Differential-Linear Cryptanalysis of the Lightweight Cryptographic Algorithm KNOT

Shichang Wang[1,2], Meicheng Liu[1,2], Dongdai Lin[1,2] and Shiqi Hou[1,2]

[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
Email: {wangshichang,liumeicheng,ddlin,houshiqi}@iie.ac.cn

Abstract. KNOT is one of the 32 candidates in the second round of NIST's lightweight cryptography standardization process. The KNOT family consists of bit-slice lightweight Authenticated Encryption with Associated Data (AEAD) and hashing algorithms. In this paper, we evaluate the security for the initialization phase of two members of the KNOT-AEAD family by differential-linear cryptanalysis.
More exactly, we analyze KNOT-AEAD(128,256,64) and KNOT-AEAD (128,384,192) which have 128-bit secret keys. As a result, for 15-round KNOT-AEAD(128,256,64), our attack takes $2^{48.8}$ time complexity and $2^{47.5}$ blocks to recover the full 128-bit key. To the best of our knowledge, this is the first full key-recovery attack on 15-round KNOT-AEAD(128, 256,64), and it achieves three more rounds compared with the existing work. Regarding 17-round KNOT-AEAD(128,384,192), time complexity of $2^{59.2}$ and data complexity of $2^{58.2}$ are required to launch a key-recovery attack, which is five rounds better than the known result. We stress here that our attacks do not threaten the security of KNOT-AEAD.

Keywords: Differential-Linear cryptanalysis · Lightweight cryptography · KNOT.

1 Introduction

The National Institute of Standards and Technology (NIST) is selecting and standardizing lightweight authenticated encryption and hashing algorithms that are suitable for use in constrained environments (e.g. sensor networks, healthcare, the Internet of Things). Of the 56 Round 1 candidates, 32 algorithms are selected by NIST to Round 2, such as [30, 5, 4, 1, 15]. The designers of cryptographic primitives always use the minimum of active S-boxes as the indicator for resistance against differential or linear (hull) attacks. However, this ignores some powerful variants of differential and linear cryptanalyses, for example, differential-linear attacks. It is necessary to carefully evaluate the security against differential-linear attacks for the lightweight cryptographic algorithms.

Differential[12] and linear[25] attacks are two of the most fundamental techniques of cryptanalysis. It is usually very difficult to find some long enough

differentials and linear approximations for these ciphers which are well designed against differential and linear attacks. However, in some cases, with a short differential and linear approximation, an effective attack might be launched. In 1994, Langford and Hellman [22] firstly showed that a differential of E_0 and a highly biased linear approximation of E_1 could be combined into a distinguisher for the entire cipher E where E was divided into two subciphers E_0 and E_1 such that $E = E_1 \circ E_0$ by a technique called *differential-linear cryptanalysis*. In [10], Biham et al. extended and improved this technique to obtain wider scope of applications. The differential-linear technique was used to attack many cryptographic primitives, such as the block cipher Serpent [11, 19, 24] which was in the AES finalists, the lightweight authenticated encryption Ascon [18] that was primary choice in the final portfolio of the CAESAR competition, Chaskey [23, 6] and ChaCha [2, 16, 6, 17]. In 2017, Blondeau et al. [13] applied the link between differential and linear attacks and developed a concise theory of the differential-linear cryptanalysis. Then they gave an exact expression of the bias under the assumption that the two parts of the cipher were independent. Recently, Bar-on et al. [3], in EUROCRYPT 2019, presented the *Differential-Linear Connectivity Table (DLCT)* to take into account the dependency between two parts of the cipher, and showed that in many cases, the adversary could exploit it to launch more effective attacks. Besides, they derived that the DLCT could be constructed effectively using the Fast Fourier Transform. In CRYPTO 2020, Beierle et al. [6] presented several improvements in the context of the differential-linear attacks of ARX ciphers and successfully applied them to Chaskey and ChaCha. In ASIACRYPT 2020, Gutiérrez et al. [20] proposed the differential-linear cryptanalysis reached 17 rounds of Gimli which was a cryptographic permutation published at CHES 2017 [8]. Gimli is also the core primitive of a submission to NIST lightweight cryptography project which is one of the 32 candidates in the second round.

KNOT [30] is designed by Zhang et al., which is a family of bit-slice lightweight Authenticated Encryption with Associated Data (AEAD) and hashing algorithms. This family is based on the KNOT permutations which iteratively apply an SPN round transformation. There are both four members in KNOT-AEAD and KNOT-Hash and both of their primary members have a state of 256 bits. In 2019, KNOT was selected by NIST as one of the 32 candidates in the second round of lightweight cryptography (LWC) standardization process. In the specification of KNOT [30], the designers evaluated the security of KNOT permutation against various attacks, such as (impossible) differential, linear, division cryptanalysis etc. Later, they [31] further updated the results of security analysis of KNOT-AEAD and KNOT-Hash. As far as we know, there is no third-party security analysis yet.

1.1 Our Contributions

In this paper, we evaluate the security for the initialization phase of two members of the KNOT-AEAD family by some techniques of differential-linear cryptanalysis. Our attacks significantly improve the previous analysis results on them.

To relieve in certain extent the influence of the dependency between differential and linear parts in the differential-linear cryptanalysis, the targeted cipher E is usually divided as $E = E'_l \circ E_m \circ E_d$. Since the diffusion layers of KNOT permutations are very simple, for E_m of up to 8 rounds, the determined output difference at a single bit can be observed when the input difference has only one non-zero bit. To obtain a differential-linear distinguisher which covers as many rounds as possible and has higher correlation, our strategy is to restrict the input difference and output linear mask of E_m to be single-bit. The detailed procedure of searching such differential-linear distinguisher is presented in Section 4.1. In addition, to amplify the correlation, some condition equations are imposed to make the differential of E_d determined which might be key-dependent. Then the correlation of conditional differential-linear distinguisher on the targeted cipher E can be treated as the one on a degraded cipher $E'_l \circ E_m$. Further, the key-recovery attacks are launched based on the conditional differential-linear distinguishers. We apply these cryptanalytic techniques to KNOT-AEAD(128,256,64) and KNOT-AEAD(128,384,192), of which the former is the primary member recommended by the designers. As a result, for 15-round KNOT-AEAD(128,256,64) (out of 52 full rounds) our attack takes $2^{48.8}$ time complexity and $2^{47.5}$ blocks to recover the full 128-bit key. With regard to 17-round KNOT-AEAD(128,384,192) (out of 76 full rounds), we require time complexity of $2^{59.2}$ and data complexity of $2^{58.2}$ to launch a key-recovery attack.

Comparison of results Next, we compare our attacks with the known analysis against KNOT-AEAD(128,256,64) and KNOT-AEAD(128,384,192), which are summarized in Table 1.

Table 1. Summary of attacks on KNOT-AEAD

Cipher	Rounds	Type of attack	Time	Data	Ref.
†KNOT-AEAD v1	14	distinguisher	$O(2^{62.2})$	$O(2^{62.2})$	[31]
	12	key-recovery attack	$O(2^{60})$	$O(2^{60})$	[31]
	15	key-recovery attack	$2^{48.8}$	$2^{47.5}$	Sect. 4.2
†KNOT-AEAD v2	13	distinguisher	$O(2^{60.8})$	$O(2^{60.8})$	[31]
	12	key-recovery attack	$O(2^{60})$	$O(2^{60})$	[31]
	17	key-recovery attack	$2^{59.2}$	$2^{58.2}$	Sect. 4.3

† Here we adopt the notations used by designers in [31], i.e., KNOT-AEAD(128,256,64) and KNOT-AEAD(128,384,192) denoted by KNOT-AEAD v1 and KNOT-AEAD v2 respectively.

$ In our attacks, we can recover the full 128-bit secret key for 15-round KNOT-AEAD v1, and one bit of the secret key for 17-round KNOT-AEAD v2. In the previous analysis, only one bit of the secret key can be recovered for 12-round KNOT-AEAD v1 and v2.

In [31], the designers of KNOT give the security analysis of KNOT-AEAD and KNOT-Hash. Especially, for KNOT-AEAD(128,256,64), they presented a

14-round distinguishing attack of complexity $O(2^{62.2})$ by considering a truncated difference propagation and a 12-round key-recovery attack of complexity $O(2^{60})$ through a linear approximation involving one key bit. Similarly, for KNOT-AEAD(128,384,192), they showed a 13-round distinguishing attack of complexity $O(2^{60.8})$ by considering a truncated difference propagation and a 12-round key-recovery attack of complexity $O(2^{60})$ through a linear approximation involving one key bit. In a distinguishing attack, the algorithm (or distinguisher) allows to distinguish the ciphertext produced by the target cipher from a random permutation with high probability, but no information of the secret key can be obtained. Using the conditional differential-linear attacks, we obtain the key-recovery attack of 15-round KNOT-AEAD(128,256,64) which is three more rounds compared with their result. A key-recovery attack for 17-round KNOT-AEAD(128,384,192) is derived by our cryptanalytic techniques, and it is five rounds better than their analysis. For the detail of time and data complexities, please refer to Table 1.

1.2 Paper Organization

This paper is organized as follows. In Section 2, we describe the KONT-AEAD algorithms and review some basic notations and MILP-based automatic search for differential and linear trails. In Section 3, an overview of the classic differential-linear attack is showed, followed by some recent developments on it. We present the details of the conditional differential-linear attacks against KNOT-AEAD in Section 4. First, we show the procedure of our strategy for searching good differential-linear distinguishers in Section 4.1. Then we carry out our key-recovery attacks against 15-round KNOT-AEAD(128,256,64) and 17-round KNOT(128,384,192) in Section 4.2 and Section 4.3 respectively. Finally in Section 5, we give a brief summary of this paper.

2 Preliminaries

In this section we firstly describe the KNOT-AEAD algorithms and their underlying permutations. Then we provide an overview of some basic notations and MILP-based automatic search for differential and linear trails.

2.1 Description of KNOT-AEAD

The KNOT family is designed by Zhang et al., which includes bit-slice lightweight AEAD and hashing algorithms [30]. In this subsection, we give firstly a brief specification of KNOT-AEAD on which our differential-linear attacks are applied.

The underlying permutation of each KNOT member iteratively applies an SPN-based round transformation. There are three round transformations which are different only in the size b of block, $b = 256, 384, 512$. Each of the round transformations consists of three steps: $AddRoundConstans_b$, $SubColumn_b$ and $ShiftRow_b$. Let p_b denote a round transformation and $p_b = ShiftRow_b \circ SubCol-$

$umn_b \circ AddRoundConstans_b(S)$, where S is the b-bit state. As done in [30], a b-bit state is pictured as a $4 \times \frac{b}{4}$ rectangular array of bits. The first $\frac{b}{4}$ bits are arranged in 0-th row, denoted by a_0, the next $\frac{b}{4}$ bits are arranged in 1-st row, denoted by a_1, and so on. The j-th bit of i-th row is denoted by $a_{i,j}$ or $a_i[j]$. In the following, a cipher state is described in a two-dimensional way, as illustrated in Fig. 1.

$$\begin{bmatrix} a_{0,\frac{b}{4}-1} & \cdots & a_{0,1} & a_{0,0} \\ a_{1,\frac{b}{4}-1} & \cdots & a_{1,1} & a_{1,0} \\ a_{2,\frac{b}{4}-1} & \cdots & a_{2,1} & a_{2,0} \\ a_{3,\frac{b}{4}-1} & \cdots & a_{3,1} & a_{3,0} \end{bmatrix}$$

Fig. 1. A cipher state

The AddRoundConstant$_b$ transformation It consists of a simple bitwise XOR of a d-bit round constant generated by the corresponding d-bit LFSR to the first d bits of the intermediate state, with $d = 6, 7, 8$. Since we can ignore this transformation in our attacks, we omit the detailed description of round constants here.

The SubColumn$_b$ transformation It is composed of $\frac{b}{4}$ parallel applications of S-boxes to the 4 bits in the same column. The S-box used in KNOT is a 4-bit to 4-bit S-box S: $\{0,1\}^4 \rightarrow \{0,1\}^4$. The action of this S-box in hexadecimal notation is given in Table 2. Let x_0, x_1, x_2, x_3 and y_0, y_1, y_2, y_3 respectively denote the

Table 2. The S-box of KNOT

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

input and output of the KNOT S-box, where x_0 and y_0 are the least significant bits respectively. The algebraic normal form (ANF) of the S-box is the following:

$$\begin{aligned} y_0 &= x_0x_1 \oplus x_2 \oplus x_0x_2 \oplus x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_2x_3 \\ y_1 &= x_1 \oplus x_2 \oplus x_0x_3 \oplus x_2x_3 \oplus x_1x_2x_3 \\ y_2 &= 1 \oplus x_0 \oplus x_1 \oplus x_2 \oplus x_1x_2 \oplus x_3 \\ y_3 &= x_1 \oplus x_0x_1 \oplus x_2 \oplus x_3. \end{aligned}$$

The ShiftRow$_b$ transformation It makes up of a left rotation to each row over different offsets. The 0-th row is not rotated, i-th row is left rotated over c_i bit for $1 \leq i \leq 3$. The parameters (c_1, c_2, c_3) are $(1, 8, 25)$, $(1, 8, 55)$, $(1, 16, 25)$ for $b = 256, 384, 512$ respectively.

There are 4 members in KNOT-AEAD family whose modes are based on Duplex mode MonkeyDuplex [9]. Let KNOT-AEAD(k, b, r) denote a KNOT-AEAD member with k-bit key, b-bit state and r-bit rate. Note that the key length, the nonce length and the tag length are all equal to k bits for each member. In the following, we only concentrate on two members KNOT-AEAD(128,256,64) which is the primary member and KNOT-AEAD(128,384,192). Each member of KNOT-AEAD has 4 phases: initialization, processing associated data, encryption and finalization, which is illustrated in Fig. 2. The authenticated encryp-

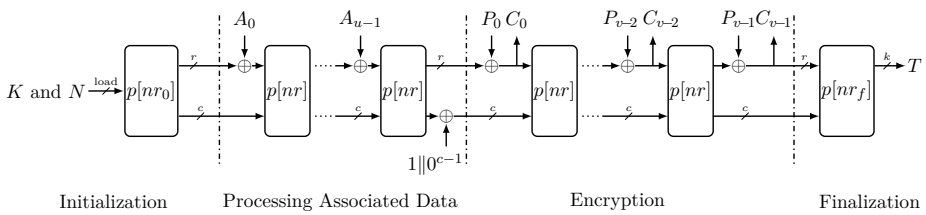

Fig. 2. The encryption of KNOT-AEAD

tion process is initialized by loading the key K and the nonce N, $S = K||N$ for KNOT-AEAD(128,256,64) and $S = (0^{128}||K||N) \oplus (1||0^{383})$ for KNOT-AEAD(128,384,192). Then the initial state is processed by $p[nr_0]$, i.e., nr_0 rounds of the round transformation, where $nr_0 = 52$ for KNOT-AEAD(128,256,64) and $nr_0 = 76$ for KNOT-AEAD(128,384,192). The associated data block A_i is XORed and then $p[nr]$ is applied to the intermediate state in sequence for $i = 0, \cdots, u-1$. The constant $1||0^{c-1}$ is XORed to the capacity part of state after the last $p[nr]$ in the process of associated data. Each plaintext block P_i is processed similarly to A_i for $i = 0, \cdots, v-1$, while the corresponding ciphertext block C_i is the output. In the finalization, $p[nr_f]$ is applied and the tag T is the output. The data, i.e., the blocks of processed plaintext and associated data, is limited to 2^{64} for both member by the designers of KNOT.

2.2 Notations

In this subsection, some basic notations are presented, followed by a fundamental lemma — Piling-up Lemma.

Let $\mathbb{F}_2 = \{0, 1\}$ be the finite field with two elements. The correlation of a binary random variable x is defined as $Cor(x) = \Pr[x = 0] - \Pr[x = 1] = 2\Pr[x = 0] - 1$. Similarly, the correlation of a Boolean function on an input with some distribution can be defined as following.

Definition 1. *The correlation of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as $Cor(f) = \Pr[f(X) = 0] - \Pr[f(X) = 1]$, where $X = (x_0, \cdots, x_{n-1})$ is a vector of binary random variables.*

Note that the probability that $f(X)$ equals zero can be expressed as

$$\Pr[f(X)=0] = \sum_{(a_0,\cdots,a_{n-1})\in\{X|f(X)=0\}} \Pr[X=(a_0,\cdots,a_{n-1})].$$

If $x_0,\cdots,x_{n-1}$ are independent with probability $D = \{\Pr[x_i = 0] = (1 + c_i)/2 | x_i \in X, 0 \le i \le n-1\}$, then

$$\Pr[f(X)=0] = \sum_{(a_0,\cdots,a_{n-1})\in\{X|f(X)=0\}} \prod_{i=0}^{n-1} \frac{1+(-1)^{a_i}c_i}{2},$$

where c_i is correlation of the binary variable x_i. Furthermore, if $x_0,\cdots,x_{n-1}$ are independently uniform distributed, then we can get $Cor(f) = 2^{-n}(\#\{X \in \mathbb{F}_2^n : f(X)=0\} - \#\{X \in \mathbb{F}_2^n : f(X)=1\})$, which is consistent with the definition in [14].

Lemma 1 (Piling-up Lemma [25]). *Let $x_0,\cdots,x_{n-1}$ be n independent binary random variables with probability $\Pr[x_i = 0] = p_i$. Then the following holds*

$$\Pr[x_0 \oplus \cdots \oplus x_{n-1} = 0] = \frac{1}{2} + 2^{n-1}\prod_{i=0}^{n-1}(p_i - \frac{1}{2}),$$

or alternatively, $Cor(x_0 \oplus \cdots \oplus x_{n-1}) = \prod_{i=0}^{n-1} Cor(x_i)$.

2.3 MILP-Based Automatic Search for Differential and Linear Trails

In this subsection, we give a brief description about the Mixed Integer Linear Programming (MILP)-based automatic search method for differential and linear trails and how we apply it to the KNOT permutations. For more details of this method, please refer to [29, 28].

In [26], Mouha et al. introduced the MILP model to count the number of active S-boxes for those word-oriented block ciphers. In ASIACRYPT 2014, Sun et al. [29] extended the framework to bit-oriented ciphers. Let x_i denote the difference variable for the i-th bit. That is, $x_i = 1$ if the difference at the i-th bit is active; otherwise, $x_i = 0$.

Suppose the two vectors $(x_0, x_1, \cdots, x_{\omega-1})$ and $(y_0, y_1, \cdots, y_{\nu-1})$ are the input and output bit differences of an $\omega \times \nu$ S-box S_t. Let the bit variable A_t denote the activity of this S-box. That is to say, $A_t = 1$ if S_t is active, and $A_t = 0$ otherwise. The following constraints can be used to ensure that non-zero input difference of the S-box must active it:

$$\begin{cases} A_t - x_k \ge 0, \ k = 0,\ldots,\omega-1, \\ -A_t + \sum_{j=0}^{\omega-1} x_j \ge 0. \end{cases} \tag{1}$$

Let a discrete point $(x_0,\cdots,x_{\omega-1},y_0,\cdots,y_{\nu-1}) \in \mathbb{R}^{\omega+\nu}$ denote the input-output differential pattern of an S-box, and then we can get a finite set of

discrete points Q, which includes all possible differential patterns of the S-box. By computing the H-Representation of the convex hull of Q, many linear inequalities are obtained, each of which can be used to remove some impossible differential patterns of the S-box. In [28], they presented the greedy algorithm to select a subset of the H-Representation of the convex hull with less inequalities, which can be used to exactly describe the differential patterns of the S-box.

The objective function can be to minimize the sum of all variables $\sum_t A_t$, which indicates the number of the active S-boxes appearing in the schematic description of the encryption. Using any MILP optimizer such as Gurobi, the model can be solved and good differential characteristics are returned.

For the S-box of the KNOT permutations, since there are 3 possible probabilities, i.e., 1, 2^{-2}, 2^{-3}, we append two extra bits (p, q) to encode the probability of the propagation. Therefore, a vector $(x_0, \cdots, x_3, y_0, \cdots, y_3, p, q) \in \mathbb{R}^{10}$ can describe a differential pattern with probability for the S-box. Then with the help of SageMath, 543 inequalities are derived through computing the H-Representation of the convex hull and the number of inequalities is reduced to 23 by the greedy algorithm in [28]. Since the linear layer of KNOT makes up of the left rotation, there is no need to introduce new inequalities. Besides, we can ignore the bitwise XOR of constants in the differential trail.

The objective function is minimization of the formula $\sum_t (p_t + 2q_t)$, which means the total probability of the differential trail through the encryption algorithm.

With regard to searching for linear trails, the modeling process is similiar with the aforementioned.

3 The Framework of Differential-Linear Attacks

We begin with an overview of the classical differential-linear attack, and then we recall some recent developments on it.

3.1 The Classic Differential-Linear Attack

Let E be an entire cipher that can be decomposed into two subciphers E_d and E_l such that $E = E_l \circ E_d$ where the differential and linear cryptanalyses are applied into E_d and E_l respectively. More precisely, assume that a differential $\Delta_{in} \xrightarrow{p} \Delta_m$ for E_d holds with probability $\Pr[E_d(X) \oplus E_d(X \oplus \Delta_{in}) = \Delta_m] = p$, and a linear approximation $\Gamma_m \xrightarrow{q} \Gamma_{out}$ for E_l holds with probability $\Pr[\Gamma_m \cdot X = \Gamma_{out} \cdot E_l(X)] = \frac{1}{2}(1 + q)$ (or with correlation q), where $\cdot$ denotes the inner product between two vectors. The differential-linear attack combines the above two distinguishers and the procedure of new distinguisher is presented in the following.

Procedure of the differential-linear distinguisher To distinguish E from a random permutation R, the adversary samples N plaintext pairs (P, P') such that $P \oplus P' = \Delta_{in}$ and checks whether the corresponding ciphertext pairs (C, C')

agree on the parity of output subset of bits in Γ_{out} one by one. The detailed procedure is presented as Algorithm 1. In the case that the differential in E_d

Algorithm 1 Procedure of the differential-linear distinguisher

Ensure: The cipher E or random permutation R.
1: Set a counter T to 0.
2: **for** N plaintext pairs $(P, P \oplus \Delta_{in})$ **do**
3: Increment T if $\Gamma_{out} \cdot (O(P) \oplus O(P \oplus \Delta_{in})) = 0$, where O is a encryption oracle E or R.
4: **end for**
5: **if** $\frac{2T}{N} - 1$ deviates enough from 0 **then**
6: The data is draw from the cipher E.
7: **else**
8: A random permutation R.
9: **end if**

fails, the equality $\Gamma_{out} \cdot E(X) = \Gamma_{out} \cdot E(X \oplus \Delta_{in})$ is usually assumed to hold in approximately half. Under the assumption that E_d and E_l are independent, the probability of differential-linear distinguisher can be natively estimated using Piling-up Lemma, $\Pr[\Gamma_{out} \cdot E(X) = \Gamma_{out} \cdot E(X \oplus \Delta_{in})] = p(\frac{1}{2}(1+q^2)) + (1-p)\frac{1}{2} = \frac{1}{2}(1+pq^2)$ (assuming that $\Gamma_m \cdot \Delta_m = 0$). Therefore, by preparing $N = O(p^{-2}q^{-4})$ chosen plaintext pairs $(P, P \oplus \Delta_{in})$, one can distinguish the cipher E from a random permutation using Algorithm 1 (refer to [13, 27] for the details about the data complexity and success probability).

In Theorem 2 of [27], Selçuk showed the analytical results of the success probability of a key-recovery attack in linear cryptanalysis. The main difference between the linear context and the differential-linear context is that the sign of the bias in the latter case is unaffected by any key bit (as all the affected key bits are used twice and thus canceled). In [13], Blondeau et al. adapted the framework of [27] and gave the success probability of a key-recovery attack in the differential-linear context as

$$P_S = \Phi(2\sqrt{N}|p_{dl} - \frac{1}{2}| - \Phi^{-1}(1 - 2^{-a})), \tag{2}$$

where Φ is the cumulative distribution function of the standard normal distribution, p_{dl} is the probability of differential-linear distinguisher, N is the number of chosen plaintext pairs and a is the advantage of the attack as defined in [27].

Exact analysis for the correlation of differential-linear distinguisher In [13], Blondeau et al. showed an exact expression of the correlation by *differential-linear hull*. With the following notations, the result is presented as Theorem 1.

$$\mathcal{E}_{\Delta_{in},\Gamma_{out}} = 2\Pr[\Gamma_{out} \cdot (E(X) \oplus E(X \oplus \Delta_{in})) = 0] - 1$$
$$\varepsilon_{\Delta_{in},\Gamma_m} = 2\Pr[\Gamma_m \cdot (E_d(X) \oplus E_d(X \oplus \Delta_{in})) = 0] - 1$$
$$c_{\Gamma_m,\Gamma_{out}} = 2\Pr[\Gamma_m \cdot X = \Gamma_{out} \cdot E_l(X)] - 1$$

Theorem 1 (Differential-Linear Hull [13]). *Assume that the part E_d and E_l of the block cipher $E = E_l \circ E_d$ are independent. Using the notation defined in the above, for all $\Delta_{in} \in \{0,1\}^n \setminus 0^n$ and $\Gamma_{out} \in \{0,1\}^n \setminus 0^n$, we have*

$$\mathcal{E}_{\Delta_{in},\Gamma_{out}} = \sum_{\Gamma_m \in \{0,1\}^n} \varepsilon_{\Delta_{in},\Gamma_m} c^2_{\Gamma_m,\Gamma_{out}}. \tag{3}$$

In Eq. (3) all the linear approximation trails are taken into account when estimating the correlation of the differential-linear approximation. Of course, it is usually hard to evaluate the correlation by the above expression. In practice, one mostly has to make the assumption of one *strong* linear approximation in E_l or *supporting subset* in the intermediate layer, and verify the results experimentally.

3.2 Recent improvements

In practice, the assumption of independence between E_d and E_l might arise a problem that results in wrong estimation for the correlation. Currently, the only way to get some evidence of this independence assumption is to experimentally compute the correlation of differential-linear approximation over a reduced number of rounds of the cipher. As done in recent works [3, 6, 20, 23], the subcipher E_l is further divided to two parts E'_l and E_m to obtain a more accurate estimation for the correlation of differential-linear distinguisher. That is the cipher is divided as $E = E'_l \circ E_m \circ E_d$ and the overall attack framework is depicted in Fig. 3. In [3], Bar-On et al. introduced a theoretical method called *Differential-*

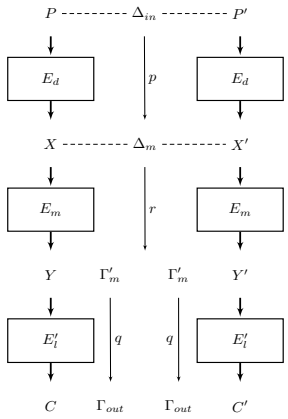

Fig. 3. The structure of differential-linear distinguisher

Linear Connectivity Table (DLCT) to cover the middle part E_m. Moreover, they showed that the DLCT could be efficiently constructed using the Fast Fourier Transform. With the DLCT, they further improved the differential-linear attacks on ICEPOLE and 8-round DES.

Subsequently, Beierle et al. [6] presented several improvements of the differential-linear attacks against ARX ciphers and successfully applied them to Chaskey and ChaCha. In their work, the correlation of middle part E_m is experimentally evaluated. Let r denote the correlation of the middle part E_m of the cipher, i.e., $r = Cor[\Gamma'_m \cdot (E_m(X) \oplus E_m(X \oplus \Delta_m))]$. Similarly, under the assumption of independence between the subciphers, the probability of differential-linear distinguisher can be simply estimated using Piling-up Lemma, $\Pr[\Gamma_{out} \cdot E(X) = \Gamma_{out} \cdot E(X \oplus \Delta_{in})] = \frac{1}{2}(1 + prq^2)$, and one can distinguish the cipher E from random permutation using $N = O(p^{-2}r^{-2}q^{-4})$ chosen plaintext pairs $(P, P \oplus \Delta_{in})$ using Algorithm 1. We pay attention to the part E_d where a differential $\Delta_{in} \xrightarrow{p} \Delta_m$ holds with probability $\Pr[E_d(X) \oplus E_d(X \oplus \Delta_{in}) = \Delta_m] = p$. Let $\mathbb{X}_d$ denote the set of all input values that define the right pairs for the differential, i.e., $\mathbb{X}_d = \{X \in \{0,1\}^n | E_d(X) \oplus E_d(X \oplus \Delta_{in}) = \Delta_m\}$. To amplify the correlation of differential-linear distinguisher, Beierle et al. [6] exploited the special structure of E_d called *fully or probabilistic independent parts* which could be rather likely observed in many ARX ciphers, such as ChaCha and Chaskey. With the help of *fully or probabilistic independent parts*, given one element $X \in \mathbb{X}_d$, one can generate many other elements in $\mathbb{X}_d$ for free or with some probability (almost 1), independently of the secret key. However, we can not use this method for a general permutation E_d due to that an arbitrary permutation might not have this special structure.

4 Differential-Linear Cryptanalysis of KNOT-AEAD

In this section, we present the details of the conditional differential-linear attacks against KNOT-AEAD. The KNOT's design document [30] shows, if the length of the associated data is zero, then no padding is applied and no associated data is processed. In our attacks, we omit the processing associated data phase, and our attack target is the initialization of KNOT-AEAD where nr'_0 rounds of the round transformation are applied to the initial state. In the following, we denote the targeted procedure by nr'_0-round KNOT-AEAD(k, b, r) which is shown in Fig. 4. Our attacks are performed in known plaintext attack scenario, i.e., P_0

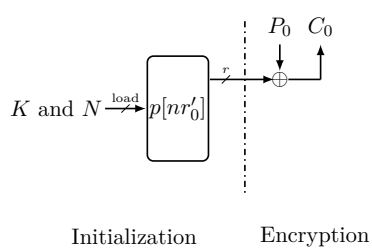

Fig. 4. The targeted procedure of KNOT-AEAD

and C_0 can be accessed. Only single-key model is taken into account in which the input difference is restricted on the nonce N which can be controlled by the adversary, and the output linear mask is restricted on the first 64 or 192 bits of a state, i.e., the rate part of a state.

4.1 Searching Differential-Linear Distinguishers

The diffusion layers of KNOT permutations are very simple and only make up of a left rotation to each row. For example, for 8-round E_m (from 3-rd to 10-th round), when the single-bit input difference is at 0-th bit of 2-nd row, the output difference at 55-th bit of 2-nd row is determined to be 1. To make the analysis cover as many rounds as possible and observe higher correlation, we only consider single-bit input difference and output linear mask of E_m. For obtaining a good differential-linear distinguisher, the procedure of our strategy is presented in the following:

STEP 1 Search the best difference trail given an arbitrary single-bit output difference on E_d by using the MILP-aided (differential trail) searching method. The best difference is denoted by $\Delta_{in} \rightarrow \Delta_m$ with the weight restriction $wt(\Delta_m) = 1$.

STEP 2 For all the single-bit input difference Δ_m, exhaustively search the best output linear mask whose weight is restricted to 1 on E_m by experiment-based correlation estimation method. The best differential-linear characteristic is denoted by $\Delta_m \rightarrow \Gamma'_m$ with the weight restriction $wt(\Delta_m) = wt(\Gamma'_m) = 1$.

STEP 3 Final, for all the single-bit input linear mask, exhaustively search the best linear trail on E'_l by using MILP-aided (linear trail) searching method. The best difference is denoted by $\Gamma'_m \rightarrow \Gamma_{out}$ with the weight restriction $wt(\Gamma'_m) = 1$.

Then many differential-linear distinguishers are generated by combining results in the above three steps and the best one is chosen to launch our attacks.

4.2 Attack on 15-round KNOT-AEAD(128,256,64)

In this subsection, we firstly show how to find the differential-linear distinguisher of 15-round KNOT-AEAD(128,256,64). Then a key-recovery attack is presented based on this distinguisher.

Differential-Linear Distinguisher Since the (differential-linear) characteristics of KNOT are rotation-invariant within a 64-bit word, all the single-bit differences (or linear masks) can be classified into four cases. For each case, we search the best result separately.

We searched all the differential trails of E_d up to 8 round in which the weight of output difference is restricted 1. But, there are three possible effective differential trails which can be used in our attacks, two 1-round trails with probability

2^{-3} and output difference at words of 0/2-th row, one 2-round trail with probability 2^{-6} and output difference at word of 2-nd row, for details please refer to Appendix A.1. To increase the number of rounds attacked as much as possible, we choose the 2-round trail in the following analysis, which is shown in Table 3. In the presentation of trails, the row ordering is from right to left, then from top-down, i.e., 0, 1, 2 and 3-th row respectively.

Table 3. A 2-round differential trail with probability 2^{-6}

Round	Difference
Input	0x0100000000000000 0x0100000000000000
	0x0000000000000000 0x0000000000000000
1st round	0x0000000000000000 0x0100000000000000
	0x0000000000000000 0x0000000000000000
2nd round	0x0000000000000000 0x0000000000000000
	0x0000000000000000 0x0000000000000001

To amplify the correlation, some condition equations are imposed to make the differential of E_d determined. Similar techniques are appeared in conditional differential cryptanalysis [7, 21]. Then the correlation of conditional differential-linear distinguisher on the targeted cipher $E = E'_l \circ E_m \circ E_d$ can be treated as the one on a degraded cipher $E'_l \circ E_m$. With the symbolic computation in SageMath, the following 6 condition equations are imposed to make the differential trail in Table 3 determined, i.e., 3 bits of 1-st round $\Delta a_0^{(1)}[56] = n_0[56]k_1[56] \oplus n_0[56] \oplus n_1[56]k_1[56] \oplus n_1[56] \oplus k_0[56] \oplus 1 = 1$, $\Delta a_1^{(1)}[57] = k_0[56]k_1[56] \oplus k_1[56] \oplus 1 = 0$, $\Delta a_3^{(1)}[17] = n_0[56] \oplus n_1[56] = 0$, and 3 bits of 2-nd round $\Delta a_0^{(2)}[56] = a_1^{(1)}[56]a_3^{(1)}[56] \oplus a_1^{(1)}[56] \oplus a_2^{(1)}[56] = 0$, $\Delta a_1^{(2)}[57] = a_3^{(1)}[56] = 0$, $\Delta a_3^{(2)}[17] = a_1^{(1)}[56] = 0$, where $a_i^{(r)}[j]$ ($\Delta a_i^{(r)}[j]$) denotes the (difference) expression of the j-th bit of i-th 64-bit word of r-th round, $k_i[j]$ and $n_i[j]$ are the j-th bit of i-th 64-bit word of the secret K and nonce N respectively. Generally, we impose the 6 condition equations to make the corresponding differential trails determined, i.e., $\Delta a_0^{(1)}[i] = 1$, $\Delta a_1^{(1)}[(i+1) \bmod 64] = 0$, $\Delta a_3^{(1)}[(i+25) \bmod 64] = 0$, and $\Delta a_0^{(2)}[i] = 0$, $\Delta a_1^{(2)}[(i+1) \bmod 64] = 0$, $\Delta a_3^{(2)}[(i+25) \bmod 64] = 0$, for $0 \le i \le 63$, which are simplified as the below:

$$\begin{gathered}
k_0[i] = 0, \\
k_1[i] = 1, \\
n_0[i] \oplus n_1[i] = 0, \\
n_0[i-1]k_1[i-1] \oplus (n_1[i-1] \oplus 1)k_0[i-1]k_1[i-1] \oplus n_1[i-1] \oplus k_0[i-1] = 0, \\
n_0[i-8] = (n_1[i-8] \oplus 1)k_0[i-8] \oplus n_1[i-8] \oplus k_1[i-8] \oplus 1, \\
(n_0[i-25] \oplus 1)n_1[i-25] = k_0[i-25] \oplus k_1[i-25]
\end{gathered} \tag{4}$$

where indices should be modulo 64 which denote the order of bits in a word. In the following, we will set the above condition equations to be satisfied and perform sampling experiments of correlation estimation of E_m.

For the subcipher E'_l, we searched all the linear trails up to 5 round in which the weight of input linear mask is restricted 1. But there is only one possible effective linear trail which can be used in our attacks, a 1-round trail with correlation 2^{-2} and input linear mask at word of 2-nd row, which is shown in Table 4.

Table 4. A 1-round linear trail with correlation 2^{-2}

Round	Linear mask
Input	0x0000000000000000 0x0000000000000000
	0x0000000000000000 0x0000000000000001
1st round	0x0000000000000000 0x0000000000000001
	0x0000000000000000 0x0000000000000000

Targeting the result of 15-round KNOT-AEAD(128,256,64), we exhaustively searched all the 64 single-bit output linear masks at word of 2-nd row for E_m of 12 rounds under the condition that the above difference equations in Eq. (4) are satisfied. We use at most 2^{35} random nonces for each of the 2^4 random keys, so we can only measure a correlation of about $|Cor| > c \cdot 2^{-17.5} = 2^{-14}$ (where $c \approx \sqrt{128}$ for a reasonable estimation error). For the input difference showed in Table 3 (i.e., $i = 56$), the best output linear mask is at 27-th bit of 2-nd row with correlation $-2^{-11.9}$. Furthermore, once one or more condition equations of Eq. (4) are not satisfied, we can not detect any correlation. As a result, we obtain a differential-linear distinguisher of correlation $-2^{-15.9}$ for 15-round KNOT(128,256,64) by splicing the above 1-round linear trail.

Key-Recovery Attack As defined in [21], conditions that control the difference propagations can be classified into three types in a chosen plaintext/initial value (IV) attack scenario, Type 0 conditions only involving bits of IV, Type 1 conditions only involving bits of IV and the secret key, Type 2 conditions only involving bits of the secret key. Note that the adversary can impose the condition equations of Type 0 for free in a chosen plaintext/IV attack scenario. Since condition equations might be key-dependent (only considering Type 1), we need guess the values of expressions that consist of bits of the secret key and choose the value of corresponding IV bits according to the condition equations. For the case of Type 2, the differential-linear cryptanalysis might degrade into a weak key-recovery attack. Assume that there are $l = l_1 + l_2$ independent expressions of key bits, l_1 conditions of Type 1 and l_2 conditions of Type 2. The general procedure of the conditional differential-linear attack is summarized as follows.

Procedure of the conditional differential-linear attack For each guess of expressions of key bits in Type 1, like in Algorithm 1, the adversary samples N pairs of initial state (X, X') such that $X \oplus X' = \Delta_{in}$ and counts how many times the corresponding ciphertext pairs agree on the parity of output subset of bits in Γ_{out}. To make the differential propagation in E_d determined, some bits of IV called conditional IV bits are set according to the guessed values of expressions of key bits and the condition equations (containing Type 0). Besides, the sampled pairs of initial state should be generated by flipping non-conditional IV bits. In order to obtain an effective distinguisher, there must be enough non-conditional IV bits to generate initial states to observe the correlation of differential-linear distinguisher. In Algorithm 2, the details of this procedure are presented and the complexity of Algorithm 2 is $2^{l_1+1}N$. Note that we ignore the conditions of Type 2 in Algorithm 2, additional 2^{l_2} executions of Algorithm 2 are needed if there are enough other differential-linear distinguishers.

Algorithm 2 Procedure of the conditional differential-linear attack

Ensure: Set of candidates for some expressions of key bits of the cipher E.

```
for All 2^{l_1} possible values of expressions of key bits imposed in E_d do
    Set the conditional IV bits according to the condition equations.
    Prepare N pairs of initial states (X, X ⊕ Δ_in) by varying non-conditional IV
    bits.
    Set a counter T to 0.
    for N initial states (X, X ⊕ Δ_in) do
        Increment T if Γ_out · (E(X) ⊕ E(X ⊕ Δ_in)) = 0.
    end for
    if 2T/N − 1 deviates enough from 0 then
        Keep the current value of expressions of key bits in C_K as a candidate.
    end if
end for
Reduce the system of expressions of key bits in C_K and return the solutions.
```

For 15-round KNOT-AEAD(128,256,64), we launch the key-recovery attack by Algorithm 2. Among Eq. (4), 1 of 6 belongs to Type 0 conditions, 3 of 6 belong to Type 1 conditions and 2 of 6 belong to Type 2 conditions. First, we impose the Type 0 condition $n_0[i] \oplus n_1[i] = 0$ for free. And the value of $n_0[i-25]$ is set to zero (or $n_1[i-25] = 1$) to prevent the corresponding condition equation to degrade into a Type 2 condition. In Algorithm 2, the conditional IV bits are chosen as $CIV = \{n_0[i], n_0[i-25]\} \cup \{n_1[i-1], n_0[i-8], n_1[i-25]\}$ and $l_1 = 3$. Once the values of the flipped bits $FIV = \{n_0[i-1] = 0, n_1[i-8]\}$ are given, we can distinguish $k_0[i] = 0, k_1[i] = 1$ from the other cases using Algorithm 2 with the time and data complexities $2^{l_1+1}N = 2^{3+1+37.3} = 2^{41.3}$ and the success probability is almost 1 according to the formula in [13], Eq. (2). On average, we will succeed by repeatedly executing Algorithm 2 for $2^{l_2} = 4$ times with the index i varying which identifies the differential-linear distinguisher. In the case $k_0[i] = 0, k_1[i] = 1$, three or four extra expressions of key bits can be

recovered, i.e., $k_0[i-1] = k_1[i-1] = 1$ or $k_0[i-1] = c_1$ (when $k_0[i-1]k_1[i-1] = 0$), $(v \oplus 1)k_0[i-8] \oplus v \oplus k_1[i-8] \oplus 1 = c_2, k_0[i-25] \oplus k_1[i-25] = c_3$, where $n_0[i-1] = 0$ and v is the value of $n_1[i-8]$. Note that when $k_0[i-1] = k_1[i-1] = 1$, the corresponding condition equation holds no matter what the value of $n_1[i-1]$ is. With the additional complexity of $2^{1+1+37.3} = 2^{39.3}$, one or two equations of key bits can be derived by flipping the value of FIV bit by bit, i.e., $k_1[i-1] \oplus k_0[i-1] = c_1'$ (when $k_0[i-1]k_1[i-1] = 0$), $(\overline{v} \oplus 1)k_0[i-8] \oplus \overline{v} \oplus k_1[i-8] \oplus 1 = c_2'$. Therefore, we can totally obtain 7 independently linear equations of key bits in the case $k_0[i] = 0, k_1[i] = 1$.

For i $(0 \leq i \leq 63)$, we can perform the above process 64 times and obtain some equations of key bits. On average, there are 16 indices i's such that $k_0[i] = 0, k_1[i] = 1$, and $16 \times 7 = 112$ equations of key bits can be derived in total. We have checked that the above linear system has on average 80 independently linear equations by thousands of experiments. In conclusion, we can recover the 128-bit secret key for 15-round KNOT(128,256,64) with the expected time complexity $(64 \times 2^{41.3} + 16 \times 2 \times 2^{39.3}) + 2^{48} = 2^{48.8}$ and data complexity of $64 \times 2^{41.3} + 16 \times 2 \times 2^{39.3} = 2^{47.5}$.

4.3 Attack on 17-round KNOT-AEAD(128,384,192)

Similar to the attack of the primary AEAD member, we present the key-recovery attack against KNOT-AEAD(128,384,192) in the following.

Differential-Linear Distinguisher We searched all the differential trails with single-bit output difference for E_d up to 9 round. But there are only two possible effective differential trails, two 1-round trails with probability 2^{-3} and output difference at words of 0/2-th row respectively, for details please refer to Appendix A.2. Note that the constants of initial state are treated as variables in our MILP-aided differential trail searching method. In the following attack, we choose the 1-round differential trail which is shown in Table 5.

Table 5. A 1-round differential trail with probability 2^{-3}

Round	Difference
Input	0x000000000000000000000000 0x010000000000000000000000 0x000000000000000000000000 0x000000000000000000000000
1st round	0x000000000000000000000000 0x000000000000000000000000 0x000000000000000000000000 0x000000000000000000000001

Taking the constants of initial state into account, we recompute the propagation of the above 1-round differential trail by SageMath. When the input difference is at i-th bit of 0-row $(0 \leq i \leq 94)$, a single-bit output difference at $((i+8) \bmod 96)$-th bit of 2-nd row can be observed with probability 2^{-1} or 2^{-2}.

The difference propagation will be certain when the below condition equations of difference are satisfied.

$$\begin{array}{lll} n_{i+96} = 0, & k_{i+64} = 0, & 0 \le i \le 31, \\ k_{i-32} = 0, & k_{i+96} = 0, & 32 \le i \le 63, \\ k_{i-32} = 0, & & 64 \le i \le 94. \end{array} \tag{5}$$

For the subcipher E_m of 13 rounds, all the single-bit output linear masks are exhaustively searched when the input difference is at 0-th bit of 2-nd row. As a result, we detected high correlation at 14-th bit of 0-th row with $2^{-12.6}$.

For the subcipher E'_l, we searched all the linear trails up to 7 round in which the weight of input linear mask is restricted 1. Consequently, some rounds can be expanded forward when the single-bit input linear mask is at 0-th row or 2-nd row. From them, we choose three effective linear trails which can be used in our attacks, trails for 1/2/3 round(s) with correlation $2^{-2}/2^{-5}/2^{-7}$ respectively and input linear masks at words of 0-th row for them, for details please refer to Appendix B. By splicing the 3-round linear trail which is shown in Table 6, we obtain a differential-linear trail of correlation $2^{-26.6}$ for 17-round KNOT(128,384,192).

Table 6. A 3-round linear trail with correlation 2^{-7}

Round	Linear mask	
Input	0x000000000000000000000000	0x000000000000000000000001
	0x000000000000000000000000	0x000000000000000000000000
1st round	0x000000000000000000000000	0x000000000000000000000001
	0x000000000000000000000000	0x000000000000000000000100
2nd round	0x000000000000000000000000	0x000000000000000000000101
	0x000000000000000000000000	0x000000000000000000000100
3rd round	0x000000000000000000000202	0x000000000000000000000100
	0x000000000000000000000000	0x000000000000000000000000

Key-Recovery Attack In the case $k_{i-32} = 0$ ($64 \le i \le 94$), the high correlation can be observed with time and data complexities of $2^{1+57.2} = 2^{58.2}$ with success probability of almost 1 by Algorithm 2 where no key expression needs to be guessed. We will succeed on average by repeatedly running Algorithm 2 for 2 times with different indices. In conclusion, with the time complexity of $2^{1+58.2} = 2^{59.2}$ and data complexity of $2^{58.2}$, we can on average distinguish 17-round KNOT(128,384,192) from random permutation and recover one bit of the secret key. Similar analysis can be obtained for any $i \in [0, \cdots, 63]$.

5 Conclusion

In this paper, by some techniques of differential-linear cryptanalysis, we focus on the security for the initialization phase of two members of the KNOT-AEAD

family. Based on an observation that the diffusion layers of KNOT permutations are very simple, our strategy is to restrict the input difference and output linear mask of E_m to be single-bit such that we can obtain a differential-linear distinguisher which covers as many rounds as possible and has higher correlation. In addition, to amplify the correlation, some condition equations are imposed to make the differential of E_d determined which might be key-dependent. Then we can carry the key-recovery attacks based on the conditional differential-linear distinguishers. We apply these cryptanalytic techniques to KNOT-AEAD(128,256,64) and KNOT-AEAD(128,384,192). As a result, our attacks significantly improve the previous analysis results on them.

A Appendix A

In Appendix A, we present only the differential trails which can be effectively used in our differential-linear attacks.

A.1 KNOT-AEAD(128,256,64)

The other two 1-round differential trails of KNOT-AEAD(128,256,64) is showed in the following.

Table 7. A 1-round differential trail with probability 2^{-3}

Round	Difference
Input	0x0000000000000001 0x0000000000000001 0x0000000000000000 0x0000000000000000
1st round	0x0000000000000000 0x0000000000000001 0x0000000000000000 0x0000000000000000

Table 8. A 1-round differential trail with probability 2^{-3}

Round	Difference
Input	0x0000000000000000 0x0100000000000000 0x0000000000000000 0x0000000000000000
1st round	0x0000000000000000 0x0000000000000000 0x0000000000000000 0x0000000000000001

A.2 KNOT-AEAD(128,384,192)

The other one 1-round differential trail of KNOT-AEAD(128,384,192) is showed in the following.

Table 9. A 1-round differential trail with probability 2^{-3}

Round	Difference
Input	0x000000000000000000000001 0x000000000000000000000001 0x000000000000000000000000 0x000000000000000000000000
1st round	0x000000000000000000000000 0x000000000000000000000001 0x000000000000000000000000 0x000000000000000000000000

B Appendix B

In Appendix B, we present the other two linear trails of KNOT(128,384,192) which can be effectively used in our differential-linear attacks.

Table 10. A 1-round linear trail with correlation 2^{-2}

Round	Linear mask
Input	0x000000000000000000000000 0x000000000000000000000001 0x000000000000000000000000 0x000000000000000000000000
1st round	0x000000000000000000000002 0x000000000000000000000001 0x000000000000000000000000 0x000000000000000000000000

Table 11. A 2-round linear trail with correlation 2^{-5}

Round	Linear mask
Input	0x000000000000000000000000 0x000000000000000000000001 0x000000000000000000000000 0x000000000000000000000000
1st round	0x000000000000000000000000 0x000000000000000000000001 0x000000000000000000000000 0x000000000000000000000100
2nd round	0x000000000000000000000002 0x000000000000000000000101 0x000000000000000000000000 0x000000000000000000000000

References

1. Mark Aagaard, Riham AlTawy, Guang Gong, Kalikinkar Mandal, Raghvendra Rohit, and Nusa Zidaric. Wage: An authenticated cipher submission to the nist lwc competition. *https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/wage-spec-round2.pdf.*
2. Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New features of latin dances: Analysis of salsa, chacha, and rumba. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International*

Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers, volume 5086 of *Lecture Notes in Computer Science*, pages 470–488. Springer, 2008.

3. Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. DLCT: A new tool for differential-linear cryptanalysis. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 313–342. Springer, 2019.
4. Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Schwaemm and esch: Lightweight authenticated encryption and hashing using the sparkle permutation family. *https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/sparkle-spec-round2.pdf*.
5. Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. Skinny-aead and skinny-hash v1.1. *https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/SKINNY-spec-round2.pdf*.
6. Christof Beierle, Gregor Leander, and Yosuke Todo. Improved differential-linear attacks with applications to ARX ciphers. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.
7. Ishai Ben-Aroya and Eli Biham. Differential cryptanalysis of lucifer. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 1993.
8. Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.
9. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Permutation-based encryption, authentication and authenticated encryption. *Directions in Authenticated Ciphers*, pages 159–170, 2012.
10. Eli Biham, Orr Dunkelman, and Nathan Keller. Enhancing differential-linear cryptanalysis. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 254–266. Springer, 2002.
11. Eli Biham, Orr Dunkelman, and Nathan Keller. Differential-linear cryptanalysis of serpent. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 9–21. Springer, 2003.

12. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.
13. Céline Blondeau, Gregor Leander, and Kaisa Nyberg. Differential-linear cryptanalysis revisited. *J. Cryptol.*, 30(3):859–888, 2017.
14. Anne Canteaut. Lecture notes on cryptographic boolean functions. *https://www.rocq.inria.fr/secret/Anne.Canteaut/poly.pdf*.
15. Anne Canteaut, Sébastien Duval, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Thomas Pornin, and André Schrottenloher. Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. *https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/saturnin-spec-round2.pdf*.
16. Arka Rai Choudhuri and Subhamoy Maitra. Significantly improved multi-bit differentials for reduced round salsa and chacha. *IACR Trans. Symmetric Cryptol.*, 2016(2):261–287, 2016.
17. Murilo Coutinho and T. C. Souza Neto. New multi-bit differentials to improve attacks against chacha. *IACR Cryptol. ePrint Arch.*, 2020:350, 2020.
18. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of ascon. In Kaisa Nyberg, editor, *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, volume 9048 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2015.
19. Orr Dunkelman, Sebastiaan Indesteege, and Nathan Keller. A differential-linear attack on 12-round serpent. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings*, volume 5365 of *Lecture Notes in Computer Science*, pages 308–321. Springer, 2008.
20. Antonio Flórez-Gutiérrez, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, André Schrottenloher, and Ferdinand Sibleyras. New results on gimli: Full-permutation distinguishers and improved collisions. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 33–63. Springer, 2020.
21. Simon Knellwolf, Willi Meier, and María Naya-Plasencia. Conditional differential cryptanalysis of nlfsr-based cryptosystems. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2010.
22. Susan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 1994.
23. Gaëtan Leurent. Improved differential-linear cryptanalysis of 7-round chaskey with partitioning. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in*

Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I, volume 9665 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2016.

24. Jiqiang Lu. A methodology for differential-linear cryptanalysis and its applications. *Des. Codes Cryptogr.*, 77(1):11–48, 2015.
25. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.
26. 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, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.
27. Ali Aydin Selçuk. On probability of success in linear and differential cryptanalysis. *J. Cryptol.*, 21(1):131–147, 2008.
28. 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. Cryptology ePrint Archive, Report 2014/747, 2014. https://ia.cr/2014/747.
29. 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, *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 158–178. Springer, 2014.
30. Wentao Zhang, Tianyou Ding, Bohan Yang, Zhenzhen Bao, Zejun Xiang, Fulei Ji, and Xuefeng Zhao. Knot: Algorithm specifications and supporting document. *https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/knot-spec-round.pdf*.
31. Wentao Zhang, Tianyou Ding, Chunning Zhou, and Fulei Ji. Security analysis of knot-aead and knot-hash. *https://csrc.nist.gov/CSRC/media/Events/lightweight-cryptography-workshop-2020/documents/papers/security-analysis-of-KNOT-lwc2020.pdf*.