Improved Linear Cryptanalysis of reduced-round SIMON-32 and SIMON-48

Mohamed Ahmed Abdelraheem[1] ⋆, Javad Alizadeh[2]⋆⋆, Hoda A. Alkhzaimi[3], Mohammad Reza Aref[2], Nasour Bagheri[4], and Praveen Gauravaram[5] ⋆⋆⋆

[1] SICS Swedish ICT, Sweden, mohamed.abdelraheem@sics.se
[2] ISSL, E.E. Department, Sharif University of Technology, Iran, alizadja@gmail.com
[3] Section for Cryptology, DTU Compute, Technical University of Denmark, Denmark, hoalk@dtu.dk
[4] E.E. Department of Shahid Rajaee Teachers Training University and the School of Computer Science of Institute for Research in Fundamental Sciences (IPM), Iran, NBagheri@srttu.edu
[5] Queensland University of Technology, Brisbane, Australia, praveen.gauravaram@qut.edu.au

Abstract. In this paper we analyse two variants of SIMON family of light-weight block ciphers against linear cryptanalysis and present the best linear cryptanalytic results on these variants of reduced-round SIMON to date.

We propose a time-memory trade-off method that finds differential/linear trails for any permutation allowing low Hamming weight differential/linear trails. Our method combines low Hamming weight trails found by the correlation matrix representing the target permutation with heavy Hamming weight trails found using a Mixed Integer Programming model representing the target differential/linear trail. Our method enables us to find a 17-round linear approximation for SIMON-48 which is the best current linear approximation for SIMON-48. Using only the correlation matrix method, we are able to find a 14-round linear approximation for SIMON-32 which is also the current best linear approximation for SIMON-32.

The presented linear approximations allow us to mount a 23-round key recovery attack on SIMON-32 and a 24-round Key recovery attack on SIMON-48/96 which are the current best results on SIMON-32 and SIMON-48. In addition we have an attack on 24 rounds of SIMON-32 with marginal complexity.

Keywords: SIMON, linear cryptanalysis, linear hull, correlation matrix, Mixed Integer Programming (MIP)

1 Introduction

Over the past few years, the necessity for limited cryptographic capabilities in resource-constraint computing devices such as RFID tags has led to the design of several lightweight cryptosystems [8, 12, 13, 15, 17, 18, 19, 30]. In this direction, Beaulieu *et al.* of the U.S. National Security Agency (NSA) designed SIMON family of lightweight block ciphers that are targeted towards optimal hardware performance [9]. Meeting hardware requirements of low-power and limited gate devices is the main design criteria of SIMON.

SIMON has plaintext block sizes of 32, 48, 64, 96 and 128 bits, each with up to three key sizes. SIMON-N/K denotes a variant of SIMON with block and key sizes of N and K bits respectively. With the proposed block and key lengths, SIMON is a family of ten lightweight block ciphers. Since the publication of SIMON, each cipher in this family has undergone reduced round cryptanalysis against linear [2, 3, 4, 5, 6, 24], differential [3, 4, 11, 28], impossible differential [14], rectangular [3, 4] and integral [29] attacks.

⋆ This work was done while the author was a postdoc at the Technical University of Denmark

⋆⋆ Javad Alizadeh, Mohammad Reza Aref and Nasour Bagheri were partially supported by Iran-NSF under grant no. 92.32575.

⋆⋆⋆ Praveen Gauravaram is supported by Australian Research Council Discovery Project grant number DP130104304.

Contributions. In this paper, we analyse the security of SIMON-32 and SIMON-48. First we analyze the security of reduced-round SIMON-32 and SIMON-48 against several variants of linear cryptanalysis and report the best results to date with respect to any form of cryptanalysis in terms of the number of rounds attacked on SIMON-32/64 and 48/96. Our attacks are described below and results are summarised in Table 1.

- We propose a time-memory trade-off method that combines low Hamming weight trails found by the correlation matrix (consumes huge memory) with heavy Hamming weight trails found by the Mixed Integer Programming (MIP) method [26] (consumes time depending on the specified number of trails to be found). The method enables us to find a 17-round linear approximation for SIMON-48 which is the best current approximation.
- We found a 14-round linear hull approximation for SIMON-32 using a squared correlation matrix with input/output masks of Hamming weight ≤ 9.
- Using our approximations, we are able to break 23 and 24 rounds of SIMON-32, 23 rounds of SIMON-48/72 and 24 rounds of SIMON-48/96 with a marginal time complexity $2^{63.9}$.

Previous results on SIMON used in our paper. The work in [20] provides an explicit formula for computing the probability of a 1-round differential characteristic of the SIMON's non-linear function. It also provides an efficient algorithm for computing the squared correlation of a 1-round linear characteristic of the SIMON nonlinear function which we used in our linear cryptanalysis to SIMON-48.

The work in [24] defines a MIP linear model that finds linear trails for SIMON. The solution of the MIP model sometimes yield a false linear trail but most of the time it yields a valid linear trail. When a solution is found whether valid or invalid, we add a new constraint to the MIP model that prevents the current solution from occurring in the next iteration.

Related work on SIMON. The most improved results in terms of the number of rounds attacked, data and time complexity presented, up-to-date of this publication, are in the scope of differential, linear and integral attacks as reflected in Table 1. Focusing on the different cryptanalysis results of SIMON-32, SIMON-48/72 and SIMON-48/96, Abed *et al.* [3, 4] have presented that classical differential results yield attacks on 18 for the smallest variant and 19 rounds for SIMON-48 with data and time stated in Table 1. This was improved to 21 rounds for SIMON-32 and $22-24$ rounds for SIMON-48/72 and SIMON-48/96 by Wang *et al.* [27, 28] using dynamic key guessing and automatic enumeration of differential characteristics through imposing conditions on differential paths to reduce the intended key space searched.

Independent to our work, Ashur [7] described a method for finding linear trails that work only against SIMON-like ciphers. This method finds a multivariate polynomial in GF(2) representing the r-round linear approximation under consideration. Each solution of the multivariate polynomial corresponds to a valid trail that is part of the many linear trails that forms the linear approximation. This suggests that the probability that the r-round linear approximation is satisfied is equivalent to the number of solutions for its corresponding multivariate polynomial divided by the size of the solution space. For $r = 2$, the authors mentioned that the space size is 2^{10}. For more rounds the space gets bigger as many bits will be involved in the corresponding multivariate polynomial. Finding the number of solutions of a multivariate polynomial is a hard problem. To overcome this, the author uses the above method to form what is called a "linear super-trail" which glues two short linear hulls (a short linear hull has a small number of rounds that make it is feasible to find the number of solutions of the corresponding multivariate polynomial) in order to form a super-trail.

In contrast, our time-memory trade-off method which basically combines two different linear trails found using a squared correlation matrix (trails with light Hamming weight) and a mixed

integer programming model (trails with heavy Hamming weight) is not SIMON specific, it is very generic and can be used for any permutation allowing low Hamming weight linear/differential trails to find linear/differential trails. As described in Section 5.3, we have better attacks on both SIMON-32 (using squared correlation matrix) and SIMON-48 (using time-memory trade-off) compared to the results of [7].

Kölbl *et al.* [20] used SAT/SMT solvers to find optimal differential and linear characteristics. They also found the best 14-round differential approximation $0x00000008 \xrightarrow{14-round} 0x08000000$ for SIMON-32 with probability $2^{-30.81}$ but they do not provide any key recovery attacks. They also provided a 13-round differential approximation $0x00000040 \xrightarrow{13-round} 0x40000000$ with differential probability $2^{-28.79}$ contributed from $\approx 2^{25.21}$ differential trails. Using SAT/SMT solvers, they enumerated all the differential trails for the 13-round differential approximation within one month. However, our computations for the 14-round linear approximations shown in Table 2 took only few hours to build the squared correlation matrix and very few minutes to estimate their squared correlations.

Our 14-round linear approximations have squared correlations $2^{-30.58}$ contributed from $\approx 2^{28}$ using a squared correlation matrix with Hamming weight ≤ 9 which are better than the 14-round differential approximation with probability $2^{-30.81}$ presented in [20]. It is difficult to compare differential and linear approximations in SIMON, though they look very similar. But one explanation to why our linear approximations are better could be because our matrix method allows us to estimate the squared correlations for many approximations and thus choose which is best faster than the method presented in [20]. We also noticed that when limiting the Hamming weight of input/output differences/masks, the correlation matrix of SIMON-32 has more non zero elements compared to the difference matrix of SIMON-32. For example correlation matrix of SIMON32 with Hamming weight ≤ 8 has $\approx 2^{26.99}$ non zero elements which is more than the $2^{26.77}$ nonzero elements of the difference matrix of SIMON-32 with Hamming weight ≤ 8 (for ≤ 3, both matrices have the same number of nonzero elements). This might indicate that linear approximations might be better than the differential ones at least when using the matrix method with limited Hamming weights.

However, for large block sizes of SIMON, the approach used in [20] outperforms the matrix method and this is due to the fact that the matrix method is a greedy method that performs very well for small block sizes such as SIMON-32 but due to its large memory consumption it does not yield better results for large block sizes of SIMON[1]. To benefit from the greedy matrix method, we combine the matrix method with the MIP method in order to find better linear approximations for SIMON-48.

Organization. The paper is structured as follows. In Section 2 we describe SIMON. In Section 3 concepts and notation required for linear cryptanalysis of SIMON are presented. In Section 4 the used Time-Memory Trade-off method is described. In Section 5 we used squared correlation matrix to establish a linear hull of SIMON and investigate the data and time complexity for the smallest variant of SIMON. We conclude the paper in Section 6.

[1] For SIMON-64 the matrix method with Hamming weight ≤ 6 does not perform very well compared to PRESENT (which has block size 64-bit) where the best linear approximations were found using the matrix method with Hamming weight ≤ 4 [1]. This is due to the fact that trails with very low Hamming weight perform very well in PRESENT compared to SIMON-64

Table 1. State-of-the-art cryptanalysis of SIMON-(32/64, 48/72, 48/96)

SIMON		Diff.					Imp.Diff.	Z-Corr.	Integ.	Multi.Lin.	Lin.			Lin. Hull		
		[4]	[11]	[28]	[27]	[25]	[14]	[29]	[29]	[7]	[3]	[5]	[2]	[24]	[25]	**This work**
32/64	*#rounds*	18	19	21	21	−−	19	20	21	24	11	13	17	21	−−	**23**
	Time	$2^{46.0}$	$2^{32.0}$	$2^{46.0}$	$2^{55.25}$	−−	$2^{62.56}$	$2^{56.96}$	$2^{63.0}$	$2^{63.57}$	−−	−−	$2^{52.5}$	−−	−−	$\mathbf{2^{50}}$
	Data	$2^{31.2}$	$2^{31.0}$	$2^{31.0}$	$2^{31.0}$	−−	$2^{32.0}$	$2^{32.0}$	$2^{31.0}$	$2^{31.57}$	$2^{23.0}$	$2^{32.0}$	$2^{32.0}$	$2^{30.19}$	−−	$\mathbf{2^{30.59}}$
48/72	*#rounds*	19	20	22	23	16	20	20	−−	23	14	16	19	−−	−−	**23**
	Time	$2^{52.0}$	$2^{52.0}$	$2^{63.0}$	$2^{63.25}$	−−	$2^{70.69}$	$2^{59.7}$	−−	$2^{68.4}$	−−	−−	2^{70}	−−	−−	$\mathbf{2^{62.10}}$
	Data	$2^{46.0}$	$2^{46.0}$	$2^{45.0}$	2^{47}	$2^{44.65}$	2^{48}	2^{48}	−−	$2^{44.4}$	$2^{47.0}$	$2^{46.0}$	$2^{46.0}$	−−	−−	$\mathbf{2^{47.78}}$
48/96	*#rounds*	19	20	22	24	16	21	21	−−	24	14	16	20	21	23	**24**
	Time	$2^{76.0}$	$2^{75.0}$	$2^{71.0}$	$2^{87.25}$	−−	$2^{94.73}$	$2^{72.63}$	−−	$2^{92.4}$	−−	−−	$2^{86.5}$	−−	−−	$\mathbf{2^{83.10}}$
	Data	$2^{46.0}$	$2^{46.0}$	$2^{45.0}$	2^{47}	$2^{44.65}$	$2^{38.0}$	$2^{48.0}$	−−	$2^{44.4}$	$2^{47.0}$	$2^{46.0}$	$2^{46.0}$	$2^{42.28}$	$2^{44.92}$	$\mathbf{2^{47.78}}$

2 Description of SIMON

SIMON has a classical Feistel structure with the round block size of $N = 2n$ bits where n is the word size representing the left or right branch of the Feistel scheme at each round. The number of rounds is denoted by r and depends on the variant of SIMON.

We denote the right and left halves of plaintext P and ciphertext C by (P_R, P_L) and (C_R, C_L) respectively. The output of round r is denoted by $X^r = X_L^r \| X_R^r$ and the subkey used in a round r is denoted by K^r. Given a string X, $(X)_i$ denotes the i^{th} bit of X. Bitwise circular left-rotation of string a by b positions to the left is denoted by $a \lll b$. Further, $\oplus$ and & denote bitwise XOR and AND operations respectively.

Each round of SIMON applies a non-linear, non-bijective (and hence non-invertible) function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ to the left half of the state. The output of F is added using XOR to the right half along with a round key followed by swapping of two halves. The function F is defined as

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

The subkeys are derived from a master key. Depending on the size K of the master key, the key schedule of SIMON operates on two, three or four n-bit word registers. We refer to [9] for the detailed description of SIMON structure and key scheduling.

3 Preliminaries

Correlation Matrix. Linear cryptanalysis finds a linear relation between some plaintext bits, ciphertext bits and some secret key bits and then exploits the bias or correlation of this linear relation. In other words, the adversary finds an input mask α and an output mask β which yields a higher absolute *bias* $\epsilon_F(\alpha, \beta) \in [-\frac{1}{2}, \frac{1}{2}]$. In other words

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

deviates from $\frac{1}{2}$ where $\langle \cdot, \cdot \rangle$ denotes an inner product. Let $a = (a_1, \ldots, a_n), b = (b_1, \ldots, b_n) \in \mathbb{F}_2^n$. Then

$$a \cdot b \triangleq a_1 b_1 \oplus \cdots \oplus a_n b_n$$

denotes the *inner product* of a and b. The correlation of a linear approximation is defined as

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

Another definition of the correlation which we will use later is

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

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

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

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

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

Given a composite function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that $F = F_r \circ \cdots \circ F_2 \circ F_1$,, we estimate the correlation of an r-round linear approximation (α_0, α_r) by considering the correlation of each linear characteristic between α_0 and α_r. The correlation of i^{th} linear characteristic $(\alpha_0 = \alpha_{0i}, \alpha_{1i}, \cdots, \alpha_{(r-1)i}, \alpha_r = \alpha_{ri})$ is

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

It is well known [16] that the correlation of a linear approximation is the sum of all correlations of linear trails starting with the same input mask α and ending with the same output mask β, i.e. $C_F(\alpha_0, \alpha_r) = \sum_{i=1}^{N_l} C_i$ where N_l is the number of all possible linear characteristics between (α_0, α_r).

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

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

where $d_i \in \mathbb{F}_2$ refers to the sign of the addition of the subkey bits on the i^{th} linear trail. In order to estimate the data complexity of a linear attack, one uses the average squared correlation over all the keys which is equivalent to the sum of the squares of the correlations of all trails, $\sum_i C_i^2$, assuming independent round keys [16].

Let C denotes the correlation matrix of an n-bit key-alternating cipher. C has size $2^n \times 2^n$ and $C_{i,j}$ corresponds to the correlation of an input mask, say α_i, and output mask, say β_j. Now the correlation matrix for the keyed round function is obtained by changing the signs of each row in C according to the round subkey bits or the round constant bits involved. Squaring each entry of the correlation matrix gives us the squared correlation matrix M. Computing M^r gives us the squared correlations after r number of rounds. This can not be used for real block ciphers that have block sizes of at least 32 bits as in the case of SIMON-32/64. Therefore, in order to find linear approximations one can construct a submatrix of the correlation (or the squared correlation) matrix [1, 12]. In Section 5, we construct a squared correlation submatrix for SIMON in order to find good linear approximations.

3.1 Mixed Integer Programming Method (MIP)

Mouha *et al.*'s [21] presented a mixed integer programming model that minimizes the number of active Sboxes involved in a linear or differential trail. Their work was mainly on byte oriented ciphers. Later, Mouha's framework was extended to accommodate bit oriented ciphers. More

recently, at Asiacrypt 2014 [26], the authors described a method for constructing a model that finds the actual linear/differential trail with the specified number of active Sboxes. Of course, there would be many solutions but whenever a solution is found the MIP model is updated by adding a new constraint that discards the current found solution from occurring in the next iteration for finding another solution.

For every input/ouput bit mask or bit difference at some round state, a new binary variable x_i is introduced such that $x_i = 1$ iff the corresponding bit mask or bit difference is non-zero. For every Sbox at each round, a new binary variable a_j is introduced such that $a_j = 1$ if the input mask or difference of the corresponding Sbox is nonzero. Thus, a_j indicates the activity of an Sbox. Now, the natural choice of the *objective function* f of our MIP model is to minimize the number of active Sboxes, i.e., $f = \sum_j a_j$. If our goal from the above integer programming model is to only find the minimum number of active Sboxes existing in a differential/linear trial of a given bit-oriented cipher, then we are only concerned about the binary values which represent the activity of the Sboxes involved in the differential/linear trail a_v. Thus, in order to speed up solving the model, one might consider restricting the activity variables and the dummy variables to be binary and allow the other variables to be any real numbers. This will turn the integer programming model into a Mixed Integer Programming model which is easier to solve than an Integer programming model. However, since we want to find the differential/linear trails which means finding the exact values of all the bit-level inputs and outputs, then all these state variables must be binary which give us an integer programming model rather than a mixed integer programming model.

In order to find the differential/linear trails of a given input/output differential/linear approximation, we set the corresponding binary variables for each input/output to 1 if it is an active bit in the input/output and to 0 otherwise. In this paper, we follow the MIP model for linear cryptanalysis presented in [24] (minimize the number of variables appearing in quadratic terms of the linear approximation of SIMON's non-linear function) and use the algorithm presented in [20] for computing the squared correlation for the SIMON nonlinear function.

In Section 4, we propose a hybrid method that combines the matrix method and the MIP method to amplify the differential probability or the squared correlation of a specified input and output differences or masks. Using this method we are able to find a 17-round linear approximation for SIMON-48.

4 Time-Memory Trade-off Method

Since the matrix method consumes huge memory and the MIP method takes time to enumerate a certain number of trails. It seems reasonable to trade-off the time and memory by combining both methods to get better differential/correlation estimations. Here we combine the correlation matrix method with the recent technique for finding differentials and linear hulls in order to obtain a better estimation for the correlations or differentials of a linear and differential approximations respectively.

The idea is to find good disjoint approximations through the matrix and the mixed integer programming model. Assume that our target is an r-round linear hull (α, β), where α is the input mask and β is the output mask. The matrix method is used to find the resulting correlation from trails that have Hamming weight at most m for each round, from now on we will call them "light trails". The MIP method is used to find the resulting correlation from trails that have Hamming weight at least $m + 1$ at one of their rounds, from now on we will call them "heavy trails".

Now if the target number of rounds is high, then the MIP method might not be effective in finding good estimation for the heavy trails as it will take time to collect all those trails.

Therefore, in order to overcome this, we split the cipher into two parts, the first part contains the first r_1 rounds and the second part contains the remaining $r_2 = r - r_1$ rounds. Assume $r_1 > r_2$, where r_2 is selected in such a way that the MIP solution is reachable within a reasonable computation time. Now, we show how to find two disjoint classes that contains heavy trails. The first class contains an r_1-round linear hull (α, γ_i) consisting of light trails found through the matrix method at the first r_1 rounds glued together with an r_2-round linear hulls (γ_i, β) consisting of heavy trails found through the MIP method. We call this class, the lower-round class. The second class basically reverse the previous process, by having an r_1-round linear hull of heavy weight trails found through MIP method glued with an r_2-round linear hull containing light trails found through the matrix method. We call this class the upper-round class. Now, adding the estimations from these two classes (upper-round and lower-round classes) gives us the estimation of the correlation of the heavy trails which will be added to the r-round linear hull of the light trails found through the matrix method. We can also include a middle-round class surrounded by upper lightweight trails and lower lightweight trails found by the matrix method.

Next we describe how to find the heavy trails using MIP with the Big M constraints which is a well known technique in optimization.

4.1 Big M Constraints

Suppose that only one of the following two constraints is to be active in a given MIP model.

$$\text{either} \quad \sum_{i,j} f_i X_{ij} \geq c_1 \tag{1}$$

$$\text{or} \quad \sum_{i,k} g_i X_{ik} \geq c_2 \tag{2}$$

The above situation can be formalized by adding a binary variable y as follows:

$$\sum_{i,j} f_i X_{ij} + My \geq c_1 \tag{3}$$

$$\sum_{i,k} g_i X_{ik} + M(1-y) \geq c_2 \tag{4}$$

where M is a big positive integer and the value of y indicates which constraint is active. So y can be seen as an indicator variable. One can see that when $y = 0$, the first constraint is active while the second constraint is inactive due to the positive big value of M. Conversely, when y $= 1$, the second constraint is active.

The above formulation can be generalized to the case where we have q constraints under the condition that only p out of q constraints are active. The generalization can be represented as follows:

$$\begin{aligned}
\sum_{i,j} f_i X_{ij} + My_1 &\geq c_1 \\
\sum_{i,k} g_i X_{ik} + My_2 &\geq c_2 \\
&\vdots \\
\sum_{i,l} h_i X_{il} + My_q &\geq c_q \\
\sum_{i=1}^{l} y_i &= q - p
\end{aligned}$$

where y_i is binary for all i. Sometimes, we might be interested on the condition where at least p out of the q constraints are active. This can be achieved by simply changing the last equation in the constraints above, $\sum_{i=1}^{l} y_i = q - p$ to $\sum_{i=1}^{l} y_i \leq q - p$. This turns out to be useful in our Hybrid method as it will allow us to find r-round trails which have a heavy Hamming weight on at least one of the r rounds.

5 Linear Hull Effect in SIMON-32 and SIMON-48

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

5.1 Correlation of the SIMON F Function

This section provides an analysis on some linear properties of the SIMON F function regarding the squared correlation. This will assist in providing an intuition around the design rationale when it comes to linear properties of SIMON round Function F. A general linear analysis was applied on the F function of SIMON, with regards to limits around the squared correlations for all possible Hamming weights on input masks α and output masks β, for SIMON-32/64.

5.2 Constructing Correlation Submatrix for SIMON

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

Proposition 1. *Correlation of a one-round linear approximation [10]. Let $\alpha = (\alpha_L, \alpha_R)$ and $\beta = (\beta_L, \beta_R)$ be the input and output masks of a one-round linear approximation of SIMON. Let α_F and β_F be the input and output masks of the SIMON F function. Then the correlation of the linear approximation (α, β) is $C(\alpha, \beta) = C_F(\alpha_F, \beta_F)$ where $\alpha_F = \alpha_L \oplus \beta_R$ and $\beta_F = \beta_L = \alpha_R$.*

As our goal is to perform a linear attack on SIMON, we construct a squared correlation matrix in order to compute the average squared correlation (the sum of the squares of the correlations of all trails) in order to estimate the required data complexity. Algorithm 1 constructs a squared correlation submatrix whose input and output masks have Hamming weight less than a certain Hamming weight m, where the correlation matrix is deduced from the algorithm proposed in [20].

The size of the submatrix is $\sum_{i=0}^{m} \binom{2n}{i} \times \sum_{i=0}^{m} \binom{2n}{i}$ where n is the block size of SIMON's F function. One can see that the time complexity is in the order of $2^n \sum_{i=0}^{m} \binom{2n}{i}$ arithmetic operations. The submatrix size is large when $m > 5$, but most of its elements are zero and therefore it can easily fit in memory using a sparse matrix storage format. The table below

Algorithm 1: Construction of SIMON's Correlation Submatrix

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

for all output masks β with Hamming weight $\leq m$ **do**
  Extract from β the left/right output masks β_L and β_R.
  $\alpha_R \leftarrow \beta_L$.
  Compute $C(\alpha_F, \beta_L)$ to SIMON's F function for all possible α_F using the algorithm proposed in [20].
  for all input masks α_F to SIMON's F function **do**
    $c \leftarrow C(\alpha_F, \beta_L)$.
    $\alpha_L \leftarrow \alpha_F \oplus \beta_R$.
    $\alpha = \alpha_L || \alpha_R$.
    if $c \neq 0$ **and** Hamming weight of $\alpha \leq m$ **then**
      $i \leftarrow map(\alpha)$. {map α to a row index i in the matrix M}
      $j \leftarrow map(\beta)$. {map α to a column index j in the matrix M}
      $M(i, j) = c \times c$.
    end if
  end for
end for

shows the number of nonzero elements of the squared correlation submatrices of SIMON-32/K when $1 \leq m \leq 9$. These matrices are very sparse. For instance, based on our experimental results when $m \leq 8$, the density of the correlation matrix is very low, namely $\frac{133253381}{15033173 \times 15033173} \approx 2^{-20.7}$.

5.3 Improved Linear Approximations

One can see that Algorithm 1 is highly parallelizable. This means the dominating factor is the memory complexity instead of time complexity. We constructed a sparse squared correlation matrix of SIMON-32/K with input and output masks that have Hamming weight ≤ 8. Using this matrix, we find a 14-round linear approximations with an average squared correlation $< 2^{-32}$ for SIMON-32/K. We also get better estimations for the previously found linear approximations which were estimated before using only a single linear characteristic rather than considering many linear characteristics with the same input and output masks. For example, in [4], the squared correlation of the 9-round single linear characteristic with input mask `0x01110004` and output mask `0x00040111` is 2^{-20}. Using our matrix, we find that this same approximation has a squared correlation $\approx 2^{-18.4}$ with $11455 \approx 2^{13.5}$ trails, which gives us an improvement by a factor of $2^{1.5}$. Note that this approximation can be found using a smaller correlation matrix of Hamming weight ≤ 4 and we get an estimated squared correlation equal to $2^{-18.83}$ and only 9 trails. Therefore, the large number of other trails that cover Hamming weights ≥ 5 is insignificant as they only cause a factor of $2^{0.5}$ improvement.

Also, the 10-round linear characteristic in [6] with input mask `0x01014404` and output mask `0x10004404` has squared correlation 2^{-26}. Using our correlation matrix, we find that this same approximation has an estimated squared correlation $2^{-23.2}$ and the number of trails is $588173 \approx 2^{19.2}$. This gives an improvement by a factor of 2^3. Note also that this approximation can be found using a smaller correlation matrix with Hamming weight ≤ 5 and we get an estimated squared correlation equal to $2^{-23.66}$ and only 83 trails. So the large number of other trails resulting covering Hamming weights ≥ 5 is insignificant as they only cause a factor of $2^{0.4}$ improvement. Both of these approximations give us squared correlations less than 2^{-32} when considering more than 12 rounds.

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

Improved 14-round Linear Hulls on SIMON-32 (Squared correlation matrix only). Consider a squared correlation matrix M whose input and output masks have Hamming weight m. When $m \geq 6$, raising the matrix to the rth power, in order to estimate the average squared correlation, will not work as the resulting matrix will not be sparse even when r is small. For example, we are able only to compute M^6 where M is a squared correlation matrix whose masks have Hamming weight ≤ 6. Therefore, we use matrix-vector multiplication or row-vector matrix multiplications in order to estimate the squared correlations for any number of rounds r.

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

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

Table 2. 14-round linear hulls for SIMON-32/K found, using Hamming weight ≤ 9

α	β	$\log_2 c^2$	$\log_2 N_t$
0x80000000	0x00800020, 0x00800060, 0x00808020, 0x00808060	-30.5815	28.11
0x02000000	0x00028000, 0x00028001, 0x00028200, 0x00028201	-30.5815	28.10
0x00800000	0x80002000, 0x80002080, 0x80006000, 0x80006080	-30.5816	28.06
0x00400000	0x40001000, 0x40001040, 0x40003000, 0x40003040	-30.5815	28.11
0x00040000	0x04000100, 0x04000104, 0x04000300, 0x04000304	-30.5816	28.10
0x*00010000*	0x*01000040*, 0x01000041, 0x010000C0, 0x010000C1	-30.5814	28.11

Improved 17-round Linear Hulls on SIMON-48 (Squared correlation matrix + MIP). Using a squared correlation matrix of SIMON-48 having input and output masks with Hamming weight ≤ 6 and size 83278000×83278000, we found that a 17-round linear approximation with input mask $0x404044000001$ and output mask $0x000001414044$ ($0x404044000001 \xrightarrow{17-round} 0x000001C04044$) has squared correlation $2^{-49.3611}$. Also the output masks $0x000001414044$ and $0x000001414044$ yield a similar squared correlation $2^{-49.3611}$. Unlike the case for SIMON-32 where we can easily use brute force to compute the squared correlation of a 1-round linear approximation, the squared correlation matrix for SIMON-48 was created using the algorithm proposed in [20]. Again the matrix is sparse and it has $48295112 \approx 2^{25.53}$ nonzero elements.

However, it seems difficult to build matrices beyond Hamming weight 6 for SIMON-48. Therefore we use our time-memory trade-off method to improve the squared correlation of the linear approximation $0x404044000001 \xrightarrow{17-round} 0x000001414044$.

To find the lower class where the heavy trails are on the bottom are glued with the light trails on top. The light trails are found using the matrix method for 11 rounds and the heavy trails are found using the MIP method for 6 rounds. Combining them both we get the 17-round lower class trails. In more detail, we fix the input mask to $0x404044000001$ and we use the matrix method to find the output masks after 11 rounds with the most significant squared correlation. The best output masks are $0x001000004400$, $0x001000004410$ and $0x0010000044C0$, each give an 11-round linear hull with squared correlation $2^{-28.6806}$ coming from 268 light trails. We first

create a 6-round MIP model with $0x001000004400$ as an input mask and with the target output mask $0x000001414044$ as the output mask for the 6-round MIP model $0x001000004400 \xrightarrow{6-round} 0x000001414044$. In order to find heavy trails we added the big M constraints described in Section 4.1 and set $M = 200$ and all the c_i's to 7 from the end of round 1 to beginning of round 5. So $q = 5$, setting $p = 1$ and using $\sum_{i=1}^{l} y_i \leq q - p = 4$, we guarantee that the trails found will have Hamming weight at least 7 at one of the rounds. The constraints should be set as follows:

$$\begin{aligned}
\sum_{i=0}^{47} s_{48+i} + 200y_1 &\geq 7 \\
\sum_{i=0}^{47} s_{96+i} + 200y_2 &\geq 7 \\
\sum_{i=0}^{47} s_{144+i} + 200y_3 &\geq 7 \\
\sum_{i=0}^{47} s_{192+i} + 200y_4 &\geq 7 \\
\sum_{i=0}^{47} s_{240+i} + 200y_5 &\geq 7 \\
\sum_{i=1}^{5} y_i &\leq 4
\end{aligned}$$

where y_j is a binary variable and $s_{48.j+i}$ is a binary variable representing the intermediate mask value in the jth round at the ith position.

Limiting our MIP program to find 512 trails for the specified approximation, we find that the estimated squared correlation is $2^{-22.3426}$. Combining the light trails with the heavy, we get a 17-round sub approximation whose squared correlation is $2^{-28.6806} \times 2^{-22.3426} = 2^{-51.0232}$. To get a better estimation, we repeated the above procedure for the other output masks $0x001000004410$ and $0x0010000044C0$ and get an estimated squared correlation equivalent to $2^{-28.6806} \times 2^{-24.33967} = 2^{-53.02027}$ and $2^{-28.6806} \times 2^{-24.486272} = 2^{-53.166872}$ respectively. Adding all these three sub linear approximations we get an estimated squared correlation equivalent to $2^{-51.0232} + 2^{-53.02027} + 2^{-53.166872} \approx 2^{-50.4607}$. Moreover, we repeat the same procedure for the 27 next best 11-round linear approximations and we get $2^{-49.3729}$ as a total estimated squared correlation for our 17-round lower class trails ($0x404044000001 \xrightarrow{17-round} 0x000001414044$). All these computations took less than 20 hrs on a standard laptop (See Table 11 in the Appendix).

Similarly to find the upper class where the heavy trails are on the top, are glued with the light trails on bottom. The light trails are found using the matrix method for 11 rounds and the heavy trails are found using the MIP method for 6 rounds under the same big M constraints described above. Combining them both we get the 17-round upper class trails. In more detail, we fix the output mask to $0x000001414044$ and we use the matrix method to find the input masks with the most significant squared correlation after 11 rounds. The best input masks are $0x004400001000$, $0x004410001000$, $0x004C00001000$ and $0x004C10001000$, each give an 11-round linear hull with squared correlation $2^{-28.6806}$ coming from 268 light trails. We first create a 6-round MIP model with $0x004400001000$ as an output mask and the target input mask $0x404044000001$ as the input mask for the 6-round MIP model $0x404044000001 \xrightarrow{6-round} 0x004400001000$. Limiting our MIP program to find 512 trails for the specified approximation, we find that the estimated

squared correlation is $2^{-22.3426}$. Combining the light trails with the heavy, we get a 17-round sub approximation whose squared correlation is $2^{-28.6806} \times 2^{-22.3426} = 2^{-51.0232}$. Repeating the above procedure for the other three input masks $0x04410001000$, $0x004C00001000$ and $0x004C10001000$, we get an estimated squared correlation equivalent to $2^{-28.6806} \times 2^{-24.33967} = 2^{-53.02027}$, $2^{-28.6806} \times 2^{-24.486272} = 2^{-53.166872}$ and $2^{-28.6806} \times 2^{-23.979259} = 2^{-52.659859}$ respectively. Adding all these four sub linear approximations we get an estimated squared correlation equivalent to $2^{-51.0232} + 2^{-53.02027} + 2^{-53.166872} + 2^{-52.659859} \approx 2^{-50.1765}$. Repeating the same procedure for the 26 next best input masks and adding them up, we get a total squared correlation equivalent to $2^{-49.3729}$ as a total estimated squared correlation for our 17-round upper class trails ($0x404044000001 \xrightarrow{17-round} 0x000001414044$). All these computations took less than 18 hrs on a standard laptop (See Table 12 in the Appendix).

Adding the contributions of the lower and upper classes found through the above procedure to the contribution of the light trails found through the matrix method, we get $2^{-49.3729} + 2^{-49.3729} + 2^{-49.3611} = 2^{-47.7840} \approx 2^{-47.78}$ as a total estimation for the squared correlation of the 17-round linear hull ($0x404044000001 \xrightarrow{17-round} 0x000001414044$).

5.4 Key Recovery Attack on 24 and 23 Rounds of SIMON-32/K using 14-Round Linear Hull

We extend the given linear hull for 14 rounds of SIMON-32/K (highlighted masks in the last row of Table 2) by adding some rounds to the beginning and the end of the cipher. The straightforward approach is to to start with the input mask of the 14-round linear hull (e.g. $(\Gamma_0, -)$) and go backwards to add some rounds to the beginning. With respect to Figure 1, we can append an additional round to the beginning of the cipher. Since SIMON injects the subkey at the end of its round function, this work does not have any computational complexity. More precisely, for the current 14-round linear hull, we evaluate $((X_L^i)_0 \oplus (X_R^{i+14})_6 \oplus (X_L^{i+14})_8)$ to filter wrong guesses. On the other hand, we have $(X_L^i)_0 = (F(X_L^{i-1}))_0 \oplus ((X_R^{i-1})_0 \oplus (K^i)_0$, where $(F(X_L^{i-1}))_0 = (X_L^{i-1})_{14} \oplus ((X_L^{i-1})_{15} \& (X_L^{i-1})_8)$. Hence, if we add a round in the backwards direction, i.e. round $i-1$, we know X_R^{i-1} and X_L^{i-1} we can determine $F(X_L^{i-1})$. Then it is possible to use the following equation to filter wrong keys, instead of $((X_L^i)_0 \oplus (X_R^{i+14})_6 \oplus (X_L^{i+14})_8)$, where $(K^i)_0$ is an unknown but a constant bit (in Figure 1 such bits are marked in red):

$$(F(X_L^{i-1}))_0 \oplus (X_R^{i-1})_0 \oplus (K^i)_0 \oplus (X_R^{i+14})_6 \oplus (X_L^{i+14})_8 = (X_L^{i-1})_{14} \oplus ((X_L^{i-1})_{15} \& (X_L^{i-1})_8) \oplus (X_R^{i-1})_0 \oplus (K^i)_0 \oplus (X_R^{i+14})_6 \oplus (X_L^{i+14})_8.$$

We can continue our method to add five rounds to the beginning of linear hull at the cost of guessing some bits of subkeys. To add more rounds in the backwards direction, we must guess the bit

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

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

The same approach can be used to add five rounds to the end of linear hull at the cost of guessing some bits of subkeys. More details are depicted in Figure 1.

On the other hand, in [29], Wang *et al.* presented a divide and conquer approach to add extra rounds to their impossible differential trail. We note that it is possible to adapt their approach to extend the key recovery using the exist linear hull over more rounds. Hence, one can use the

14-round linear hull and extend it by adding extra rounds to its beginning and its end. We add five rounds to the beginning and five rounds to the end of the linear hull to attack 24-round variant of SIMON-32/K. This key recovery attack processes as follows:

1. Let T_{max} and T_{cur} be counters (initialized by 0) and SK_{can} be a temporary register to store the possible candidate of the subkey.
2. Collect $2^{30.59}$ known plaintext and corresponding ciphertext pairs (p_i, c_i) for 24-round SIMON-32/64 and store them in a table $\mathcal{T}$.
3. Guess a value for the subkeys involved in the first five rounds of reduced SIMON-32/K, i.e. $(K^{i-4})[0, 2 \dots 4, 5, 6, 7, 9 \dots 13, 14] \| (K^{i-3})[4, 5, 6, 8, 11, 12, 13, 14, 15] \| (K^{i-2})[0, 6, 7, 13, 14] \| (K^{i-1})[8, 15]$ and do as follows (note that the red subkey bits involved in the rounds are the constant bits and do not have to be guessed):
   (a) For any $p_j \in \mathcal{T}$ calculate the partial encryption of the first five rounds of reduced SIMON-32/K and find $\mathcal{V}_j = (X_L^i)[0] \oplus (K^i)[0] \oplus (K^{i-1})[14] \oplus (K^{i-2})[12] \oplus (K^{i-3})[10] \oplus (K^{i-5})[8]$.
   (b) Guess the bits of subkeys $K^{i+19}[0 \dots 4, 5, 6, 7, 8 \dots 10, 11, 12, 13, 14, 15]$, $K^{i+18}[1, 2, 3, 4, 5, 6, 8, 10, 11, 12, 14, 15]$, $K^{i+17}[0, 3, 4, 6, 7, 12, 13]$, and $K^{i+16}[5, 14]$, step by step.
   (c) For any $c_j \in \mathcal{T}$:
      i. calculate the partial decryption of the last five rounds of reduced SIMON-32/K and find $\mathcal{W}_j = (X_L^{i+14})[8] \oplus (X_R^{i+14})[6] \oplus (K^{i+15})[6] \oplus (K^{i+16})[4, 8] \oplus (K^{i+17})[2] \oplus (K^{i+18})[0]$.
      ii. If $\mathcal{V}_j = \mathcal{W}_j$ then increase T_{cur}.
   (d) If $T_{max} < T_{cur}$ (or resp. $T_{max} < (2^{32} - T_{cur})$) update T_{max} and SK_{can} by T_{cur} (resp. $2^{32} - T_{cur}$) and the current guessed subkey respectively.
4. Return SK_{can}.

Following the approach presented in [29], guessing the bits of subkeys$K^{i+19}[0 \dots 4, 5, 6, 7, 8 \dots 10, 11, 12, 13, 14, 15]$, $K^{i+18}[1, 2, 3, 4, 5, 6, 8, 10, 11, 12, 14, 15]$, $K^{i+17}[0, 3, 4, 6, 7, 12, 13]$, and $K^{i+16}[5, 14]$, step by step, to find the amount of $\mathcal{W}_j = (X_L^{i+14})[8] \oplus (X_R^{i+14})[6] \oplus (K^{i+15})[6] \oplus (K^{i+16})[4, 8] \oplus (K^{i+17})[2] \oplus (K^{i+18})[0]$, for any c_j, are done as follows:

1. Let T_2 be a vector of 2^{32} counters which correspond to all possible values of $\mathcal{V}_j \| (X_L^{i+19})[0 \dots 7, 10 \dots 14] \| (X_R^{i+19})[0 \dots 6, 8 \dots 15] \| (X_R^{i+18})[8, 9, 15]$ (denoted as S_2^1). Guess the subkey bit $(K^{i+19})[8, 9, 15]$. decrypt partially for each possible value of S_1^1 $(\mathcal{V}_j \| (X_L^{i+19}) \| (X_R^{i+19}))$ to obtain the value of $(X_R^{i+18})[8, 9, 15]$ (and hence S_2^1), then increase the corresponding counter T_{2,S_2^1}.
2. Guess the subkey bits $(K^{i+19})[5, 14]$, $(K^{i+19})[1, 10, 11]$, $(K^{i+19})[12]$, $(K^{i+19})[13]$, and $(K^{i+19})[0, 2, 3, 4, 6, 7]$ step by step (see Table 3), do similarly to the above and finally get the values of the counters corresponding to the state $\mathcal{V}_j \| (X_L^{i+18})[0 \dots 6, 8, 10 \dots 12, 14, 15] \| (X_R^{i+18})$ (denoted as S_0^2).
3. Let X_1 be a vector of 2^{29} counters which correspond to all possible values of $\mathcal{V}_j \| (X_L^{i+18})[0 \dots 5, 8, 10 \dots 12, 14, 15] \| (X_R^{i+18})[0 \dots 4, 6 \dots 15] \| (X_R^{i+17})[6]$ (denoted as S_1^2). Guess the subkey bit $(K^{i+18})[6]$. For each possible value of S_0^2 $(\mathcal{V}_j \| (X_L^{i+18})[0 \dots 6, 8, 10 \dots 12, 14, 15] \| (X_R^{i+18}))$, do partial decryption to derive the value of $(X_R^{i+17})[6]$ and add T_{7,S_7^1} to the corresponding counter X_{1,S_1^2} according to the value of S_1^2. After that, guess the subkey bits $(K^{i+18})[15]$, $(K^{i+18})[1]$, $(K^{i+18})[3, 12]$, $(K^{i+18})[2]$, $(K^{i+18})[11]$, $(K^{i+18})[10]$, $(K^{i+18})[14]$, and $(K^{i+18})[4, 5, 8]$, step by step (see Table 4). Do similarly to the above and eventually obtain the values of the counters corresponding to the state $\mathcal{V}_j \| (X_L^{i+17})[0', 2 \dots 4, 6, 7, 12, 13] \| (X_R^{i+17})[0 \dots 6, 8, 10 \dots 12, 14, 15]$ (denoted as S_0^3) where $(X_R^{i+17})[0'] = (X_R^{i+17})[0] \oplus (K^{i+18})[0]$.
4. Let Y_1 be a vector of 2^{21} counters which correspond to all possible values of $\mathcal{V}_j \| (X_L^{i+17})[0, 2, 3, 6, 7, 12, 13] \| (X_R^{i+17})[0 \dots 2, 4 \dots 6, 8, 10 \dots 12, 14, 15] \| (X_R^{i+16})[4]$ (denoted as S_1^3). Guess the subkey bit $(K^{i+17})[4]$. For each possible value of S_0^3 $(\mathcal{V}_j \| (X_L^{i+17})[0, 2 \dots 4, 6, 7, 12, 13] \| (X_R^{i+17})$

$[0 \dots 6, 8, 10 \dots 12, 14, 15]$), do partial decryption to derive the value of $(X_R^{i+16})[4]$ and add X_{9,S_9^2} to the corresponding counter Y_{1,S_1^3} according to the value of S_1^3. After that, guess the subkey bits $(K^{i+17})[3]$, $(K^{i+17})[12]$, $(K^{i+17})[13]$, $(K^{i+17})[7]$, and $(K^{i+17})[0, 6]$, step by step (see Table 5). Do similarly to the above and eventually obtain the values of the counters corresponding to the state $\mathcal{V}_j \| (X_L^{i+16})[4, 5, 8, 14] \| (X_R^{i+16})[0, 2', 3, 4, 6, 7, 12, 13]$ (denoted as S_0^4) where $\| (X_R^{i+16})[2'] = (X_R^{i+16})[2] \oplus (K^{i+17})[2]$.

5. Let Z_1 be a vector of 2^6 counters which correspond to all possible values of $\mathcal{V}_j \| (X_L^{i+15})[6] \| (X_R^{i+15})[4, 5, 8, 14]$ (denoted as S_1^4) where $(X_R^{i+15})[4'] = (X_R^{i+15})[4] \oplus (K^{i+16})[4]$ and $(X_R^{i+15})[8'] = (X_R^{i+15})[8] \oplus (K^{i+16})[8]$. Guess the subkey bits $(K^{i+16})[5, 14]$ and for each possible value of S_0^4 $(\mathcal{V}_j \| (X_L^{i+16})[4, 5, 8, 14] \| (X_R^{i+16})[0, 2, 3, 4, 6, 7, 12, 13])$ do partial decryption to derive the value of $(X_R^{i+15})[5, 14]$ and add Y_{6,S_6^3} to the corresponding counter Z_{1,S_1^4} according to the value of S_1^4.
6. Let W_{1,S_1^5} be a vector of 2^4 counters which correspond to all possible values of $\mathcal{V}_j \| (X_L^{i+14})[4', 8'] \| (X_R^{i+14})[6']$ (denoted as S_1^5) where $(X_R^{i+14})[6'] = (X_R^{i+14})[6] \oplus (K^{i+15})[6]$, $(X_L^{i+14})[4'] = (X_L^{i+14})[4] \oplus (K^{i+16})[4] \oplus (K^{i+17})[2] \oplus (K^{i+18})[0]$, and $(X_L^{i+14})[8'] = (X_L^{i+14})[8] \oplus (K^{i+16})[8]$. This state are extracted of S_1^4 and add Z_{1,S_1^4} to the corresponding counter W_{1,S_1^5} according to the value of S_1^5 (See Table 7).
7. Let O be a vector of 2^2 counters which correspond to all possible values of $\mathcal{V}_j \| \mathcal{W}_j$ (Note that $\mathcal{W}_j = (X_L^{i+14})[8] \oplus (X_R^{i+14})[6] \oplus (K^{i+15})[6] \oplus (K^{i+16})[4, 8] \oplus (K^{i+17})[2] \oplus (K^{i+18})[0]$ and can be extracted from S_1^5). Each possible value of S_1^5 is converted to $\mathcal{V}_j \| \mathcal{W}_j$ and W_{1,S_1^5} and is added to the relevant counter in O according to the value of $\mathcal{V}_j \| \mathcal{W}_j$. Suppose that O_0 means that $\mathcal{V}_j = 0$ and $\mathcal{W}_j = 0$ and O_3 means that $\mathcal{V}_j = 1$ and $\mathcal{W}_j = 1$. If $O_0 + O_3 \geq T_{max}$ or $2^{32} - (O_0 + O_3) \geq T_{max}$ keep the guessed bits of subkey information as a possible subkey candidate, and discard it otherwise.

Attack Complexity. The time complexity of each sub-step was computed as shown in the Tables 3, 4, 5, 6 and 7. The time complexity of the attack is about $2^{63.9}$. It is clear that, the complexity of this attack is only slightly less than exhaustive search. However, if we reduce the last round and attack 23 round of SIMON-32/K then the attack complexity reduces to 2^{50} which is yet the best key-recovery attack on SIMON-32/K for such number of rounds.

5.5 Key Recovery Attack on SIMON-48/K using 17-Round Linear Hull

Given the 17-round approximation for SIMON-48, introduced in Section 5.3, we apply the approach presented in Section 5.4 to extend key recovery over more number of rounds. Our key recovery for SIMON-48/72 and SIMON-48/96 covers 23 and 24 rounds respectively. The data complexity for these attacks is $2^{-47.78}$ and their time complexities are $2^{62.10}$ and $2^{83.10}$ respectively. Since the attack procedure is similar to the approach presented in section 5.4, we do not repeat it. Related tables and complexity of each step of the attack for SIMON-48/96 has been presented in Appendix B (The time complexity of each sub-step was computed as shown in the Tables 8, 9, and 10). To attack SIMON-48/72, we add three rounds in forward direction instead of the current four rounds. Hence, the adversary does not need to guess the average 21 bits of the key in the last round of Figure 2.

6 Conclusion

In this paper, we propose a time-memory tradeoff that finds better differential/linear approximation. The method benefits from the correlation matrix method and the MIP method to improve the estimated squared correlation or differential probability. Using MIP we can find the

trails that are missed by the matrix method. This method enables us to find a 17-round linear hull for SIMON-48. Moreover, we have analyzed the security of some variants of SIMON against different variants of linear cryptanalysis, i.e. classic and linear hull attacks. We have investigated the linear hull effect on SIMON-32/64 and SIMON-48/96 using the correlation matrix of the average squared correlations and presented best linear attack on this variant.

Regarding SIMON-64, the squared correlation matrix which we are able to build and process holds masks with Hamming weight ≤ 6. Using only the matrix and going for more than 20 rounds, the best squared correlation we found has very low squared correlation $< 2^{-70}$ and this is because we are missing good trails with heavy Hamming weights. Applying our time-memory trade-off has not been effective due to the large number of rounds. However, trying to find good trails with heavy Hamming weight in the middle beside the upper and lower classes might yield better results. We note here that we have been looking for fast solutions. It could be that trying to add up many linear trails for some days or weeks can yield better results. Our method seems to be slow due to the slow processing of the huge squared correlation matrix. So it would be very interesting to build a dedicated sparse squared correlation matrix for SIMON-64 in order to speed up the selection of the intermediate masks in our time-memory trade-off method. This will allow us to select many intermediate masks which might yield better results. One interesting target would be also to apply this method to the block cipher PRESENT which also allows low Hamming weight trails and see if we can go beyond the current best 24-round linear approximations [1].

Comparing the complexities of our linear attacks for SIMON-32 with the differential attacks for SIMON-32 exploiting the differentials presented in [20] is an open issue as no key recovery attacks was described in [20] .

Our time-memory trade-off method uses the MIP approach to find the heavy trails, it would be interesting to investigate the performance of our method when the MIP approach is replaced with other approaches such as the SAT/SMT models used in [20] or the dedicated branch-and-bound method used in [11].

Acknowledgments

The authors would like to thank Lars Knudsen, Stefan Kölbl, Martin M. Lauridsen, Arnab Roy and Tyge Tiessen for many useful discussions about linear and differential cryptanalysis of SIMON. Special thanks go to Anne Canteaut for the valuable comments and suggestions to improve the quality of the paper.

References

1. Mohamed Ahmed Abdelraheem. Estimating the Probabilities of Low-Weight Differential and Linear Approximations on PRESENT-Like Ciphers. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *ICISC 2012*, volume 7839 of *Lecture Notes in Computer Science*, pages 368–382. Springer, 2012.
2. Mohamed Ahmed Abdelraheem, Javad Alizadeh, Hoda AlKhzaimi, Mohammad Reza Aref, Nasour Bagheri, Praveen Gauravaram, and Martin M. Lauridsen. Improved Linear Cryptanalysis of Round Reduced SIMON. *IACR Cryptology ePrint Archive*, 2014:681, 2014.
3. Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential Cryptanalysis of Reduced-Round Simon. *IACR Cryptology ePrint Archive*, 2013:526, 2013.
4. Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential Cryptanalysis of Round-Reduced Simon and Speck. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, volume 8540 of *Lecture Notes in Computer Science*, pages 525–545. Springer, 2015.
5. Javad Alizadeh, Hoda A. Alkhzaimi, Mohammad Reza Aref, Nasour Bagheri, Praveen Gauravaram, Abhishek Kumar, Martin M. Lauridsen, and Somitra Kumar Sanadhya. Cryptanalysis of SIMON Variants with Connections. In *RFIDSec'14*, volume 8651 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014.
6. Javad Alizadeh, Nasour Bagheri, Praveen Gauravaram, Abhishek Kumar, and Somitra Kumar Sanadhya. Linear Cryptanalysis of Round Reduced SIMON. *IACR Cryptology ePrint Archive*, 2013:663, 2013.

7. Tomer Ashur. Improved linear trails for the block cipher simon. Cryptology ePrint Archive, Report 2015/285, 2015. `http://eprint.iacr.org/`.
8. Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A Lightweight Hash. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2010.
9. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. *IACR Cryptology ePrint Archive*, 2013:404, 2013.
10. Eli Biham. On Matsui's Linear Cryptanalysis. In Alfredo De Santis, editor, *EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 341–355. Springer, 1994.
11. Alex Biryukov, Arnab Roy, and Vesselin Velichkov. Differential analysis of block ciphers SIMON and SPECK. 8540:546–570, 2015.
12. Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: A Lightweight Hash Function. In Preneel and Takagi [22], pages 312–325.
13. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
14. Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 179–199. Springer, 2014.
15. Christophe De Cannière and Bart Preneel. Trivium. In Robshaw and Billet [23], pages 244–266.
16. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.
17. Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON Family of Lightweight Hash Functions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.
18. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED Block Cipher. In Preneel and Takagi [22], pages 326–341.
19. Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. The Grain Family of Stream Ciphers. In Robshaw and Billet [23], pages 179–190.
20. Stefan Kölbl, Gregor Leander, and Tyge Tiessen. Observations on the SIMON Block Cipher Family. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 161–185. Springer, 2015.
21. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Inscrypt 2011*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.
22. Bart Preneel and Tsuyoshi Takagi, editors. *CHES*.
23. Matthew J. B. Robshaw and Olivier Billet, editors. *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*. Springer, 2008.
24. Danping Shi, Lei Hu, Siwei Sun, Ling Song, Kexin Qiao, and Xiaoshuang Ma. Improved Linear (hull) Cryptanalysis of Round-reduced Versions of SIMON. *IACR Cryptology ePrint Archive*, 2014:973, 2014.
25. Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu. Towards Finding the Best Characteristics of Some Bit-oriented Block Ciphers and Automatic Enumeration of (Related-key) Differential and Linear Characteristics with Predefined Properties. *IACR Cryptology ePrint Archive*, 2014:747, 2014.
26. Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Computer Science*, pages 158–178. Springer Berlin Heidelberg, 2014.
27. Ning Wang, Xiaoyun Wang, Keting Jia, and Jingyuan Zhao. Differential Attacks on Reduced SIMON Versions with Dynamic Key-guessing Techniques. *IACR Cryptology ePrint Archive*, 2014:448, 2014.
28. Ning Wang, Xiaoyun Wang, Keting Jia, and Jingyuan Zhao. Improved Differential Attacks on Reduced SIMON Versions. *IACR Cryptology ePrint Archive*, 2014:448, 2014.
29. Qingju Wang, Zhiqiang Liu, Kerem Varici, Yu Sasaki, Vincent Rijmen, and Yosuke Todo. Cryptanalysis of Reduced-Round SIMON32 and SIMON48. In Willi Meier and Debdeep Mukhopadhyay, editors, *INDOCRYPT 2014*, volume 8885 of *Lecture Notes in Computer Science*, pages 143–160. Springer, 2014.
30. Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D. Aagaard, and Guang Gong. The Simeck family of lightweight block ciphers, 2015. To appear in the proceeding of the Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2015.

A Steps of the Key Recovery Attack on SIMON-32/64

Table 3. Step 1 of key recovery attack on SIMON-32/64

i	Input (S_i^1)	Guessed subkey bit	Output (S_{i+1}^1)	Counter of S_{i+1}^1
0	$(X_L^{i-5})\|\|(X_R^{i-5})$	$(K^{i-4})[0,2\dots4,5,6,7,9\dots13,14]\|\|(K^{i-3})[4,5,6,8,11,12,13,14,15]\|\|(K^{i-2})[0,6,7,13,14]\|\|(K^{i-1})[8,15]$	$\mathcal{V}_j = (X_L^i)[0] \oplus (K^i)[0] \oplus (K^{i-1})[14] \oplus (K^{i-2})[12] \oplus (K^{i-3})[10] \oplus (K^{i-5})[8]$	T_{1,S_1^1}
1	$\mathcal{V}_j\|\|(X_L^{i+19})$ $\|\|(X_R^{i+19})$	$(K^{i+19})[8,9,15]$	$\mathcal{V}_j\|\|(X_L^{i+19})[0\dots7,10\dots14]$ $\|\|(X_R^{i+19})[0\dots6,8\dots15]$ $\|\|(X_R^{i+18})[8,9,15]$	T_{2,S_2^1}
2	$\mathcal{V}_j\|\|(X_L^{i+19})[0\dots7,10\dots14]$ $\|\|(X_R^{i+19})[0\dots6,8\dots15]$ $\|\|(X_R^{i+18})[8,9,15]$	$(K^{i+19})[5,14]$	$\mathcal{V}_j\|\|(X_L^{i+19})[0\dots4,6,7,10\dots13]$ $\|\|(X_R^{i+19})[0\dots6,8\dots12,14,15]$ $\|\|(X_R^{i+18})[5,8,9,14,15]$	T_{3,S_3^1}
3	$\mathcal{V}_j\|\|(X_L^{i+19})[0\dots4,6,7,10\dots13]$ $\|\|(X_R^{i+19})[0\dots6,8\dots12,14,15]$ $\|\|(X_R^{i+18})[5,8,9,14,15]$	$(K^{i+19})[1,10,11]$	$\mathcal{V}_j\|\|(X_L^{i+19})[0,2\dots4,6,7,12,13]$ $\|\|(X_R^{i+19})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})[1,5,8,9,10,11,14,15]$	T_{4,S_4^1}
4	$\mathcal{V}_j\|\|(X_L^{i+19})[0,2\dots4,6,7,12,13]$ $\|\|(X_R^{i+19})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})[1,5,8,9,10,11,14,15]$	$(K^{i+19})[12]$	$\mathcal{V}_j\|\|(X_L^{i+19})[0,2\dots4,6,7,13]$ $\|\|(X_R^{i+19})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})[1,5,8,9,10,11,12,14,15]$	T_{5,S_5^1}
5	$\mathcal{V}_j\|\|(X_L^{i+19})[0,2\dots4,6,7,13]$ $\|\|(X_R^{i+19})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})[1,5,8,9,10,11,12,14,15]$	$(K^{i+19})[13]$	$\mathcal{V}_j\|\|(X_L^{i+19})[0,2\dots4,6,7]$ $\|\|(X_R^{i+19})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})[1,5,8,9,10,11,12,13,14,15]$	T_{6,S_6^1}
6	$\mathcal{V}_j\|\|(X_L^{i+19})[0,2\dots4,6,7,12,13]$ $\|\|(X_R^{i+19})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})[1,5,8,9,10,11,14,15]$	$(K^{i+19})[0,2,3,4,6,7]$	$\mathcal{V}_j\|\|(X_L^{i+18})[0\dots6,8,10\dots12,14,15]$ $\|\|(X_R^{i+18})$	T_{7,S_7^1}

substep 0: $2^{23} \times 2^{30.59} \times 5/24 = 2^{51.33}$
substep 1: $2^{23} \times 2^{33} \times 2^3 \times 3 \times 1/(16 \times 24) = 2^{52}$
substep 2: $2^{23} \times 2^{32} \times 2^4 \times 2 \times 1/(16 \times 24) = 2^{51.42}$
substep 3: $2^{23} \times 2^{31} \times 2^{6.5} \times 3 \times 1/(16 \times 24) = 2^{53.5}$
substep 4: $2^{23} \times 2^{30} \times 2^{7.5} \times 1/(16 \times 24) = 2^{51.92}$
substep 5: $2^{23} \times 2^{30} \times 2^{8.5} \times 1/(16 \times 24) = 2^{52.92}$
substep 6: $2^{23} \times 2^{30} \times 2^{14} \times 6 \times 1/(16 \times 24) = 2^{61}$

Table 4. Step 2 of key recovery attack on SIMON-32/64

i	Input (S_i^2)	Guessed subkey bit	Output (S_{i+1}^2)	Counter of S_{i+1}^2
0	$\mathcal{V}_j \| (X_L^{i+18})[0 \dots 6, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+18})$	$(K^{i+18})[6]$	$\mathcal{V}_j \| (X_L^{i+18})[0 \dots 5, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 15]$ $\| (X_R^{i+17})[6]$	X_{1,S_1^2}
1	$\mathcal{V}_j \| (X_L^{i+18})[0 \dots 5, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 15]$ $\| (X_R^{i+17})[6]$	$(K^{i+18})[15]$	$\mathcal{V}_j \| (X_L^{i+18})[0 \dots 5, 8, 10 \dots 12, 14]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 13, 15]$ $\| (X_R^{i+17})[6, 15]$	X_{2,S_2^2}
2	$\mathcal{V}_j \| (X_L^{i+18})[0 \dots 5, 8, 10 \dots 12, 14]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 13, 15]$ $\| (X_R^{i+17})[6, 15]$	$(K^{i+18})[1]$	$\mathcal{V}_j \| (X_L^{i+18})[0, 2 \dots 5, 8, 10 \dots 12, 14]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 13]$ $\| (X_R^{i+17})[1, 6, 15]$	X_{3,S_3^2}
3	$\mathcal{V}_j \| (X_L^{i+18})[0, 2 \dots 5, 8, 10 \dots 12, 14]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 13]$ $\| (X_R^{i+17})[1, 6, 15]$	$(K^{i+18})[3, 12]$	$\mathcal{V}_j \| (X_L^{i+18})[0, 2, 4, 5, 8, 10, 11, 14]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 10, 12, 13]$ $\| (X_R^{i+17})[1, 3, 6, 12, 15]$	X_{4,S_4^2}
4	$\mathcal{V}_j \| (X_L^{i+18})[0, 2, 4, 5, 8, 10, 11, 14]$ $\| (X_R^{i+18})[0 \dots 4, 6 \dots 10, 12, 13]$ $\| (X_R^{i+17})[1, 3, 6, 12, 15]$	$(K^{i+18})[2]$	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8, 10, 11, 14]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 10, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 12, 15]$	X_{5,S_5^2}
5	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8, 10, 11, 14]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 10, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 12, 15]$	$(K^{i+18})[11]$	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8, 10, 14]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 9, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 11, 12, 15]$	X_{6,S_6^2}
6	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8, 10, 14]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 9, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 11, 12, 15]$	$(K^{i+18})[10]$	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8, 14]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 8, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 10, 11, 12, 15]$	X_{7,S_7^2}
7	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8, 14]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 8, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 10, 11, 12, 15]$	$(K^{i+18})[14]$	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 8, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 10, 11, 12, 14, 15]$	X_{8,S_8^2}
8	$\mathcal{V}_j \| (X_L^{i+18})[0, 4, 5, 8]$ $\| (X_R^{i+18})[0, 2 \dots 4, 6 \dots 8, 12, 13]$ $\| (X_R^{i+17})[1, 2, 3, 6, 10, 11, 12, 14, 15]$	$(K^{i+18})[4, 5, 8]$	$\mathcal{V}_j \| (X_L^{i+17})[0, 2 \dots 4, 6, 7, 12, 13]$ $\| (X_R^{i+17})[0' \dots 6, 8, 10 \dots 12, 14, 15]$ where $(X_R^{i+17})[0'] = (X_R^{i+17})[0] \oplus (K^{i+18})[0]$	X_{9,S_9^2}

substep 0: $2^{23} \times 2^{14} \times 2^{30} \times 2^{0.5} \times 1/16 \times 24 = 2^{58.92}$
substep 1: $2^{23} \times 2^{14} \times 2^{29} \times 2 \times 1/16 \times 24 = 2^{58.42}$
substep 2: $2^{23} \times 2^{14} \times 2^{28} \times 2^2 \times 1/16 \times 24 = 2^{58.42}$
substep 3: $2^{23} \times 2^{14} \times 2^{27} \times 2^3 \times 2 \times 1/16 \times 24 = 2^{58.42}$
substep 4: $2^{23} \times 2^{14} \times 2^{26} \times 2^4 \times 1/16 \times 24 = 2^{58.42}$
substep 5: $2^{23} \times 2^{14} \times 2^{25} \times 2^5 \times 1/16 \times 24 = 2^{58.42}$
substep 6: $2^{23} \times 2^{14} \times 2^{24} \times 2^6 \times 1/16 \times 24 = 2^{58.42}$
substep 7: $2^{23} \times 2^{14} \times 2^{23} \times 2^7 \times 1/16 \times 24 = 2^{58.42}$
substep 8: $2^{23} \times 2^{14} \times 2^{23} \times 2^{9.5} \times 3 \times 1/16 \times 24 = 2^{62.5}$

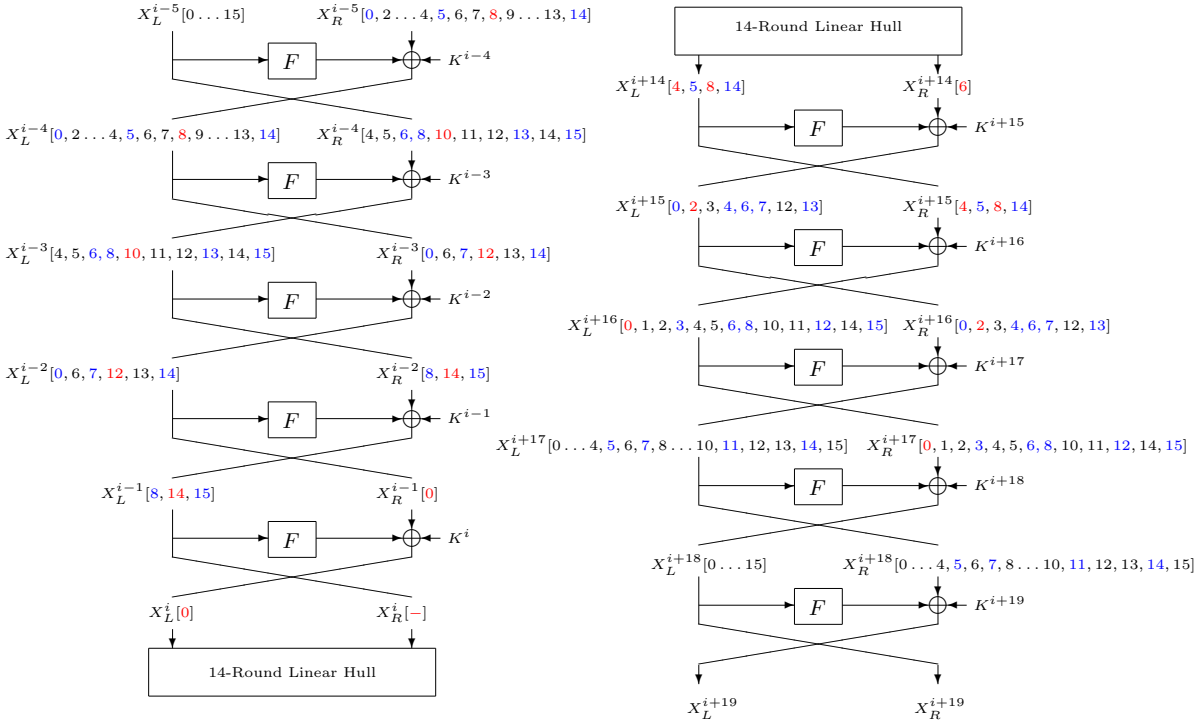

Fig. 1. Adding some rounds to the 14-round linear hull for SIMON-32/K.

Table 5. Step 3 of key recovery attack on SIMON-32/64

i	Input (S_i^3)	Guessed subkey bit	Output (S_{i+1}^3)	Counter of S_{i+1}^3
0	$\mathcal{V}_j \| (X_L^{i+17})[0, 2 \dots 4, 6, 7, 12, 13]$ $\| (X_R^{i+17})[0 \dots 6, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+17})[0 \dots 2, 4 \dots 6, 8, 10 \dots 12, 14, 15]$	$(K^{i+17})[4]$	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 3, 6, 7, 12, 13]$ $\| (X_R^{i+17})[0 \dots 2, 4 \dots 6, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+16})[4]$	Y_{1,S_1^3}
1	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 3, 6, 7, 12, 13]$ $\| (X_R^{i+17})[0 \dots 2, 4 \dots 6, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+16})[4]$	$(K^{i+17})[3]$	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6, 7, 12, 13]$ $\| (X_R^{i+17})[0, 4 \dots 6, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+16})[3, 4]$	Y_{2,S_2^3}
2	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6, 7, 12, 13]$ $\| (X_R^{i+17})[0, 4 \dots 6, 8, 10 \dots 12, 14, 15]$ $\| (X_R^{i+16})[3, 4]$	$(K^{i+17})[12]$	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6, 7, 13]$ $\| (X_R^{i+17})[0, 4 \dots 6, 8, 11, 12, 14, 15]$ $\| (X_R^{i+16})[3, 4, 12]$	Y_{3,S_3^3}
3	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6, 7, 13]$ $\| (X_R^{i+17})[0, 4 \dots 6, 8, 11, 12, 14, 15]$ $\| (X_R^{i+16})[3, 4, 12]$	$(K^{i+17})[13]$	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6, 7]$ $\| (X_R^{i+17})[0, 4 \dots 6, 8, 14, 15]$ $\| (X_R^{i+16})[3, 4, 12, 13]$	Y_{4,S_4^3}
4	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6, 7]$ $\| (X_R^{i+17})[0, 4 \dots 6, 8, 14, 15]$ $\| (X_R^{i+16})[3, 4, 12, 13]$	$(K^{i+17})[7]$	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6]$ $\| (X_R^{i+17})[0, 4, 5, 8, 14, 15]$ $\| (X_R^{i+16})[3, 4, 7, 12, 13]$	Y_{5,S_5^3}
5	$\mathcal{V}_j \| (X_L^{i+17})[0, 2, 6]$ $\| (X_R^{i+17})[0, 4, 5, 8, 14, 15]$ $\| (X_R^{i+16})[3, 4, 7, 12, 13]$	$(K^{i+17})[0, 6]$	$\mathcal{V}_j \| (X_L^{i+16})[4, 5, 8, 14]$ $\| (X_R^{i+16})[0, 2', 3, 4, 6, 7, 12, 13]$ where $(X_R^{i+16})[2'] = (X_R^{i+16})[2] \oplus (K^{i+17})[2]$	Y_{6,S_6^3}

substep 0: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{22} \times 2^{0.5} \times 1/(16 \times 24) = 2^{60.42}$
substep 1: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{21} \times 2^{1.5} \times 1/(16 \times 24) = 2^{60.42}$
substep 2: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{19} \times 2^{2.5} \times 1/(16 \times 24) = 2^{59.42}$
substep 3: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{18} \times 2^{3} \times 1/(16 \times 24) = 2^{58.92}$
substep 4: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{16} \times 2^{3.5} \times 1/(16 \times 24) = 2^{57.42}$
substep 5: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{15} \times 2^{4.5} \times 2 \times 1/(16 \times 24) = 2^{58.42}$

Table 6. Step 4 of key recovery attack on SIMON-32/64

i	Input (S_i^4)	Guessed subkey bit	Output (S_{i+1}^4)	Counter of S_{i+1}^4
0	$\mathcal{V}_j \| (X_L^{i+16})[4,5,8,14] \| (X_R^{i+16})[0,2,3,4,6,7,12,13]$	$(K^{i+16})[5,14]$	$\mathcal{V}_j \| (X_L^{i+15})[6] \| (X_R^{i+15})[4',5,8',14]$ where $(X_R^{i+15})[4'] = (X_R^{i+15})[4] \oplus (K^{i+16})[4]$ and $(X_R^{i+15})[8'] = (X_R^{i+15})[8] \oplus (K^{i+16})[8]$	Z_{1,S_1^4}

substep 0: $2^{23} \times 2^{14} \times 2^{9.5} \times 2^{4.5} \times 2^{13} \times 2 \times 2 \times 1/(16 \times 24) = 2^{57.42}$

Table 7. Step 5 of key recovery attack on SIMON-32/64

i	Input (S_i^5)	Guessed subkey bit	Output (S_{i+1}^5)	Counter of S_{i+1}^5
0	$\mathcal{V}_j \| (X_L^{i+15})[6] \| (X_R^{i+15})[4,5,8,14]$		$\mathcal{V}_j \| (X_L^{i+14})[4',8'] \| (X_R^{i+14})[6']$ where $(X_R^{i+14})[6'] = (X_R^{i+14})[6] \oplus (K^{i+15})[6]$, $(X_L^{i+14})[4'] = (X_L^{i+14})[4] \oplus (K^{i+16})[4] \oplus (K^{i+17})[2] \oplus (K^{i+18})[0]$, and $(X_L^{i+14})[8'] = (X_L^{i+14})[8] \oplus (K^{i+16})[8]$.	W_{1,S_1^5}

B Steps of the Key Recovery Attack on SIMON-48/96

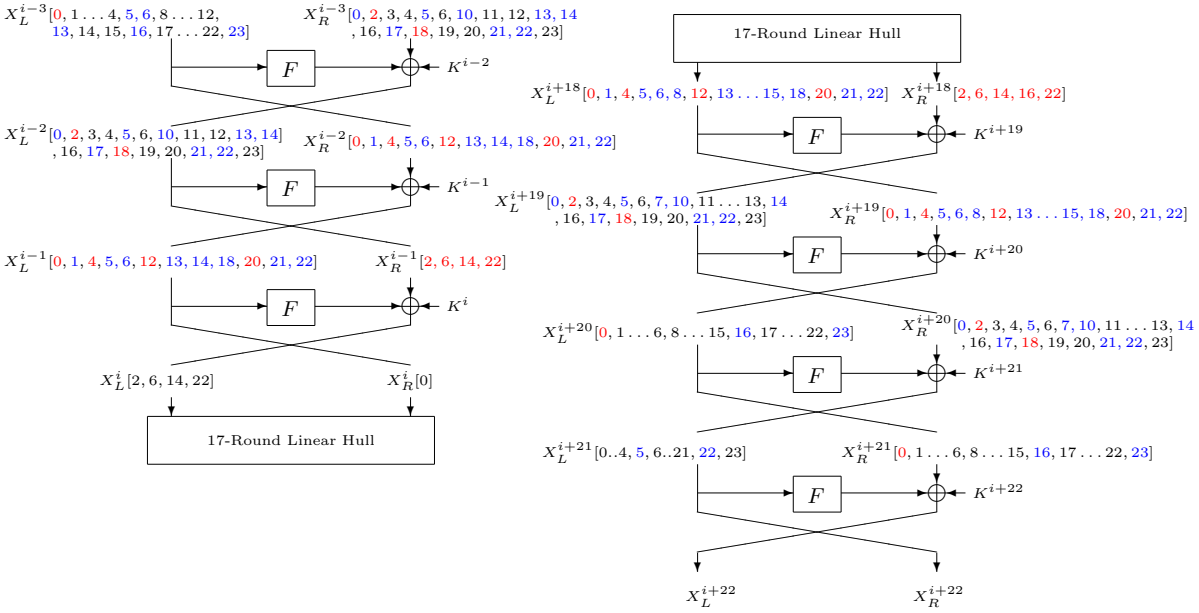

Fig. 2. Adding some rounds to the 17-round linear hull for SIMON-48/96.

$$\mathcal{V}_j = (X_L^i)[2,6,14,22] \oplus (X_R^i)[0] \oplus (K^i)[2,6,14,22] \oplus (K^{i-1})[0,4,12,20] \oplus (K^{i-2})[2,18]$$
$$\mathcal{W}_j = (X_L^{i+18})[0] \oplus (X_R^{i+18})[2,6,14,16,22] \oplus (K^{i+19})[2,6,14,16,22] \oplus (K^{i+20})[0,4,12,20] \oplus (K^{i+21})[2,18] \oplus (K^{i+22})[0]$$

Table 8. Step 1 of key recovery attack on SIMON-48/96

i	Input (S_i^1)	Guessed subkey bit	Output (S_{i+1}^1)	Counter of S_{i+1}^1
0	$(X_L^{i-3})[0\dots6,8\dots23]$ $\|(X_R^{i-3})[0,2\dots6,10\dots14,16\dots23]$	$(K^{i-2})[0,3,4,5,6,10,11,12,13,14,16,17,19,$ $20,21,22,23]\|(K^{i-1})[1,5,6,13,14,18,21,22]$	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8\dots23]$ $\|(X_R^{i+22})$	T_{1,S_1^1}
1	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8\dots23]$ $\|(X_R^{i+22})$	$(K^{i+22})[10,11,17]$	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8,9,12\dots16,18\dots23]$ $\|(X_R^{i+22})[0\dots8,10\dots23]$ $\|(X_R^{i+21})[10,11,17]$	T_{2,S_2^1}
2	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8,9,12\dots16,18\dots23]$ $\|(X_R^{i+22})[0\dots8,10\dots23]$ $\|(X_R^{i+21})[10,11,17]$	$(K^{i+22})[16,23]$	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8,9,12\dots15,18\dots22]$ $\|(X_R^{i+22})[0\dots8,10\dots14,16\dots23]$ $\|(X_R^{i+21})[10,11,16,17,23]$	T_{3,S_3^1}
3	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8,9,12\dots15,18\dots22]$ $\|(X_R^{i+22})[0\dots8,10\dots14,16\dots23]$ $\|(X_R^{i+21})[10,11,16,17,23]$	$(K^{i+22})[9]$	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+22})[0\dots7,10\dots14,16\dots23]$ $\|(X_R^{i+21})[8,10,11,16,17,23]$	T_{4,S_4^1}
4	$\mathcal{V}_j\|(X_L^{i+22})[0\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+22})[0\dots7,10\dots14,16\dots23]$ $\|(X_R^{i+21})[8,10,11,16,17,23]$	$(K^{i+22})[2,3]$	$\mathcal{V}_j\|(X_L^{i+22})[0,1,4\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+22})[0,2\dots7,10\dots14,16\dots23]$ $\|(X_R^{i+21})[2,3,8,10,11,16,17,23]$	T_{5,S_5^1}
5	$\mathcal{V}_j\|(X_L^{i+22})[0,1,4\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+22})[0,2\dots7,10\dots14,16\dots23]$ $\|(X_R^{i+21})[2,3,8,10,11,16,17,23]$	$(K^{i+22})[1,4\dots6,8,12\dots15,18\dots22]$	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10\dots14,16\dots23$ $\|(X_R^{i+21})[0\dots6,8\dots23]$	T_{6,S_6^1}

substep 0: $2^{17}\times 2^{47.78}\times 3/24 = 2^{61.78}$
substep 1: $2^{17}\times 2^{48}\times 2^{3}\times 3\times 1/(24\times 24) = 2^{60.41}$
substep 2: $2^{17}\times 2^{47}\times 2^{4}\times 2\times 1/(24\times 24) = 2^{59.83}$
substep 3: $2^{17}\times 2^{46}\times 2^{5}\times 1/(24\times 24) = 2^{58.83}$
substep 4: $2^{17}\times 2^{45}\times 2^{7}\times 2\times 1/(24\times 24) = 2^{60.83}$
substep 5: $2^{17}\times 2^{44}\times 2^{21}\times 14\times 1/(24\times 24) = 2^{76.64}$

Table 9. Step 2 of key recovery attack on SIMON-48/96

i	Input (S_i^2)	Guessed subkey bit	Output (S_{i+1}^2)	Counter of S_{i+1}^2
0	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10\dots14,16\dots23$ $\|(X_R^{i+21})[0\dots6,8\dots23]$	$(K^{i+21})[19]$	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10\dots14,16\dots18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8\dots16,18\dots23]$ $\|(X_R^{i+20})[19]$	X_{1,S_1^2}
1	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10\dots14,16\dots18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8\dots16,18\dots23]$ $\|(X_R^{i+20})[19]$	$(K^{i+21})[12,13]$	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10,11,14,16\dots18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8\dots10,12\dots16,18\dots23]$ $\|(X_R^{i+20})[12,13,19]$	X_{2,S_2^2}
2	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10,11,14,16\dots18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8\dots10,12\dots16,18\dots23]$ $\|(X_R^{i+20})[12,13,19]$	$(K^{i+21})[11]$	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10,14,16\dots18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,9,12\dots16,18\dots23]$ $\|(X_R^{i+20})[11\dots13,19]$	X_{3,S_3^2}
3	$\mathcal{V}_j\|(X_L^{i+21})[0,2\dots7,10,14,16\dots18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,9,12\dots16,18\dots23]$ $\|(X_R^{i+20})[11\dots13,19]$	$(K^{i+21})[0,17]$	$\mathcal{V}_j\|(X_L^{i+21})[2\dots7,10,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,9,12\dots15,18\dots23]$ $\|(X_R^{i+20})[0,11\dots13,17,19]$	X_{4,S_4^2}
4	$\mathcal{V}_j\|(X_L^{i+21})[2\dots7,10,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,9,12\dots15,18\dots23]$ $\|(X_R^{i+20})[0,11\dots13,17,19]$	$(K^{i+21})[7]$	$\mathcal{V}_j\|(X_L^{i+21})[2\dots6,10,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,9,12\dots15,18\dots22]$ $\|(X_R^{i+20})[0,7,11\dots13,17,19]$	X_{5,S_5^2}
5	$\mathcal{V}_j\|(X_L^{i+21})[2\dots6,10,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,9,12\dots15,18\dots22]$ $\|(X_R^{i+20})[0,7,11\dots13,17,19]$	$(K^{i+21})[10]$	$\mathcal{V}_j\|(X_L^{i+21})[2\dots6,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+20})[0,7,10,11\dots13,17,19]$	X_{6,S_6^2}
6	$\mathcal{V}_j\|(X_L^{i+21})[2\dots6,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+20})[0,7,10,11\dots13,17,19]$	$(K^{i+21})[4,5]$	$\mathcal{V}_j\|(X_L^{i+21})[2,3,6,14,16,18,20\dots23$ $\|(X_R^{i+21})[0\dots2,4\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+20})[0,4,5,7,10,11\dots13,17,19]$	X_{7,S_7^2}
7	$\mathcal{V}_j\|(X_L^{i+21})[2,3,6,14,16,18,20\dots23]$ $\|(X_R^{i+21})[0\dots2,4\dots6,8,12\dots15,18\dots22]$ $\|(X_R^{i+20})[0,4,5,7,10,11\dots13,17,19]$	$(K^{i+21})[3,6,14,16,20,21,22,23]$	$\mathcal{V}_j\|(X_L^{i+20})[0,1,4\dots6,8,12\dots15,18,20\dots22]$ $\|(X_R^{i+20})[0,2\dots7,10\dots14,16\dots23]$	X_{8,S_8^2}

substep 0: $2^{17}\times 2^{21}\times 2^{44}\times 2\times 1/24.24 = 2^{73.83}$
substep 1: $2^{17}\times 2^{21}\times 2^{43}\times 2^{3}\times 2\times 1/24.24 = 2^{75.83}$
substep 2: $2^{17}\times 2^{21}\times 2^{42}\times 2^{4}\times 1/24.24 = 2^{74.83}$
substep 3: $2^{17}\times 2^{21}\times 2^{41}\times 2^{5}\times 2\times 1/24.24 = 2^{75.83}$
substep 4: $2^{17}\times 2^{21}\times 2^{40}\times 2^{5.5}\times 1/24.24 = 2^{74.33}$
substep 5: $2^{17}\times 2^{21}\times 2^{39}\times 2^{6}\times 1/24.24 = 2^{73.83}$
substep 6: $2^{17}\times 2^{21}\times 2^{38}\times 2^{7.5}\times 2\times 1/24.24 = 2^{75.33}$
substep 7: $2^{17}\times 2^{21}\times 2^{37}\times 2^{14}\times 8\times 1/24.24 = 2^{82.83}$

Table 10. Step 3 of key recovery attack on SIMON-48/96

i	Input (S_i^3)	Guessed subkey bit	Output (S_{i+1}^3)	Counter of S_{i+1}^3
0	$\mathcal{V}_j \| (X_L^{i+20})[0,1,4\dots6,8,12\dots15,18,20\dots22]$ $\| (X_R^{i+20})[0,2\dots7,10\dots14,16\dots23]$	$(K^{i+20})[1]$	$\mathcal{V}_j \| (X_L^{i+20})[0,4\dots6,8,12\dots15,18,20\dots22]$ $\| (X_R^{i+20})[0,2\dots7,10\dots14,16\dots22]$ $\| (X_R^{i+19})[1]$	Y_{1,S_1^3}
1	$\mathcal{V}_j \| (X_L^{i+20})[0,4\dots6,8,12\dots15,18,20\dots22]$ $\| (X_R^{i+20})[0,2\dots7,10\dots14,16\dots22]$ $\| (X_R^{i+19})[1]$	$(K^{i+20})[5]$	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12\dots15,18,20\dots22]$ $\| (X_R^{i+20})[0,2,4\dots7,10\dots14,16\dots22]$ $\| (X_R^{i+19})[1,5]$	Y_{2,S_2^3}
2	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12\dots15,18,20\dots22]$ $\| (X_R^{i+20})[0,2,4\dots7,10\dots14,16\dots22]$ $\| (X_R^{i+19})[1,5]$	$(K^{i+20})[13]$	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,14,15,18,20\dots22]$ $\| (X_R^{i+20})[0,2,4\dots7,10,12\dots14,16\dots22]$ $\| (X_R^{i+19})[1,5,13]$	Y_{3,S_3^3}
3	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,14,15,18,20\dots22]$ $\| (X_R^{i+20})[0,2,4\dots7,10,12\dots14,16\dots22]$ $\| (X_R^{i+19})[1,5,13]$	$(K^{i+20})[14]$	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,15,18,20\dots22]$ $\| (X_R^{i+20})[0,2,4\dots7,10,13,14,16\dots22]$ $\| (X_R^{i+19})[1,5,13,14]$	Y_{4,S_4^3}
4	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,15,18,20\dots22]$ $\| (X_R^{i+20})[0,2,4\dots7,10,13,14,16\dots22]$ $\| (X_R^{i+19})[1,5,13,14]$	$(K^{i+20})[21]$	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,15,18,20,22]$ $\| (X_R^{i+20})[0,2,4\dots7,10,13,14,16\dots18,20\dots22]$ $\| (X_R^{i+19})[1,5,13,14,21]$	Y_{5,S_5^3}
5	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,15,18,20,22]$ $\| (X_R^{i+20})[0,2,4\dots7,10,13,14,16\dots18,20\dots22]$ $\| (X_R^{i+19})[1,5,13,14,21]$	$(K^{i+20})[22]$	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,15,18,20]$ $\| (X_R^{i+20})[0,2,4\dots7,10,13,14,16\dots18,20,22]$ $\| (X_R^{i+19})[1,5,13,14,21,22]$	Y_{6,S_6^3}
6	$\mathcal{V}_j \| (X_L^{i+20})[0,4,6,8,12,15,18,20]$ $\| (X_R^{i+20})[0,2,4\dots7,10,13,14,16\dots18,20,22]$ $\| (X_R^{i+19})[1,5,13,14,21,22]$	$(K^{i+20})[6,8,15,18]$	$\mathcal{V}_j \| (X_L^{i+19})[2,6,14,16,22]$ $\| (X_R^{i+19})[0,1,4\dots6,8,12\dots15,18,20\dots22]$	Y_{7,S_7^3}
7	$\mathcal{V}_j \| (X_L^{i+19})[2,6,14,16,22]$ $\| (X_R^{i+19})[0,1,4\dots6,8,12\dots15,18,20\dots22]$		$\mathcal{V}_j \| (X_L^{i+18})[0]$ $\| (X_R^{i+18})[2,6,14,16,22]$	Y_{7,S_7^3}

substep 0: $2^{17} \times 2^{21} \times 2^{14} \times 2^{35} \times 2^{0.5} \times 1/(24 \times 24) = 2^{78.33}$
substep 1: $2^{17} \times 2^{21} \times 2^{14} \times 2^{34} \times 2^{1} \times 1/(24 \times 24) = 2^{77.83}$
substep 2: $2^{17} \times 2^{21} \times 2^{14} \times 2^{33} \times 2^{1.5} \times 1/(24 \times 24) = 2^{77.33}$
substep 3: $2^{17} \times 2^{21} \times 2^{14} \times 2^{32} \times 2^{2} \times 1/(24 \times 24) = 2^{77.83}$
substep 4: $2^{17} \times 2^{21} \times 2^{14} \times 2^{31} \times 2^{2.5} \times 1/(24 \times 24) = 2^{76.33}$
substep 5: $2^{17} \times 2^{21} \times 2^{14} \times 2^{30} \times 2^{3} \times 1/(24 \times 24) = 2^{75.83}$
substep 6: $2^{17} \times 2^{21} \times 2^{14} \times 2^{29} \times 2^{5} \times 4 \times 1/(24 \times 24) = 2^{76.83}$

C MIP Experiments

Table 11 shows the 30 sub approximations that have been used to estimate the squared correlations of the lower class trails. The experiments where the MIP solutions are limited to 512 trails per approximation took exactly 70125.382718 seconds which is less than 20 hrs using a standard laptop.

Table 12 shows the 30 sub approximations that have been used to estimate the squared correlations of the upper class trails. The experiments where the MIP solutions are limited to 512 trails per approximation took exactly 62520.033249 seconds which is less than 18 hrs using a standard laptop.

Table 11. Lower Class Trails found through our time-memory trade-off method, $c_{i1}^2 \equiv$ the squared correlation of the ith 11-round linear approximation with light trails found through the correlation matrix, $c_{i2}^2 \equiv$ the squared correlation of the ith 6-round linear approximation with heavy trails found through the MIP method, $c_{i1}^2 c_{i2}^2 \equiv$ is the squared correlation of the ith 17-round linear approximation and $\sum c_{i1}^2 c_{i2}^2$ is the total estimated squared correlation of the lower class trails of our 17-round linear hull after including $i \leq 30$ linear approximations

i	Matrix trails	$\log_2 c_{i1}^2$	MIP trails	$\log_2 c_{i2}^2$	$\log_2 \sum c_{i1}^2 c_{i2}^2$
1	$404044000001 \xrightarrow{11-round} 001000004400$	-28.6806	$001000004400 \xrightarrow{6-round} 000001414044$	-22.342570	-51.023180
2	$404044000001 \xrightarrow{11-round} 001000004410$	-28.6806	$001000004410 \xrightarrow{6-round} 000001414044$	-24.339670	-50.700671
3	$404044000001 \xrightarrow{11-round} 001000004C00$	-28.6806	$001000004C00 \xrightarrow{6-round} 000001414044$	-24.486365	-50.460718
4	$404044000001 \xrightarrow{11-round} 001000004C10$	-28.6806	$001000004C10 \xrightarrow{6-round} 000001414044$	-23.979129	-50.176458
5	$404044000001 \xrightarrow{11-round} 003000004400$	-30.6806	$003000004400 \xrightarrow{6-round} 000001414044$	-22.342570	-49.988669
6	$404044000001 \xrightarrow{11-round} 003000004410$	-30.6806	$003000004410 \xrightarrow{6-round} 000001414044$	-24.339586	-49.945219
7	$404044000001 \xrightarrow{11-round} 003000004420$	-30.6806	$003000004420 \xrightarrow{6-round} 000001414044$	-27.953899	-49.941728
8	$404044000001 \xrightarrow{11-round} 003000004430$	-30.6806	$003000004430 \xrightarrow{6-round} 000001414044$	-26.956545	-49.934784
9	$404044000001 \xrightarrow{11-round} 003000004C00$	-30.6806	$003000004C00 \xrightarrow{6-round} 000001414044$	-24.486642	-49.896909
10	$404044000001 \xrightarrow{11-round} 003000004C10$	-30.6806	$003000004C00 \xrightarrow{6-round} 000001414044$	-24.486642	-49.844727
11	$404044000001 \xrightarrow{11-round} 003000004C20$	-30.6806	$003000004C20 \xrightarrow{6-round} 000001414044$	-26.880410	-49.837883
12	$404044000001 \xrightarrow{11-round} 003000005400$	-30.6806	$003000005400 \xrightarrow{6-round} 000001414044$	-31.046525	-49.837503
13	$404044000001 \xrightarrow{11-round} 003000005410$	-30.6806	$003000005410 \xrightarrow{6-round} 000001414044$	-32.568502	-49.837371
14	$404044000001 \xrightarrow{11-round} 003000005420$	-30.6806	$003000005420 \xrightarrow{6-round} 000001414044$	-31.189830	-49.837026
15	$404044000001 \xrightarrow{11-round} 003000005C00$	-30.6806	$003000005C00 \xrightarrow{6-round} 000001414044$	-27.773381	-49.833356
16	$404044000001 \xrightarrow{11-round} 001040004400$	-30.6806	$001040004400 \xrightarrow{6-round} 000001414044$	-22.342570	-49.683331
17	$404044000001 \xrightarrow{11-round} 001040004410$	-30.6806	$001040004410 \xrightarrow{6-round} 000001414044$	-24.339586	-49.648069
18	$404044000001 \xrightarrow{11-round} 001040004420$	-30.6806	$001040004420 \xrightarrow{6-round} 000001414044$	-27.954667	-49.645229
19	$404044000001 \xrightarrow{11-round} 001040004430$	-30.6806	$001040004430 \xrightarrow{6-round} 000001414044$	-26.957186	-49.639576
20	$404044000001 \xrightarrow{11-round} 001040004C00$	-30.6806	$001040004C00 \xrightarrow{6-round} 000001414044$	-24.486272	-49.608628
21	$404044000001 \xrightarrow{11-round} 001040004C10$	-30.6806	$001040004C10 \xrightarrow{6-round} 000001414044$	-23.979129	-49.565757
22	$404044000001 \xrightarrow{11-round} 001040004C20$	-30.6806	$001040004C20 \xrightarrow{6-round} 000001414044$	-26.879560	-49.560110
23	$404044000001 \xrightarrow{11-round} 001040404400$	-30.6806	$001040404400 \xrightarrow{6-round} 000001414044$	-30.596588	-49.559682
24	$404044000001 \xrightarrow{11-round} 001040404410$	-30.6806	$001040404410 \xrightarrow{6-round} 000001414044$	-27.765884	-49.556637
25	$404044000001 \xrightarrow{11-round} 001040404420$	-30.6806	$001040404420 \xrightarrow{6-round} 000001414044$	-30.819304	-49.556271
26	$404044000001 \xrightarrow{11-round} 001040404C00$	-30.6806	$001040404C00 \xrightarrow{6-round} 000001414044$	-32.191224	-49.556130
27	$404044000001 \xrightarrow{11-round} 003040004400$	-30.6806	$003040004400 \xrightarrow{6-round} 000001414044$	-22.342570	-49.431232
28	$404044000001 \xrightarrow{11-round} 003040004410$	-30.6806	$003040004410 \xrightarrow{6-round} 000001414044$	-24.339753	-49.401570
29	$404044000001 \xrightarrow{11-round} 003040004420$	-30.6806	$003040004420 \xrightarrow{6-round} 000001414044$	-27.954411	-49.399175
30	$404044000001 \xrightarrow{11-round} 003040004C00$	-30.6806	$003040004C00 \xrightarrow{6-round} 000001414044$	-24.486457	-49.372938

Table 12. Upper Class Trails found through our time-memory trade-off method, $c_{i1}^2 \equiv$ the squared correlation of the ith 6-round linear approximation with heavy trails found through the MIP method, $c_{i2}^2 \equiv$ the squared correlation of the ith 6-round linear approximation with light trails found through the correlation matrix , $c_{i1}^2 c_{i2}^2 \equiv$ is the squared correlation of the ith 17-round linear approximation and $\sum c_{i1}^2 c_{i2}^2$ is the total estimated squared correlation of the upper class trails of our 17-round linear hull after including $i \leq 30$ linear approximations

i	MIP trails	$\log_2 c_{i1}^2$	Matrix trails	$\log_2 c_{i2}^2$	$\log_2 \sum c_{i1}^2 c_{i2}^2$
1	$404044000001 \xrightarrow{6-round} 004400001000$	-22.342570	$004400001000 \xrightarrow{11-round} 000001414044$	-28.6806	-51.023180
2	$404044000001 \xrightarrow{6-round} 004410001000$	-24.339670	$004410001000 \xrightarrow{11-round} 000001414044$	28.6806	-50.700671
3	$404044000001 \xrightarrow{6-round} 004C00001000$	-24.486272	$004C00001000 \xrightarrow{11-round} 000001414044$	-28.6806	-50.460704
4	$404044000001 \xrightarrow{6-round} 004C10001000$	-23.979129	$004C10001000 \xrightarrow{11-round} 000001414044$	-28.6806	-50.176447
5	$404044000001 \xrightarrow{6-round} 004400003000$	-22.342570	$004400003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.988659
6	$404044000001 \xrightarrow{6-round} 004410003000$	-24.339753	$004410003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.945214
7	$404044000001 \xrightarrow{6-round} 004420003000$	-27.955435	$004420003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.941726
8	$404044000001 \xrightarrow{6-round} 004430003000$	-26.956674	$004430003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.934783
9	$404044000001 \xrightarrow{6-round} 004C00003000$	-24.486272	$004C00003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.896899
10	$404044000001 \xrightarrow{6-round} 004C10003000$	-23.979129	$004C10003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.844713
11	$404044000001 \xrightarrow{6-round} 004C20003000$	-26.879317	$004C20003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.837864
12	$404044000001 \xrightarrow{6-round} 005400003000$	-31.046525	$005400003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.837483
13	$404044000001 \xrightarrow{6-round} 005410003000$	-32.568502	$005410003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.837483
14	$404044000001 \xrightarrow{6-round} 005420003000$	-31.189830	$005420003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.837007
15	$404044000001 \xrightarrow{6-round} 005C00003000$	-27.77338	$005C00003000 \xrightarrow{11-round} 000001414044$	-30.6806	-49.833337
16	$404044000001 \xrightarrow{6-round} 004400001040$	-22.342570	$004400001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.683313
17	$404044000001 \xrightarrow{6-round} 004400003040$	-22.342570	$004400003040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.547431
18	$404044000001 \xrightarrow{6-round} 004410001040$	-24.339670	$004410001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.515307
19	$404044000001 \xrightarrow{6-round} 004410003040$	-24.339670	$004410003040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.483882
20	$404044000001 \xrightarrow{6-round} 004420001040$	-27.955691	$004420001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.481349
21	$404044000001 \xrightarrow{6-round} 004420003040$	-27.954155	$004420003040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.478817
22	$404044000001 \xrightarrow{6-round} 004430001040$	-26.956417	$004430001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.473776
23	$404044000001 \xrightarrow{6-round} 004C00001040$	-24.486457	$004C00001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.446160
24	$404044000001 \xrightarrow{6-round} 004C00003040$	-24.486550	$004C00003040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.419065
25	$404044000001 \xrightarrow{6-round} 004C10001040$	-23.979259	$004C10001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.381407
26	$404044000001 \xrightarrow{6-round} 004C20001040$	-26.879195	$004C20001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.376435
27	$404044000001 \xrightarrow{6-round} 404400001040$	-30.596588	$404400001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.376058
28	$404044000001 \xrightarrow{6-round} 404410001040$	-27.765898	$404410001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.373377
29	$404044000001 \xrightarrow{6-round} 404420001040$	-30.819304	$404420001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.373054
30	$04044000001 \xrightarrow{6-round} 404C00001040$	-32.191224	$404C00001040 \xrightarrow{11-round} 000001414044$	-30.6806	-49.372930