
200	6 9 6 E 6	- - - - - 8 2 4 - - 6 8 3 8 6 D 6 - E 9 3 - - 4 - - 6 8 3 - - -	200	56	96.49	44.79	72	85	96.55
201	6 9 A 8 2	- - - - - 4 4 4 - - 7 1 1 4 - D A - 9 - 1 4 4 4 - - 3 1 1 - - -	201	54	94.46	45.31	70	83	94.52
202	6 9 A 2 2	- - - - - 4 - - - - 7 - - - 4 4 9 A - 3 1 - 4 - - - - - 3 - - - - -	202	32	72.04	40.67	48	62	72.15
203	6 9 A 6 6	- - - - - C 2 - - - 7 - - - C 6 9 A - 7 1 - 4 - - - - - 7 - - - - -	203	38	78.17	39.82	54	68	78.27
204	6 9 A A 2	- - - - - 4 - 4 - - 7 - 1 4 4 D A - B 1 1 4 - 4 - - 3 - 1 - - -	204	48	88.36	43.90	64	77	88.42
205	6 9 A E 6	- - - - - C 2 4 - - 7 - 1 C 6 D A - F 1 1 4 - 4 - - 7 - 1 - - -	205	54	94.46	43.12	70	83	94.52
206	6 9 E 8 2	- - - - - 4 4 4 - - 7 9 3 4 - D E - 9 8 3 4 4 4 - - 3 9 3 - - -	206	56	96.49	47.28	71	84	96.53
207	6 9 E 2 2	- - - - - 4 - - - - 7 8 2 4 4 9 E - 3 9 2 4 - - - - - 3 8 2 - - -	207	48	88.36	44.73	63	77	88.42
208	6 9 E 6 6	- - - - - C 2 - - - 7 8 2 C 6 9 E - 7 9 2 4 - - - - - 7 8 2 - - -	208	54	94.46	43.54	69	83	94.52
209	6 9 E A 2	- - - - - 4 - - - 7 8 3 4 4 D E - B 9 3 4 - 4 - - 3 8 3 - - -	209	52	92.43	47.48	67	80	92.46
210	6 9 E E 6	- - - - - C 2 4 - - 7 8 3 C 6 D E - F 9 3 4 - 4 - - 7 8 3 - - -	210	58	98.52	46.79	73	86	98.56
211	7 4 C 1 3	- - - - - C 6 - - - E B 2 C 2 4 4 2 8 A 2 C 6 - - - A B 2 - - -	211	60	100.55	46.16	76	87	100.58
212	7 4 C 3 3	- - - - - C 2 - - - E A 2 C 6 4 4 2 A B 2 C 2 - - - A A 2 - - -	212	62	102.59	44.17	76	87	102.60
213	7 4 C 7 7	- - - - - 4 - - - - E A 2 4 4 4 4 2 E B 2 C 2 - - - E A 2 - - -	213	56	96.49	45.71	72	85	96.55
214	7 4 C B 3	- - - - - C 2 4 - - E A 3 C 6 - 4 2 2 B 3 C 2 4 - - A A 3 - - -	214	64	104.61	45.12	79	89	104.63
215	7 4 C F 7	- - - - - 4 - 4 - - E A 3 4 4 - 4 2 6 B 3 C 2 4 - - E A 3 - - -	215	58	98.52	46.16	75	87	98.58
216	7 4 1 1 3	- - - - - 8 E 2 - - F 3 - 8 A 6 9 2 9 2 - 8 E 2 - - B 3 - - - -	4	62		40.55	74	84	102.60
217	7 4 1 3 3	- - - - - 8 A 2 - - F 2 - 8 E 6 9 2 B 3 - 8 A 2 - - B 2 - - - -	4	62		40.55	74	84	102.60
218	7 4 1 7 7	- - - - - - 8 2 - - F 2 - - C 6 9 2 F 3 - 8 A 2 - - F 2 - - - -	218	54	94.46	40.22	68	79	94.48
219	7 4 1 B 3	- - - - - 8 A 6 - - F 2 1 8 E 2 9 2 3 1 8 A 6 - - B 2 1 - - -	6	74		40.83	85	94	114.77
220	7 4 1 F 7	- - - - - 8 6 - - F 2 1 - C 2 9 2 7 3 1 8 A 6 - - F 2 1 - - -	220	66	106.64	40.57	79	89	106.66
221	7 4 5 1 3	- - - - - 8 E 2 - - F B 2 8 A 6 D 2 9 A 2 8 E 2 - - B B 2 - - -	6	74		40.83	85	94	114.77
222	7 4 5 3 3	- - - - - 8 A 2 - - F A 2 8 E 6 D 2 B B 2 8 A 2 - - B A 2 - - -	6	74		40.83	85	94	114.77
223	7 4 5 7 7	- - - - - - 8 2 - - F A 2 - C 6 D 2 F B 2 8 A 2 - - F A 2 - - -	220	66		40.57	79	89	106.66
224	7 4 5 B 3	- - - - - 8 A 6 - - F A 3 8 E 2 D 2 3 B 3 8 A 6 - - B A 3 - - -	6	74		40.83	85	94	114.77
225	7 4 5 F 7	- - - - - - 8 6 - - F A 3 - C 2 D 2 7 B 3 8 A 6 - - F A 3 - - -	220	66		40.57	79	89	106.66
226	7 4 9 1 3	- - - - - C E 2 - - E 3 - C A 6 1 2 8 2 - C E 2 - - - A 3 - - -	226	60	100.55	42.40	73	82	100.57
227	7 4 9 3 3	- - - - - C A 2 - - E 2 - C E 6 1 2 A 3 - C A 2 - - - A 2 - - -	227	62	102.59	40.55	73	82	102.60
228	7 4 9 7 7	- - - - - 4 8 2 - - E 2 - 4 C 6 1 2 E 3 - C A 2 - - - E 2 - - -	228	58	98.52	42.25	71	82	98.54
229	7 4 9 B 3	- - - - - C A 6 - - E 2 1 C E 2 1 2 3 1 C A 6 - - - A 2 1 - - -	229	74	114.75	40.83	84	92	114.77
230	7 4 9 F 7	- - - - - 4 8 6 - - E 2 1 4 C 2 1 2 6 3 1 C A 6 - - - E 2 1 - - -	230	70	110.70	42.53	82	92	110.71
231	7 4 D 1 3	- - - - - C E 2 - - E B 2 C A 6 5 2 8 A 2 C E 2 - - - A B 2 - - -	231	72	112.73	42.65	84	92	112.74
232	7 4 D 3 3	- - - - - C A 2 - - E A 2 C E 6 5 2 A B 2 C A 2 - - - A A 2 - - -	229	74		40.83	84	92	114.77
233	7 4 D 7 7	- - - - - 4 8 2 - - E A 2 4 C 6 5 2 E B 2 C A 2 - - - E A 2 - - -	230	70		42.53	82	92	110.71
234	7 4 D B 3	- - - - - C A 6 - - E A 3 C E 2 5 2 2 B 3 C A 6 - - - A A 3 - - -	229	74		40.83	84	92	114.77
235	7 4 D F 7	- - - - - 4 8 6 - - E A 3 4 C 2 5 2 6 B 3 C A 6 - - - E A 3 - - -	230	70		42.53	82	92	110.71
236	7 9 B 1 3	- - - - - 4 C 2 - - 6 1 - 4 8 B 3 2 - - - 4 C 2 - - - 2 1 - - -	236	48	88.36	44.19	64	76	88.40
237	7 9 B 3 3	- - - - - 4 8 2 - - 6 - - 4 C B 3 2 2 1 - 4 8 2 - - - 2 - - - - -	237	48	88.36	43.90	63	75	88.39
238	7 9 B 7 7	- - - - - C A 2 - - 6 - - C E B 3 2 6 1 - 4 8 2 - - - 6 - - - - -	238	52	92.43	42.23	67	77	92.45
239	7 9 B B 3	- - - - - 4 8 6 - - 6 - 1 4 C F 3 2 A 1 1 4 8 6 - - - 2 - 1 - - -	239	60	100.55	44.25	75	86	100.57

Fig. 18: Walsh spectrum decomposition and attack complexity (2)

id	mask of X1	mask of Y0	id'	d1	comp.1	# mul	d2	d3	comp.2
240	7 9 B F 7	- - - - - C A 6 - - 6 - 1 C E F 3 2 E 1 1 4 8 6 - - 6 - 1 - - -	240	64	104.61	42.56	79	88	104.63
241	7 9 2 1 3	- - - - - - - 4 - - - 7 1 - - - 9 A 2 1 - - - 4 - - - 3 1 - - -	241	36	76.13	38.72	52	66	76.23
242	7 9 2 3 3	- - - - - - - - - - - 7 - - - 4 9 A 2 3 1 - - - - - - - 3 - - - - -	242	30	70.00	37.31	46	60	70.10
243	7 9 2 7 7	- - - - - - 8 2 - - - 7 - - 8 6 9 A 2 7 1 - - - - - - - 7 - - - - -	243	40	80.21	37.98	56	70	80.31
244	7 9 2 B 3	- - - - - - - - 4 - - 7 - 1 - 4 D A 2 B 1 1 - - 4 - - 3 - 1 - - -	244	46	86.32	39.94	62	75	86.38
245	7 9 2 F 7	- - - - - - 8 2 4 - - 7 - 1 8 6 D A 2 F 1 1 - - 4 - - 7 - 1 - - -	245	56	96.49	41.18	72	85	96.55
246	7 9 6 1 3	- - - - - - - 4 - - - 7 9 2 - - 9 E 2 1 8 2 - 4 - - - 3 9 2 - - -	246	50	90.39	40.90	65	79	90.45
247	7 9 6 3 3	- - - - - - - - - - 7 8 2 - 4 9 E 2 3 9 2 - - - - - 3 8 2 - - -	247	46	86.32	40.78	61	75	86.38
248	7 9 6 7 7	- - - - - - 8 2 - - - 7 8 2 8 6 9 E 2 7 9 2 - - - - - 7 8 2 - - -	248	56	96.49	41.59	71	85	96.55
249	7 9 6 B 3	- - - - - - - - 4 - - 7 8 3 - 4 D E 2 B 9 3 - - 4 - - 3 8 3 - - -	249	50	90.39	42.96	65	77	90.42
250	7 9 6 F 7	- - - - - - 8 2 4 - - 7 8 3 8 6 D E 2 F 9 3 - - 4 - - 7 8 3 - - -	250	60	100.55	44.80	75	87	100.58
251	7 9 A 1 3	- - - - - - 4 4 - - - 6 1 - 4 - 9 2 2 - - - 4 4 - - - 2 1 - - - -	251	38	78.17	42.06	54	68	78.27
252	7 9 A 3 3	- - - - - - 4 - - - - 6 - - 4 4 9 2 2 2 1 - 4 - - - - - 2 - - - - -	252	36	76.13	40.72	51	65	76.19
253	7 9 A 7 7	- - - - - - C 2 - - - 6 - - C 6 9 2 2 6 1 - 4 - - - - 6 - - - - -	253	42	82.25	39.92	57	71	82.31
254	7 9 A B 3	- - - - - - 4 - 4 - - 6 - 1 4 4 D 2 2 A 1 1 4 - 4 - - 2 - 1 - - -	254	52	92.43	43.93	67	80	92.46
255	7 9 A F 7	- - - - - - C 2 4 - - 6 - 1 C 6 D 2 2 E 1 1 4 - 4 - - 6 - 1 - - -	255	58	98.52	43.16	73	86	98.56
256	7 9 E 1 3	- - - - - - 4 4 - - - 6 9 2 4 - 9 6 2 - 8 2 4 4 - - - 2 9 2 - - -	256	52	92.43	44.37	68	82	92.53
257	7 9 E 3 3	- - - - - - 4 - - - - 6 8 2 4 4 9 6 2 2 9 2 4 - - - - 2 8 2 - - -	257	52	92.43	44.74	66	80	92.46
258	7 9 E 7 7	- - - - - - C 2 - - - 6 8 2 C 6 9 6 2 6 9 2 4 - - - - - 6 8 2 - - -	258	58	98.52	43.57	72	86	98.56
259	7 9 E B 3	- - - - - - 4 - 4 - - 6 8 3 4 4 D 6 2 A 9 3 4 - 4 - - 2 8 3 - - -	259	56	96.49	47.49	70	82	96.51
260	7 9 E F 7	- - - - - - C 2 4 - - 6 8 3 C 6 D 6 2 E 9 3 4 - 4 - - 6 8 3 - - -	260	62	102.59	46.80	76	88	102.61
261	A 4 C 8 A	- - - - - - 4 4 - - - 3 B 2 4 - - C - 1 A 3 4 4 - - - 3 B 2 - - -	261	48	88.36	44.19	65	78	88.46
262	A 4 C 2 A	- - - - - - 4 - 4 - - 3 A 3 4 4 4 C - B B 2 4 - 4 - - 3 A 3 - - -	262	52	92.43	48.14	68	80	92.46
263	A 4 C 6 E	- - - - - - C 2 4 - - 3 A 3 C 6 4 C - F B 2 4 - 4 - - 7 A 3 - - -	263	58	98.52	47.09	74	86	98.56
264	A 4 C A A	- - - - - - 4 - - - - 3 A 2 4 4 - C - 3 B 3 4 - - - - 3 A 2 - - -	264	46	86.32	43.72	63	76	86.42
265	A 4 C E E	- - - - - - C 2 - - - 3 A 2 C 6 - C - 7 B 3 4 - - - - 7 A 2 - - -	265	52	92.43	42.22	69	82	92.53
266	A 4 1 8 A	- - - - - - - C 2 - - 2 3 - - 8 2 1 - - 2 1 - C 2 - - 2 3 - - - -	266	46	86.32	40.13	64	76	86.42
267	A 4 1 2 A	- - - - - - 8 6 - - 2 2 1 - C 6 1 - A 3 - - 8 6 - - 2 2 1 - - -	267	54	94.46	40.43	70	81	94.49
268	A 4 1 6 E	- - - - - - 8 A 6 - - 2 2 1 8 E 6 2 2 9 2 4 - - - 8 6 - - 6 2 1 - - -	268	62	102.59	40.72	78	88	102.61
269	A 4 1 A A	- - - - - - 8 2 - - 2 2 - - C 2 1 - 2 3 1 - 8 2 - - 2 2 - - - -	269	50	90.39	40.21	66	77	90.42
270	A 4 1 E E	- - - - - - 8 A 2 - - 2 2 - 8 E 2 1 - 6 3 1 - 8 2 - - 6 2 - - - -	270	58	98.52	40.54	74	84	98.54
271	A 4 5 8 A	- - - - - - - C 2 - - 2 B 2 - 8 2 5 - - A 3 - C 2 - - 2 B 2 - - -	66	54		40.43	71	83	94.52
272	A 4 5 2 A	- - - - - - - 8 6 - - 2 A 3 - C 6 5 - A B 2 - 8 6 - - 2 A 3 - - -	69	58		40.49	73	84	98.54
273	A 4 5 6 E	- - - - - - 8 A 6 - - 2 A 3 8 E 6 5 - E B 2 - 8 6 - - 6 A 3 - - -	70	66		40.77	81	91	106.66
274	A 4 5 A A	- - - - - - - 8 2 - - 2 A 2 - C 2 5 - 2 B 3 - 8 2 - - 2 A 2 - - -	69	58		40.49	73	84	98.54
275	A 4 5 E E	- - - - - - 8 A 2 - - 2 A 2 8 E 2 5 - 6 B 3 - 8 2 - - 6 A 2 - - -	70	66		40.77	81	91	106.66
276	A 4 9 8 A	- - - - - - 4 C 2 - - 3 3 - 4 8 2 9 - 1 2 1 4 C 2 - - 3 3 - - - -	276	56	96.49	44.23	71	82	96.51
277	A 4 9 2 A	- - - - - - 4 8 6 - - 3 2 1 4 C 6 9 - B 3 - 4 8 6 - - 3 2 1 - - -	277	60	100.55	44.48	75	86	100.57
278	A 4 9 6 E	- - - - - - C A 6 - - 3 2 1 C E 6 9 - F 3 - 4 8 6 - - 7 2 1 - - -	278	64	104.61	42.68	79	88	104.63
279	A 4 9 A A	- - - - - - 4 8 2 - - 3 2 - 4 C 2 9 - 3 3 1 4 8 2 - - 3 2 - - - -	279	56	96.49	44.23	71	82	96.51
280	A 4 9 E E	- - - - - - C A 2 - - 3 2 - C E 2 9 - 7 3 1 4 8 2 - - 7 2 - - - -	280	60	100.55	42.50	75	84	100.57
281	A 4 D 8 A	- - - - - - 4 C 2 - - 3 B 2 4 8 2 D - 1 A 3 4 C 2 - - 3 B 2 - - -	76	64		44.50	78	89	104.63
282	A 4 D 2 A	- - - - - - 4 8 6 - - 3 A 3 4 C 6 D - B B 2 4 8 6 - - 3 A 3 - - -	79	64		44.50	78	89	104.63
283	A 4 D 6 E	- - - - - - C A 6 - - 3 A 3 C E 6 D - F B 2 4 8 6 - - 7 A 3 - - -	80	68		42.73	82	91	108.69
284	A 4 D A A	- - - - - - 4 8 2 - - 3 A 2 4 C 2 D - 3 B 3 4 8 2 - - 3 A 2 - - -	79	64		44.50	78	89	104.63
285	A 4 D E E	- - - - - - C A 2 - - 3 A 2 C E 2 D - 7 B 3 4 8 2 - - 7 A 2 - - -	80	68		42.73	82	91	108.69
286	A 9 B 8 A	- - - - - - C E 2 - - B 1 - C A F B - 9 - 1 C E 2 - - B 1 - - - -	286	58	98.52	40.49	72	82	98.54
287	A 9 B 2 A	- - - - - - C A 6 - - B - 1 C E B B - 3 1 - C A 6 - - B - 1 - - -	287	58	98.52	40.49	72	83	98.54
288	A 9 B 6 E	- - - - - - 4 8 6 - - B - 1 4 C B B - 7 1 - C A 6 - - F - 1 - - -	288	54	94.46	42.23	70	83	94.52
289	A 9 B A A	- - - - - - C A 2 - - B - - C E F B - B 1 1 C A 2 - - B - - - - -	289	54	94.46	40.28	68	79	94.48
290	A 9 B E E	- - - - - - 4 8 2 - - B - - 4 C F B - F 1 1 C A 2 - - F - - - - -	290	50	90.39	41.92	66	79	90.45
291	A 9 2 8 A	- - - - - - 8 6 - - - A 1 - 8 2 D 2 - 8 - 1 8 6 - - - A 1 - - - -	291	48	88.36	42.02	65	77	88.42
292	A 9 2 2 A	- - - - - - 8 2 4 - - - A - 1 8 6 9 2 - 2 1 - 8 2 4 - - - A - 1 - - -	292	54	94.46	43.09	70	82	94.50
293	A 9 2 6 E	- - - - - - - - 4 - - - A - 1 - 4 9 2 - 6 1 - 8 2 4 - - - E - 1 - - -	293	44	84.29	41.74	61	74	84.39
294	A 9 2 A A	- - - - - - 8 2 - - - - A - - 8 6 D 2 - A 1 1 8 2 - - - A - - - - -	294	46	86.32	39.94	62	75	86.38
295	A 9 2 E E	- - - - - - - - - - - - A - - - 4 D 2 - E 1 1 8 2 - - - E - - - - -	295	36	76.13	39.24	53	67	76.31
296	A 9 6 8 A	- - - - - - 8 6 - - - A 9 2 8 2 D 6 - 8 8 3 8 6 - - - A 9 2 - - -	296	58	98.52	44.31	75	87	98.58
297	A 9 6 2 A	- - - - - - 8 2 4 - - A 8 3 8 6 9 6 - 2 9 2 8 2 4 - - - A 8 3 - - -	99	62		46.79	77	88	102.61
298	A 9 6 6 E	- - - - - - - 4 - - - A 8 3 - 4 9 6 - 6 9 2 8 2 4 - - - E 8 3 - - -	100	52		44.92	68	81	92.48
299	A 9 6 A A	- - - - - - 8 2 - - - A 8 2 8 6 D 6 - A 9 3 8 2 - - - A 8 2 - - -	97	58		43.55	73	85	98.55
300	A 9 6 E E	- - - - - - - - - - - A 8 2 - 4 D 6 - E 9 3 8 2 - - - E 8 2 - - -	98	48		42.70	64	78	88.46
301	A 9 A 8 A	- - - - - - C 6 - - - B 1 - C 2 D A - 9 - 1 C 6 - - - B 1 - - - -	301	50	90.39	40.21	65	77	90.42
302	A 9 A 2 A	- - - - - - C 2 4 - - B - 1 C 6 9 A - 3 1 - C 2 4 - - B - 1 - - -	302	54	94.46	43.09	69	81	94.49
303	A 9 A 6 E	- - - - - - 4 - 4 - - B - 1 4 4 9 A - 7 1 - C 2 4 - - F - 1 - - -	303	48	88.36	43.89	64	77	88.42
304	A 9 A A A	- - - - - - C 2 - - - B - - - C 6 D A - B 1 1 C 2 - - - B - - - - -	304	46	86.32	39.94	61	74	86.36
305	A 9 A E E	- - - - - - 4 - - - - B - - 4 4 D A - F 1 1 C 2 - - - F - - - - -	305	40	80.21	40.73	56	70	80.31
306	A 9 E 8 A	- - - - - - C 6 - - - B 9 2 C 2 D E - 9 8 3 C 6 - - - B 9 2 - - -	306	60	100.55	42.40	74	86	100.57
307	A 9 E 2 A	- - - - - - C 2 4 - - B 8 3 C 6 9 E - 3 9 2 C 2 4 - - B 8 3 - - -	109	62		46.79	76	87	102.60
308	A 9 E 6 E	- - - - - - 4 - 4 - - B 8 3 4 4 9 E - 7 9 2 C 2 4 - - F 8 3 - - -	110	56		47.48	71	84	96.53
309	A 9 E A A	- - - - - - C 2 - - - B 8 2 C 6 D E - B 9 3 C 2 - - - B 8 2 - - -	107	58		43.55	72	84	98.54
310	A 9 E E E	- - - - - - 4 - - - - B 8 2 4 4 D E - F 9 3 C 2 - - - F 8 2 - - -	108	52		44.73	67	81	92.48
311	B 4 C 1 B	- - - - - - 4 4 4 - - 2 B 3 4 - 4 4 2 8 A 2 4 4 4 - - 2 B 3 - - -	311	56	96.49	51.17	74	86	96.59
312	B 4 C 3 B	- - - - - - 4 - 4 - - 2 A 3 4 4 4 4 2 A B 2 4 - 4 - - 2 A 3 - - -	312	56	96.49	48.14	72	84	96.53
313	B 4 C 7 F	- - - - - - C 2 4 - - 2 A 3 C 6 4 4 2 E B 2 4 - 4 - - 6 A 3 - - -	313	62	102.59	47.09	78	90	102.62
314	B 4 C B B	- - - - - - 4 - - - - 2 A 2 4 4 - 4 2 2 B 3 4 - - - - 2 A 2 - - -	314	50	90.39	43.73	67	80	90.49
315	B 4 C F F	- - - - - - C 2 - - - 2 A 2 C 6 - 4 2 6 B 3 4 - - - - 6 A 2 - - -	315	56	96.49	42.24	73	86	96.59
316	B 4 1 1 B	- - - - - - - C 6 - - 3 3 1 - 8 6 9 2 9 2 - - C 6 - - 3 3 1 - - -	316	58	98.52	40.49	72	82	98.54
317	B 4 1 3 B	- - - - - - - 8 6 - - 3 2 1 - C 6 9 2 B 3 - - 8 6 - - 3 2 1 - - -	317	58	98.52	40.49	72	82	98.54
318	B 4 1 7 F	- - - - - - 8 A 6 - - 3 2 1 8 E 6 9 2 F 3 - - 8 6 - - 7 2 1 - - -	318	66	106.64	40.77	80	89	106.66
319	B 4 1 B B	- - - - - - 4 8 2 - - 3 2 - - C E 2 9 2 3 1 1 - 8 2 - - 3 2 - - - -	339	54	94.46	40.28	68	79	94.48
320	B 4 1 F F	- - - - - - 8 A 2 - - 3 2 - 8 E 2 9 2 7 3 1 - 8 2 - - 7 2 - - - -	320	62	102.59	40.60	76	86	102.60
321	B 4 5 1 B	- - - - - - - C 6 - - 3 B 3 - 8 6 D 2 9 A 2 - C 6 - - 3 B 3 - - -	121	62		40.55	75	85	102.60
322	B 4 5 3 B	- - - - - - - 8 6 - - 3 A 3 - C 6 D 2 B B 2 - 8 6 - - 3 A 3 - - -	119	62		40.55	75	85	102.60
323	B 4 5 7 F	- - - - - - 8 A 6 - - 3 A 3 8 E 6 D 2 F B 2 - 8 6 - - 7 A 3 - - -	120	70		40.82	83	92	110.71
324	B 4 5 B B	- - - - - - - 8 2 - - 3 A 2 - C 2 D 2 3 B 3 - 8 2 - - 3 A 2 - - -	119	62		40.55	75	85	102.60
325	B 4 5 F F	- - - - - - 8 A 2 - - 3 A 2 8 E 2 D 2 7 B 3 - 8 2 - - 7 A 2 - - -	120	70		40.82	83	92	110.71
326	B 4 9 1 B	- - - - - - 4 C 6 - - 2 3 1 4 8 6 1 2 8 2 - 4 C 6 - - 2 3 1 - - -	326	62	102.59	46.47	77	87	102.60
327	B 4 9 3 B	- - - - - - 4 8 6 - - 2 2 1 4 C 6 1 2 A 3 - 4 8 6 - - 2 2 1 - - -	327	64	104.61	44.50	77	87	104.63
328	B 4 9 7 F	- - - - - - C A 6 - - 2 2 1 C E 6 1 2 E 3 - 4 8 6 - - 6 2 1 - - -	328	68	108.67	42.73	81	89	108.69
329	B 4 9 B B	- - - - - - 4 8 2 - - 2 2 - 4 C 2 1 2 3 1 4 8 2 - - - 2 2 - - - -	329	60	100.55	44.25	73	84	100.57
330	B 4 9 F F	- - - - - - C A 2 - - 2 2 - C E 2 1 2 6 3 1 4 8 2 - - - 6 2 - - - -	330	64	104.61	42.56	77	86	104.63
331	B 4 D 1 B	- - - - - - 4 C 6 - - 2 B 3 4 8 6 5 2 8 A 2 4 C 6 - - 2 B 3 - - -	131	66		46.47	80	90	106.66
332	B 4 D 3 B	- - - - - - 4 8 6 - - 2 A 3 4 C 6 5 2 A B 2 4 8 6 - - 2 A 3 - - -	129	68		44.51	80	90	108.69
333	B 4 D 7 F	- - - - - - C A 6 - - 2 A 3 C E 6 5 2 E B 2 4 8 6 - - 6 A 3 - - -	130	72		42.78	84	92	112.74
334	B 4 D B B	- - - - - - 4 8 2 - - 2 A 2 4 C 2 5 2 2 B 3 4 8 2 - - 2 A 2 - - -	129	68		44.51	80	90	108.69
335	B 4 D F F	- - - - - - C A 2 - - 2 A 2 C E 2 5 2 6 B 3 4 8 2 - - 6 A 2 - - -	130	72		42.78	84	92	112.74
336	B 9 B 1 B	- - - - - - C E 6 - - A 1 1 C A B 3 2 - - - C E 6 - - A 1 1 - - -	336	62	102.59	40.68	76	85	102.60
337	B 9 B 3 B	- - - - - - C A 6 - - A - 1 C E B 3 2 2 1 - C A 6 - - A - 1 - - -	337	62	102.59	40.55	75	84	102.60
338	B 9 B 7 F	- - - - - - 4 8 6 - - A - 1 4 C B 3 2 6 1 - C A 6 - - E - 1 - - -	338	58	98.52	42.25	73	84	98.54
339	B 9 B B B	- - - - - - C A 2 - - A - - C E 7 9 2 3 1 1 C A 2 - - A - - - - -	339	58	98.52	40.35	71	81	98.54
340	B 9 B F F	- - - - - - 4 8 2 - - A - - 4 C F 3 2 E 1 1 C A 2 - - E - - - - -	340	54	94.46	41.94	69	81	94.49
341	B 9 2 1 B	- - - - - - 8 6 4 - - B 1 1 8 2 9 A 2 1 - - 8 6 4 - - B 1 1 - - -	341	62	102.59	43.25	77	86	102.60
342	B 9 2 3 B	- - - - - - 8 2 4 - - B - 1 8 6 9 A 2 3 1 - 8 2 4 - - B - 1 - - -	342	58	98.52	43.13	73	83	98.54
343	B 9 2 7 F	- - - - - - - - 4 - - B - 1 - 4 9 A 2 7 1 - 8 2 4 - - F - 1 - - -	343	48	88.36	41.84	64	77	88.42
344	B 9 2 B B	- - - - - - 8 2 - - - B - - - 8 6 D A 2 B 1 1 8 2 - - - B - - - - -	344	50	90.39	40.03	65	77	90.42
345	B 9 2 F F	- - - - - - - - - - - B - - - 4 D A 2 F 1 1 8 2 - - - F - - - - -	345	40	80.21	39.38	56	70	80.31
346	B 9 6 1 B	- - - - - - 8 6 4 - - B 9 3 8 2 9 E 2 1 8 2 8 6 4 - - B 9 3 - - -	346	68	108.67	45.25	82	91	108.69
347	B 9 6 3 B	- - - - - - 8 2 4 - - B 8 3 8 6 9 E 2 3 9 2 8 2 4 - - B 8 3 - - -	149	66		46.80	80	89	106.66
348	B 9 6 7 F	- - - - - - - - 4 - - B 8 3 - 4 9 E 2 7 9 2 8 2 4 - - F 8 3 - - -	150	56		44.96	71	83	96.52
349	B 9 6 B B	- - - - - - 8 2 - - - B 8 2 8 6 D E 2 B 9 3 8 2 - - - B 8 2 - - -	147	62		43.57	76	87	102.60
350	B 9 6 F F	- - - - - - - - - - - B 8 2 - 4 D E 2 F 9 3 8 2 - - - F 8 2 - - -	148	52		42.75	67	81	92.48
351	B 9 A 1 B	- - - - - - C 6 4 - - A 1 1 C 2 9 2 2 - - - C 6 4 - - A 1 1 - - -	351	58	98.52	43.21	73	83	98.54
352	B 9 A 3 B	- - - - - - C 2 4 - - A - 1 C 6 9 2 2 2 1 - C 2 4 - - A - 1 - - -	352	58	98.52	43.13	72	82	98.54
353	B 9 A 7 F	- - - - - - 4 - 4 - - A - 1 4 4 9 2 2 6 1 - C 2 4 - - E - 1 - - -	353	52	92.43	43.91	67	80	92.46
354	B 9 A B B	- - - - - - C 2 - - - A - - C 6 D 2 2 A 1 1 C 2 - - - A - - - - -	354	50	90.39	40.03	64	76	90.41
355	B 9 A F F	- - - - - - 4 - - - - A - - 4 4 D 2 2 E 1 1 C 2 - - - E - - - - -	355	44	84.29	40.78	59	73	84.34
356	B 9 E 1 B	- - - - - - C 6 4 - - A 9 3 C 2 9 6 2 - 8 2 C 6 4 - - A 9 3 - - -	356	64	104.61	45.21	79	89	104.63
357	B 9 E 3 B	- - - - - - C 2 4 - - A 8 3 C 6 9 6 2 2 9 2 C 2 4 - - A 8 3 - - -	159	66		46.80	79	88	106.66
358	B 9 E 7 F	- - - - - - 4 - 4 - - A 8 3 4 4 9 6 2 6 9 2 C 2 4 - - E 8 3 - - -	160	60		47.49	74	86	100.57
359	B 9 E B B	- - - - - - C 2 - - - A 8 2 C 6 D 6 2 A 9 3 C 2 - - - A 8 2 - - -	157	62		43.57	75	86	102.60

Fig. 19: Walsh spectrum decomposition and attack complexity (3)

id	mask of X1	mask of Y0	id'	d1	comp.1	# mul	d2	d3	comp.2
360	B 9 E F F	- - - - - 4 - - - - A 8 2 4 4 D 6 2 E 9 3 C 2 - - - E 8 2 - - -	158	56		44.74	70	84	96.53
361	E 4 C 8 A	- - - - - C 6 - - - 7 B 2 C 2 - C - 1 A 3 C 6 - - - 3 B 2 - - -	361	56	96.49	42.22	72	83	96.52
362	E 4 C 2 A	- - - - - C 2 4 - - 7 A 3 C 6 4 C - B B 2 C 2 4 - - 3 A 3 - - -	362	62	102.59	47.09	77	87	102.60
363	E 4 C 6 E	- - - - - 4 - 4 - - 7 A 3 4 4 4 C - F B 2 C 2 4 - - 7 A 3 - - -	363	56	96.49	48.14	72	84	96.53
364	E 4 C A A	- - - - - C 2 - - - 7 A 2 C 6 - C - 3 B 3 C 2 - - - 3 A 2 - - -	361	56		42.22	72	83	96.52
365	E 4 C E E	- - - - - 4 - - - - 7 A 2 4 4 - C - 7 B 3 C 2 - - - 7 A 2 - - -	365	50	90.39	43.72	67	80	90.49
366	E 4 1 8 A	- - - - - 8 E 2 - - 6 3 - 8 A 2 1 - - 2 1 8 E 2 - - 2 3 - - - -	366	58	98.52	40.49	74	85	98.55
367	E 4 1 2 A	- - - - - 8 A 6 - - 6 2 1 8 E 6 1 - A 3 - 8 A 6 - - 2 2 1 - - -	367	66	106.64	40.73	80	90	106.66
368	E 4 1 6 E	- - - - - - 8 6 - - 6 2 1 - C 6 1 - E 3 - 8 A 6 - - 6 2 1 - - -	368	58	98.52	40.45	74	85	98.55
369	E 4 1 A A	- - - - - 8 A 2 - - 6 2 - 8 E 2 1 - 2 3 1 8 A 2 - - 2 2 - - - -	369	62	102.59	40.55	76	86	102.60
370	E 4 1 E E	- - - - - - 8 2 - - 6 2 - - C 2 1 - 6 3 1 8 A 2 - - 6 2 - - - -	370	54	94.46	40.22	70	81	94.49
371	E 4 5 8 A	- - - - - 8 E 2 - - 6 B 2 8 A 2 5 - - A 3 8 E 2 - - 2 B 2 - - -	166	66		40.73	81	92	106.66
372	E 4 5 2 A	- - - - - 8 A 6 - - 6 A 3 8 E 6 5 - A B 2 8 A 6 - - 2 A 3 - - -	169	70		40.78	83	93	110.71
373	E 4 5 6 E	- - - - - - 8 6 - - 6 A 3 - C 6 5 - E B 2 8 A 6 - - 6 A 3 - - -	170	62		40.51	77	88	102.61
374	E 4 5 A A	- - - - - 8 A 2 - - 6 A 2 8 E 2 5 - 2 B 3 8 A 2 - - 2 A 2 - - -	169	70		40.78	83	93	110.71
375	E 4 5 E E	- - - - - - 8 2 - - 6 A 2 - C 2 5 - 6 B 3 8 A 2 - - 6 A 2 - - -	170	62		40.51	77	88	102.61
376	E 4 9 8 A	- - - - - C E 2 - - 7 3 - C A 2 9 - 1 2 1 C E 2 - - 3 3 - - - -	48	62		40.55	75	84	102.60
377	E 4 9 2 A	- - - - - C A 6 - - 7 2 1 C E 6 9 - B 3 - C A 6 - - 3 2 1 - - -	47	66		40.73	79	88	106.66
378	E 4 9 6 E	- - - - - 4 8 6 - - 7 2 1 4 C 6 9 - F 3 - C A 6 - - 7 2 1 - - -	378	62	102.59	42.50	77	88	102.61
379	E 4 9 A A	- - - - - C A 2 - - 7 2 - C E 2 9 - 3 3 1 C A 2 - - 3 2 - - - -	48	62		40.55	75	84	102.60
380	E 4 9 E E	- - - - - 4 8 2 - - 7 2 - 4 C 2 9 - 7 3 1 C A 2 - - 7 2 - - - -	380	58	98.52	42.25	73	84	98.54
381	E 4 D 8 A	- - - - - C E 2 - - 7 B 2 C A 2 D - 1 A 3 C E 2 - - 3 B 2 - - -	176	70		40.78	82	91	110.71
382	E 4 D 2 A	- - - - - C A 6 - - 7 A 3 C E 6 D - B B 2 C A 6 - - 3 A 3 - - -	176	70		40.78	82	91	110.71
383	E 4 D 6 E	- - - - - 4 8 6 - - 7 A 3 4 C 6 D - F B 2 C A 6 - - 7 A 3 - - -	180	66		42.51	80	91	106.66
384	E 4 D A A	- - - - - C A 2 - - 7 A 2 C E 2 D - 3 B 3 C A 2 - - 3 A 2 - - -	176	70		40.78	82	91	110.71
385	E 4 D E E	- - - - - 4 8 2 - - 7 A 2 4 C 2 D - 7 B 3 C A 2 - - 7 A 2 - - -	180	66		42.51	80	91	106.66
386	E 9 B 8 A	- - - - - 4 C 2 - - F 1 - 4 8 F B - 9 - 1 4 C 2 - - B 1 - - - -	386	52	92.43	44.21	68	80	92.46
387	E 9 B 2 A	- - - - - 4 8 6 - - F - 1 4 C B B - 3 1 - 4 8 6 - - B - 1 - - -	387	52	92.43	44.21	68	81	92.48
388	E 9 B 6 E	- - - - - C A 6 - - F - 1 C E B B - 7 1 - 4 8 6 - - F - 1 - - -	388	56	96.49	42.44	72	83	96.52
389	E 9 B A A	- - - - - 4 8 2 - - F - - 4 C F B - B 1 1 4 8 2 - - B - - - -	389	48	88.36	43.90	64	77	88.42
390	E 9 B E E	- - - - - C A 2 - - F - - C E F B - F B 2 C A 6 - - 7 A 3 - - -	390	52	92.43	42.23	68	79	92.45
391	E 9 2 8 A	- - - - - - 4 - - - E 1 - - - D 2 - 8 - 1 - 4 - - - A 1 - - - -	391	34	74.09	40.19	52	66	74.42
392	E 9 2 2 A	- - - - - - - 4 - - E - 1 - 4 9 2 - 2 1 - - - 4 - - A - 1 - - -	392	38	78.17	39.75	55	68	78.27
393	E 9 2 6 E	- - - - - 8 2 4 - - E - 1 8 6 9 2 - 6 1 - - - 4 - - E - 1 - - -	393	48	88.36	41.10	65	78	88.46
394	E 9 2 A A	- - - - - - - - - - E - - - 4 D 2 - A 1 1 - - - - - A - - - - -	394	30	70.00	37.31	47	61	70.18
395	E 9 2 E E	- - - - - 8 2 - - - E - - - 8 6 D 2 - E 1 1 - - - - - E - - - - -	395	40	80.21	37.98	57	71	80.39
396	E 9 6 8 A	- - - - - - 4 - - - E 9 2 - - D 6 - 8 8 3 - 4 - - - A 9 2 - - -	396	44	84.29	42.39	62	76	84.61
397	E 9 6 2 A	- - - - - - - - 4 - - E 8 3 - - 4 9 6 - 2 9 2 - - 4 - - A 8 3 - - -	199	46		42.92	62	75	86.38
398	E 9 6 6 E	- - - - - - 8 2 4 - - E 8 3 8 6 9 6 - 6 9 2 C - 4 - - F 8 3 - - -	200	56		44.79	72	85	96.55
399	E 9 6 A A	- - - - - - - - - - - E 8 2 - 4 D 6 - A 9 3 - - - - - A 8 2 - - -	197	42		40.73	58	72	82.35
400	E 9 6 E E	- - - - - 8 2 - - - E 8 2 8 6 D 6 - E 9 3 - - - - - E 8 2 - - -	198	52		41.57	68	82	92.53
401	E 9 A 8 A	- - - - - 4 4 - - - F 1 - 4 - D A - 9 - 1 4 4 - - - B 1 - - - -	401	42	82.25	42.14	58	72	82.35
402	E 9 A 2 A	- - - - - 4 - 4 - - F - 1 4 4 9 A - 3 1 - 4 - 4 - - B - 1 - - -	402	44	84.29	43.88	60	73	84.34
403	E 9 A 6 E	- - - - - C 2 4 - - F - 1 C 6 9 A - 7 1 - 4 - 4 - - F - 1 - - -	403	50	90.39	43.08	66	79	90.45
404	E 9 A A A	- - - - - 4 - - - - F - - - 4 4 D A - B 1 1 4 - - - - B - - - - -	404	36	76.13	40.72	52	66	76.23
405	E 9 A E E	- - - - - C 2 - - - F - - C 6 D A - F 1 1 4 - - - - F - - - - -	405	42	82.25	39.92	58	72	82.35
406	E 9 E 8 A	- - - - - 4 4 - - - F 9 2 4 - D E - 9 8 3 4 4 - - - B 9 2 - - -	406	52	92.43	44.37	67	81	92.48
407	E 9 E 2 A	- - - - - 4 - 4 - - F 8 3 4 4 9 E - 3 9 2 4 - 4 - - B 8 3 - - -	209	52		47.48	67	80	92.46
408	E 9 E 6 E	- - - - - C 2 4 - - F 8 3 C 6 9 E - 7 9 2 4 - 4 - - F 8 3 - - -	210	58		46.79	73	86	98.56
409	E 9 E A A	- - - - - 4 - - - - F 8 2 4 4 D E - B 9 3 4 - - - - B 8 2 - - -	207	48		44.73	63	77	88.42
410	E 9 E E E	- - - - - C 2 - - - F 8 2 C 6 D E - F 9 3 4 - - - - F 8 2 - - -	208	54		43.54	69	83	94.52
411	F 4 C 1 B	- - - - - C 6 4 - - 6 B 3 C 2 4 4 2 8 A 2 C 6 4 - - 2 B 3 - - -	411	64	104.61	49.08	80	90	104.63
412	F 4 C 3 B	- - - - - C 2 4 - - 6 A 3 C 6 4 4 2 A B 2 C 2 4 - - 2 A 3 - - -	412	66	106.64	47.09	80	90	106.66
413	F 4 C 7 F	- - - - - 4 - 4 - - 6 A 3 4 4 4 4 2 E B 2 C 2 4 - - 6 A 3 - - -	413	60	100.55	48.14	76	88	100.59
414	F 4 C B B	- - - - - C 2 - - - 6 A 2 C 6 - 4 2 2 B 3 C 2 - - - 2 A 2 - - -	414	60	100.55	42.24	75	86	100.57
415	F 4 C F F	- - - - - 4 - - - - 6 A 2 4 4 - 4 2 6 B 3 C 2 - - - 6 A 2 - - -	415	54	94.46	43.73	71	84	94.56
416	F 4 1 1 B	- - - - - 8 E 6 - - 7 3 1 8 A 6 9 2 9 2 - 8 E 6 - - 3 3 1 - - -	416	70	110.70	40.78	82	91	110.71
417	F 4 1 3 B	- - - - - 8 A 6 - - 7 2 1 8 E 6 9 2 B 3 - 8 A 6 - - 3 2 1 - - -	416	70		40.78	82	91	110.71
418	F 4 1 7 F	- - - - - - 8 6 - - 7 2 1 - - C 6 9 2 F 3 - 8 A 6 - - 7 2 1 - - -	418	62	102.59	40.51	76	86	102.60
419	F 4 1 B B	- - - - - 8 A 2 - - 7 2 - 8 E 2 9 2 3 3 1 8 A 2 - - 3 2 - - - -	419	66	106.64	40.61	78	88	106.66
420	F 4 1 F F	- - - - - - 8 2 - - 7 2 - - C 2 9 2 7 3 1 8 A 2 - - 7 2 - - - -	420	58	98.52	40.30	72	83	98.54
421	F 4 5 1 B	- - - - - 8 E 6 - - 7 B 3 8 A 6 D 2 9 A 2 8 E 6 - - 3 B 3 - - -	6	74		40.83	85	94	114.77
422	F 4 5 3 B	- - - - - 8 A 6 - - 7 A 3 8 E 6 D 2 B B 2 8 A 6 - - 3 A 3 - - -	6	74		40.83	85	94	114.77
423	F 4 5 7 F	- - - - - - 8 6 - - 7 A 3 - C 6 D 2 F B 2 8 A 6 - - 7 A 3 - - -	220	66		40.57	79	89	106.66
424	F 4 5 B B	- - - - - 8 A 2 - - 7 A 2 8 E 2 D 2 3 B 3 8 A 2 - - 3 A 2 - - -	6	74		40.83	85	94	114.77
425	F 4 5 F F	- - - - - - 8 2 - - 7 A 2 - C 2 D 2 7 B 3 8 A 2 - - 7 A 2 - - -	220	66		40.57	79	89	106.66
426	F 4 9 1 B	- - - - - C E 6 - - 6 3 1 C A 6 1 2 8 2 - C E 6 - - 2 3 1 - - -	426	68	108.67	42.64	81	89	108.69
427	F 4 9 3 B	- - - - - C A 6 - - 6 2 1 C E 6 1 2 A 3 - C A 6 - - 2 2 1 - - -	427	70	110.70	40.78	81	89	110.71
428	F 4 9 7 F	- - - - - 4 8 6 - - 6 2 1 4 C 6 1 2 E 3 - C A 6 - - 6 2 1 - - -	428	66	106.64	42.51	79	89	106.66
429	F 4 9 B B	- - - - - C A 2 - - 6 2 - C E 2 1 2 3 1 C A 2 - - 2 2 - - - -	429	66	106.64	40.61	77	86	106.66
430	F 4 9 F F	- - - - - 4 8 2 - - 6 2 - 4 C 2 1 2 6 3 1 C A 2 - - 6 2 - - - -	430	62	102.59	42.26	75	86	102.60
431	F 4 D 1 B	- - - - - C E 6 - - 6 B 3 C A 6 5 2 8 A 2 C E 6 - - 2 B 3 - - -	231	72		42.65	84	92	112.74
432	F 4 D 3 B	- - - - - C A 6 - - 6 A 3 C E 6 5 2 A B 2 C A 6 - - 2 A 3 - - -	229	74		40.83	84	92	114.77
433	F 4 D 7 F	- - - - - 4 8 6 - - 6 A 3 4 C 6 5 2 E B 2 C A 6 - - 6 A 3 - - -	230	70		42.53	82	92	110.71
434	F 4 D B B	- - - - - C A 2 - - 6 A 2 C E 2 5 2 2 B 3 C A 2 - - 2 A 2 - - -	229	74		40.83	84	92	114.77
435	F 4 D F F	- - - - - 4 8 2 - - 6 A 2 4 C 2 5 2 6 B 3 C A 2 - - 6 A 2 - - -	230	70		42.53	82	92	110.71
436	F 9 B 1 B	- - - - - 4 C 6 - - E 1 1 4 8 B 3 2 - - - 4 C 6 - - A 1 1 - - -	436	56	96.49	44.47	72	83	96.52
437	F 9 B 3 B	- - - - - 4 8 6 - - E - 1 4 C B 3 2 2 1 - 4 8 6 - - A - 1 - - -	437	56	96.49	44.23	71	82	96.51
438	F 9 B 7 F	- - - - - C A 6 - - E - 1 C E B 3 2 6 1 - 4 8 6 - - E - 1 - - -	438	60	100.55	42.50	75	84	100.57
439	F 9 B B B	- - - - - 4 8 2 - - E - - 4 C F 3 2 A 1 1 4 8 2 - - A - - - - -	439	52	92.43	43.93	67	79	92.45
440	F 9 B F F	- - - - - C A 2 - - E - - C E F 3 2 E 1 1 4 8 2 - - E - - - - -	440	56	96.49	42.31	71	81	96.51
441	F 9 2 1 B	- - - - - - 4 4 - - F 1 1 - - 9 A 2 1 - - - 4 4 - - B 1 1 - - -	441	48	88.36	41.07	64	76	88.40
442	F 9 2 3 B	- - - - - - - - 4 - - F - 1 - 4 9 A 2 3 1 - - - 4 - - B - 1 - - -	442	42	82.25	39.85	58	71	82.31
443	F 9 2 7 F	- - - - - 8 2 4 - - F - 1 8 6 9 A 2 7 1 - - - 4 - - F - 1 - - -	443	52	92.43	41.14	68	81	92.48
444	F 9 2 B B	- - - - - - - - - - - F - - - 4 D A 2 B 1 1 - - - - - B - - - - -	444	34	74.09	37.44	50	64	74.19
445	F 9 2 F F	- - - - - 8 2 - - - F - - 8 6 D A 2 F 1 1 - - - - - F - - - - -	445	44	84.29	38.07	60	74	84.39
446	F 9 6 1 B	- - - - - - 4 4 - - F 9 3 - - 9 E 2 1 8 2 - 4 4 - - B 9 3 - - -	446	54	94.46	43.09	69	81	94.49
447	F 9 6 3 B	- - - - - - - 4 - - F 8 3 - 4 9 E 2 3 9 2 - - 4 - - B 8 3 - - -	249	50		42.96	65	77	90.42
448	F 9 6 7 F	- - - - - 8 2 4 - - F 8 3 8 6 9 E 2 7 9 2 - - 4 - - F 8 3 - - -	250	60		44.80	75	87	100.58
449	F 9 6 B B	- - - - - - - - - - F 8 2 - 4 D E 2 B 9 3 - - - - - B 8 2 - - -	247	46		40.78	61	75	86.38
450	F 9 6 F F	- - - - - 8 2 - - - F 8 2 8 6 D E 2 F 9 3 - - - - - F 8 2 - - -	248	56		41.59	71	85	96.55
451	F 9 A 1 B	- - - - - 4 4 4 - - E 1 1 4 - 9 2 2 - - - 4 4 4 - - A 1 1 - - -	451	50	90.39	45.28	66	78	90.43
452	F 9 A 3 B	- - - - - 4 - 4 - - E - 1 4 4 9 2 2 2 1 - 4 - 4 - - A - 1 - - -	452	48	88.36	43.90	63	76	88.40
453	F 9 A 7 F	- - - - - C 2 4 - - E - 1 C 6 9 2 2 6 1 - 4 - 4 - - E - 1 - - -	453	54	94.46	43.12	69	82	94.50
454	F 9 A B B	- - - - - 4 - - - - E - - 4 4 D 2 2 A 1 1 4 - - - - - A - - - - -	454	40	80.21	40.77	55	69	80.27
455	F 9 A F F	- - - - - C 2 - - - E - - - C 6 D 2 2 E 1 1 4 - - - - - E - - - - -	455	46	86.32	40.01	61	75	86.38
456	F 9 E 1 B	- - - - - 4 4 4 - - E 9 3 4 - 9 6 2 - 8 2 4 4 4 - - A 9 3 - - -	456	56	96.49	47.28	72	84	96.53
457	F 9 E 3 B	- - - - - 4 - 4 - - E 8 3 4 4 9 6 2 2 9 2 4 - 4 - - A 8 3 - - -	259	56		47.49	70	82	96.51
458	F 9 E 7 F	- - - - - C 2 4 - - E 8 3 C 6 9 6 2 6 9 2 4 - 4 - - E 8 3 - - -	260	62		46.80	76	88	102.61
459	F 9 E B B	- - - - - 4 - - - - E 8 2 4 4 D 6 2 A 9 3 4 - - - - A 8 2 - - -	257	52		44.74	66	80	92.46
460	F 9 E F F	- - - - - C 2 - - - E 8 2 C 6 D 6 2 E 9 3 4 - - - - E 8 2 - - -	258	58		43.57	72	86	98.56

Fig. 20: Walsh spectrum decomposition and attack complexity (4)