DEEPAND: In-Depth Modeling of Correlated AND Gates for NLFSR-based Lightweight Block Ciphers

Amit Jana[1], Mostafizar Rahman[2], and Dhiman Saha[3]

[1] Indian Statistical Institute, Kolkata
janaamit001@gmail.com
[2] University of Hyogo
mrahman454@gmail.com
[3] de.ci.phe.red Lab, Department of Electrical Engineering and Computer Science, Indian Institute of Technology Bhilai
dhiman@iitbhilai.ac.in

Abstract. Automated cryptanalysis has taken center stage in the arena of cryptanalysis since the pioneering work by Mouha *et al.* which showcased the power of Mixed Integer Linear Programming (MILP) in solving cryptanalysis problems that otherwise, required significant effort. Since the inception, research in this area has moved in primarily two directions. One is to model more and more classical cryptanalysis tools as optimization problems to leverage the ease provided by state-of-the-art solvers. The other direction is to improve existing models to make them more efficient and/or accurate. The current work is an attempt to contribute to the latter. In this work, a general model referred to as DEEPAND has been devised to capture the correlation between AND gates in NLFSR-based lightweight block ciphers. DEEPAND builds upon and generalizes the idea of joint propagation of differences through AND gates captured using refined MILP modeling of TinyJAMBU by Saha *et al.* in FSE 2020. The proposed model has been applied to TinyJAMBU and KATAN and can detect correlations that were missed by earlier models. This leads to more accurate differential bounds for both the ciphers.

In particular, a 384-round (*full-round* as per earlier specification) Type-IV trail is found for TinyJAMBU with 14-active AND gates using the new model, while the refined model reported this figure to be 19. This also reaffirms the decision of the designers to increase the number of rounds from 384 to 640. Moreover, the model succeeds in searching a *full round* Type-IV trail of TinyJAMBU keyed permutation $\mathcal{P}_{1024}$ with probability $2^{-108}(\gg 2^{-128})$. This reveals the non-random properties of $\mathcal{P}_{1024}$ thereby showing it to be *non-ideal*. Hence it cannot be expected to provide the same security levels as robust block ciphers. Further, the provable security of TinyJAMBU AEAD scheme should be carefully revisited.

Similarly, for KATAN32, DEEPAND modeling improves the 42-round trail with 2^{-11} probability to 2^{-7}. Also, for KATAN48 and KATAN64, this model respectively improves the designer's claimed 43-round and 37-round trail probabilities. Moreover, in the related-key setting, the DEEPAND model is able to make a better 140-round boomerang distinguisher (for

both the data and time complexity) in comparison to the previous boomerang attack by Isobe *et al.* in ACISP 2013. In summary, DEEPAND seems to capture the underlying correlation better when multiple AND gates are at play and can be adapted to other classes of ciphers as well.

Keywords: MILP · KATAN · TinyJAMBU · Symmetric-Key Cryptanalysis

1 Introduction

One of the fundamental decisions in any iterative block cipher design, once we have a *good* round function, is the number of rounds. This decision is a trade-off between security and efficiency and plays an even more critical part in the context of Lightweight Cryptography which is referred to as crypto tailored for resource contained environments. A typical way to decide this is to take into account the penetration of best attack available and then adding some more rounds as the so-called *security-margin*. Traditionally, designers try to prove how many rounds are sufficient to resist a certain kind of attack. This in general is a rigorous task and primarily limited to a specific construction. For instance resistance against differential cryptanalysis [3] relies on the number of active sboxes in the best available differential trail. It has been a long standing question if these seeming critical task of cryptanalysis could be automated or aided in some generic way. Though there have been initial attempts in this direction but the first major breakthrough in this direction is attributed to Mouha *et al.* [7] who was one of the first to demonstrate how the cryptanalytic problem of determining minimum number of active sboxes could be modeled as an optimization problem which could in turn be solved by automated solvers. In particular, the authors showcased how Mixed Integer Linear Programming (MILP) can be leveraged as an ingenious cryptanalysis aid. This seminal work spawned an entirely new line of research where the goal is at one hand to increase the breadth of the strategy with new modelings (applications to linear, division, impossible differential cryptanalysis). On the other hand the idea is to improve upon the existing models to capture the underlying crypto property as closely as possible. The current work aims to add to state-of-art with *better* MILP modeling.

Interestingly, researchers have shown that there are mechanisms to precisely model valid transition for many crypto properties [13,12,11]. However, the catch is that this results in models becoming over-constrained thereby infeasible to be solved in reasonable time. On the other end of the spectrum is an over simplified model which might lead to invalid transitions. There is a rich body of work that tries to reach a middle ground by what can perhaps be referred to as *balanced* modeling [4,2]. In FSE 2020, Saha *et al.* made an interesting observation in this line of balanced modeling for the NIST-LWC [8] competition finalist TinyJAMBU [1]. The authors pointed out that correlation between multiple AND gates could lead to them becoming dependent leading to joint propagation of differential characteristics. Our research pushes the boundaries to reveal that further refinement is possible and a generalized model can be devised to extend

the findings to a class of Non-Linear Feedback Shift Register (NLFSR) based lightweight block ciphers with specific results on KATAN [5] and TinyJAMBU [1].

1.1 Our Contributions

Generalizes AND Modeling Framework The current work proposes a generalized model to capture first-order correlation in single as well as *multiple* AND gates. This is a direct improvement over the recent work [9] by Saha *et al.* where a new MILP model was developed leveraging AND gate correlations. To be precise, the analysis by Saha *et al.* exploits two subsequent AND computations with a common input position (for e.g the middle bit position b out of three inputs a, b, c to the subsequent ANDs). The present work provides further insight into this interesting correlation by extending it to *multiple* AND gates. The findings show significant impact on the actual probabilities of the differential trails. More specifically, the common input position in the two subsequent ANDs will be revealed when a particular difference pattern. For instance if $(\Delta a, \Delta b, \Delta c) = (1, 0, 1)$) one has to pay a probability for only the first AND whereas the second AND will pass freely. We further re-investigate this case and observe that due to the difference $(\Delta a, \Delta b) = (1, 0)$, the output difference (Δz_1) of the first AND directly reveals the bit b, i.e., $\Delta z_1 = b$. Once Δz_1 is fixed, passage through the second AND is for free. From another perspective, for an AND gate with two inputs a, b, if we know the bit value of a, then for a given difference pattern $(\Delta a, \Delta b) = (0, 1)$, the output difference $\Delta z = a$ will become deterministic. We would like to emphasize that the distribution of differences in AND gates under conditionally known inputs might be well-known. However, in the current work we revisit this in the light of correlations that develop and can hence be exploited in MILP modelings.

Improved Bounds for TinyJAMBU and KATAN Our research constitutes a comprehensive study of all correlation that develop (and were perhaps *missed* in earlier attempts) with or without conditionally known inputs. These correlations when incorporated in MILP models lead to the best trails known on NLFSR-based ciphers TinyJAMBU and KATAN which in-turn can be exploited to mount distinguishing and forgery attacks. It is worth noting that correlated AND, though not a new observation, earlier results were only restricted to the single AND gates. For NLFSR-based ciphers employing multiple ANDs, the current work adds newer cases there also bettering the size of the differential trail clusters generated. All finding are consolidated a new generalized *refined* model. Finally, we apply this new model in the keyed-permutation of TinyJAMBU AE and to all the KATAN -variants and show that our model captures all possible correlations between ANDs and provides a better optimal differential trails in comparison to previous models.

Notion of Conditionally Free Rounds The work builds upon two fundamental observations which are looked at from an information theoretic way in terms of conditionally known inputs and how much reduction it leads to in terms

of the overall entropy of the values and differences that related to one/multiple AND gates. The Observations 2 and 3 help identify the underlying principle behind the chained (a term introduced in [9]) AND gates by introducing the notion of what we refer as *conditionally free* rounds. The primary motivation is to redefine the notion of correlated AND operations (Lemma 1) using these observations referred above. The proof idea stems form the fact that for some specific differential inputs and output of the AND gate, we gain some extra information about the actual bits of the internal state thereby reducing the entropy. The observations are further exploited to develop a generalized MILP model for differential cryptanalysis, referred to as DEEPAND, which has the potential to captures all possible correlations between multiple ANDs in NLFSR-based (AND based) block ciphers. Consider a NLFSR-based block cipher with h AND gates. Suppose, the AND gates compute $(a_1^1 \cdot b)$, $(b \cdot a_1^2)$, $\cdots$, $(a_h^1 \cdot b)$, $(b \cdot a_h^2)$ across some rounds. Using Observation 2 and 3, we have proved that these $2h$ AND operations are correlated. Essentially, if $b = 0$ and out of these $2h$ AND computations, m AND gates are active, then due to the correlated nature between the AND gates the output of these AND computations can be fixed with probability 2^{-1} instead of 2^{-m}. The proposed DEEPAND model employs the following properties to find differential trails

1. Captures all possible correlations between several AND computations
2. It also exploits Observation 2 and 3 independently to gain advantage to penetrate some extra rounds freely in the differential trails.

As an immediate application, the DEEPAND model is applied on KATAN block cipher to find differential trails which are better than existing ones. We have explicitly shown trails where the dependency between several AND computations are captured. The model is also able to improve the related-key boomerang attacks on KATAN. To show the versatility of the model, it is also employed on keyed permutation of TinyJAMBU. The model is able to improve the differential trails of TinyJAMBU in comparison to the ones retrieved by the refined model due to fact of employing Observation 2 and 3. Finally, a forgery attack on TinyJAMBU is mounted with a probability of $2^{-67.88}$.

1.2 Outline of the Paper

This paper is organized as follows. First, the description of TinyJAMBU and KATAN are given in Section 2. In Section 3, DEEPAND model is introduced and we revisit the correlation between two subsequent AND gates in the previous refined MILP model and further, we have shown some observations regarding the non-uniform behaviour of the output distribution of the AND gate. Based on our observations, we propose a framework to capture the dependency between multiple AND gates in Section 4. Section 5 deduces a new MILP model for a class of NLFSRs with a single/multiple AND in the feedback function to efficiently search for differential trails. Our results on differential cryptanalysis for the keyed permution of TinyJAMBU and KATAN family of ciphers is described in Section 6

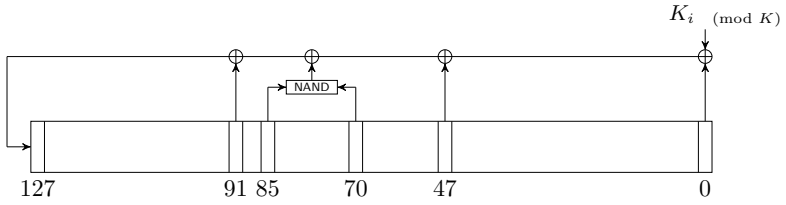

Fig. 1: The Permutation P^{k_i}

and Section 7, respectively. Finally, the concluding remarks are furnished in Section 8.

2 Preliminaries

In this section, first of all, the notations used in the paper is described. Then a brief description about TinyJAMBU and KATAN have been provided.

Table 1: TinyJAMBU Variants

AEAD Variants of TinyJAMBU Mode	Size in bits				Number of Rounds in	
	State	Key	Nonce	Tag	P_l	$\hat{P}_l$
TinyJAMBU-128	128	128	96	64	640	1024
TinyJAMBU-192	128	192	96	64	640	1152
TinyJAMBU-256	128	256	96	64	640	1280

2.1 TinyJAMBU

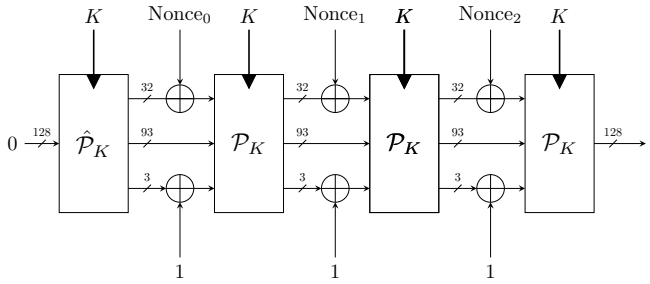

Fig. 2: The Initialization of TinyJAMBU [9]

TinyJAMBU is a variant of JAMBU that was selected as a finalist in the NIST Lightweight Cryptography competition. It uses a 128-bit NLFSR-based keyed permutation with 128-bit state size and 32-bit message block size. It offers better security than JAMBU and duplex mode with nonce reuse. The permutation, denoted by P_l^K, has l rounds and supports key sizes of 128, 192, or 256 bits. In short, we use $\mathcal{P}_l$ to denote an l-round keyed permutation of TinyJAMBU throughout the paper. The i^{th} round of the $\mathcal{P}_l$ permutation transforms a 128-bit state to another 128-bit state. The transformation is defined by $sf = s_0 \oplus s_{47} \oplus \overline{s_{70} s_{85}} \oplus s_{91} \oplus k_{i \mod |K|}$, where s_f is the transformed state and $k_{i \mod |K|}$ is the secret key. The permutation is shown in Figure 1. The TinyJAMBU mode has three variations named TinyJAMBU-128, TinyJAMBU-192, and TinyJAMBU-256, with specifications listed in Table 1.

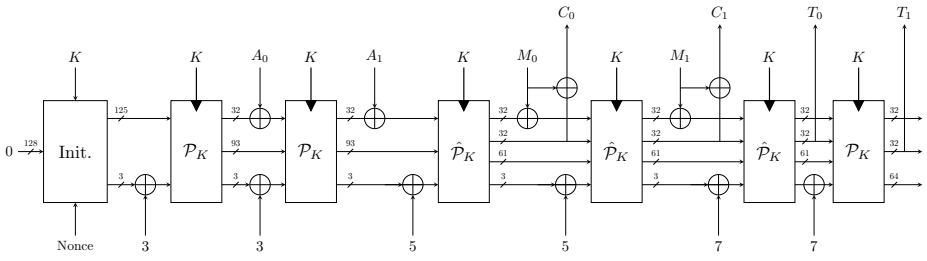

Fig. 3: The Description of TinyJAMBU Mode [9]

The TinyJAMBU encryption process has four stages: Initialization, Associated Data Processing, Encryption, and Finalization. In the Initialization stage, the state is initialized through key and nonce setup. In the Associated Data Processing stage, each data block is processed by XORing with the state, updating the state with P_l, and XORing the associated data block with the updated state. In the Encryption stage, each message block is encrypted by XORing with the state, updating the state with $\hat{P}_l$, injecting the message block into the first block of the state, and producing the ciphertext by XORing the message block with the second block of the state. In the Finalization stage, the authentication tag $T = T_0||T_1$ is generated by XORing with the state, updating the state with $\hat{P}_l$, extracting T_0 from the state, XORing again with the state, updating the state with P_l, and extracting T_2 from the resulting state. The overall structure of the TinyJAMBU mode is depicted in Figure 3, where the permutations P_l and $\hat{P}_l$ are specified in Table 1.

2.2 KATAN

The KATAN family is a very efficient NLFSR-based hardware-oriented block cipher with three variants, namely KATAN32, KATAN48, KATAN64 correspond to 32, 48, and 64-bit block sizes. All these variants have 254 rounds and use the

non-linear functions $\mathcal{NF}_1$ and $\mathcal{NF}_2$. Also, they use the same LFSR-based key schedule which takes an 80-bit key as an input. The general structure of the KATAN cipher is as follows. First, the plaintext is loaded into two registers L_1 and L_2. In each round, several bits are taken from the registers to fed into the non-linear functions, and finally, the output of $\mathcal{NF}_1$ and $\mathcal{NF}_2$ is loaded to the least significant bits to the registers. The key schedule function expands an 80-bit user-provided key k_i $(0 \leq i < 80)$ into a 508-bit subkey sk_i $(0 \leq i < 508)$ by the following linear operations,

$$sk_i = \begin{cases} k_i, & 0 \leq i < 80 \\ k_{i-80} \oplus k_{i-61} \oplus k_{i-50} \oplus k_{i-13}, & 80 \leq x < 508. \end{cases}$$

Also, the two non-linear functions are defined as follows:

$$\mathcal{NF}_1(L_1) = L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \cdot L_1[x_4]) \oplus (L_1[x_5] \cdot IR) \oplus k_a$$

$$\mathcal{NF}_2(L_2) = L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \cdot L_2[y_4]) \oplus (L_2[y_5] \cdot L_2[y_6])) \oplus k_b,$$

where IR is the pre-defined round constant value (see the specification in [5]), and k_a, k_b are the two subkey bits. The selection of the bits $x_i, 1 \leq i \leq 5$ and $y_i, 1 \leq i \leq 6$ are defined for each variant independently, and are listed in Table 2. For KATAN32, the i-th round function is depicted in Figure 4, where $k_a \leftarrow k_{2i}$ and $k_b \leftarrow k_{2i+1}$. Finally, after 254 rounds, the values of registers are output as a ciphertext. For KATAN48, the non-linear functions $\mathcal{NF}_1$ and $\mathcal{NF}_2$ are applied twice in one round of the cipher, i.e., the first pair of $\mathcal{NF}_1$ and $\mathcal{NF}_2$ is applied, and then after the update of the registers, they have applied again using the same subkeys. Similarly, in KATAN64, each round applies $\mathcal{NF}_1$ and $\mathcal{NF}_2$ three times with the same key bits. More details about the specification of KATAN-family of ciphers can be found in [5].

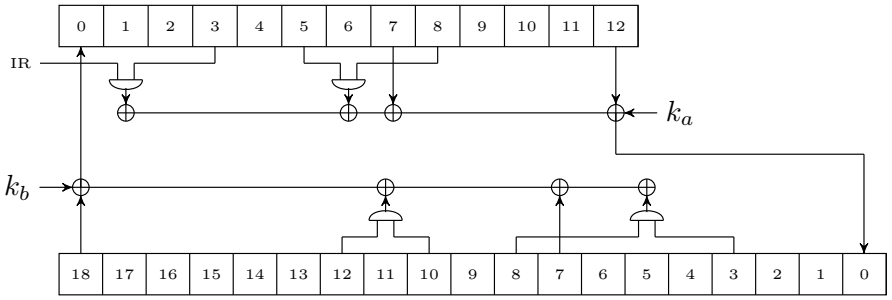

Fig. 4: Round Function of KATAN32

Table 2: Parameters of KATAN Variants

KATAN Variants	$\mid L_1 \mid$	$\mid L_2 \mid$	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4	y_5	y_6
KATAN32	13	19	12	7	8	5	3	18	7	12	10	8	3
KATAN48	19	29	18	12	15	7	6	28	19	21	13	15	6
KATAN64	25	39	24	15	20	11	9	38	25	33	21	14	9

3 Introducing DEEPAND Modeling

In this section, we introduce the basic idea behind DEEPAND which attempts to generalize the way AND gates are modeled by proposing a systematic way to capture the correlation between AND gates. We first revisit the difference distribution of the output of an AND gate under certain restrictions on the inputs. We then show how the refined model given in [9] can be interpreted as a special case of DEEPAND. We later show how DEEPAND can better capture correlations in both single and multiple AND based NLFSRs.

Our first goal is to look at the difference distribution of an AND gate. Consider an AND gate A_1 with (a, b) as its input, $(\Delta a, \Delta b)$ as its input difference, and Δz as its output difference. Then, the output difference Δz can be expressed as shown in Equation 1.

$$\begin{aligned} \Delta z &= A_1(a, b) \oplus A_1(a \oplus \Delta a, b \oplus \Delta b) \\ &= (a \cdot b) \oplus (a \oplus \Delta a) \cdot (b \oplus \Delta b) \\ &= (a \cdot \Delta b) \oplus (b \cdot \Delta a) \oplus (\Delta a \cdot \Delta b) \end{aligned} \tag{1}$$

The distribution of Δz corresponding to all values of (a, b) and $(\Delta a, \Delta b)$, is shown in Table 3 from where it is evident that for a given non-zero input difference $(\Delta a, \Delta b)$ of A_1, $\Pr(\Delta z = 0) = \Pr(\Delta z = 1) = 2^{-1}$, i.e., it behaves uniformly. However, under certain conditions, Δz behaves non-uniformly An example for this non-uniform behavior is shown in Example 1. From Table 3, the following observations have been made[4].

Example 1. $\Pr[\Delta z = 0 | (a = 0, \Delta a = 0, \Delta b = 1)] = 1$

Observation 1. *If the value of a, b, Δa, and Δb are known, then Δz becomes deterministic.*

Observation 2. *If $\Delta a = 0$, $\Delta b = 1$ and the value of a is known, then Δz can be determined with probability 1. Similarly, if $\Delta a = 0$, $\Delta b = 1$ and the value of Δz is known, then 'a' can be guessed deterministically.*

Remark. If $\Delta a = 0$, $\Delta b = 1$, then from Equation 1, $\Delta z = a$. This means, for an input difference $(\Delta a, \Delta b) = (0, 1)$, if a is known, then Δz is also known and vice versa.

[4]Observation 1 may seem trivial but it has been included for the sake of completeness.

Table 3: Difference Distribution of AND Gate

a	b	Δa	Δb	Δz
0	0	0	0	0
		0	1	0
		1	0	0
		1	1	1
0	1	0	0	0
		0	1	0
		1	0	1
		1	1	0
1	0	0	0	0
		0	1	1
		1	0	0
		1	1	0
1	1	0	0	0
		0	1	1
		1	0	1
		1	1	1

Observation 3. *If $\Delta a = 1$, $\Delta b = 0$ and the value of b is known then Δz can be determined with probability 1. Similarly, if $\Delta a = 1$, $\Delta b = 0$ and the value of Δz is known, then b can be guessed deterministically.*

Remark. The explanation is similar to the explanation of Observation 2.

Based on the above observations from Table 3, it is evident that the distribution Δz directly depends on the input bits a, b when the input difference $(\Delta a, \Delta b)$ is fixed. According to Equation (1), the Δz can be re-written in the following way.

$$\Delta z = \begin{cases} 0, & \text{if } (\Delta a, \Delta b) = (0,0), \\ a, & \text{if } (\Delta a, \Delta b) = (0,1), \\ b, & \text{if } (\Delta a, \Delta b) = (1,0), \\ a \oplus b \oplus 1, & \text{if } (\Delta a, \Delta b) = (1,1), \end{cases} \tag{2}$$

We refer to the view captured by Equation (2) as DEEPAND. With the DEEPAND view of Δz in place, we are in a position to revisit the refined model proposed by Saha *et al.* [9] for TinyJAMBU.

3.1 Refined Modeling as a Special Case of DEEPAND

We start by restating the observation made by Saha *et al.* in the so-called *Refined* Model. Consider two AND gates A_1 with (a, b), and A_2 with (b, c) as their inputs, i.e., they both share a common input as b and hence referred to as *correlated*. Also, let $(\Delta a, \Delta b)$, $(\Delta b, \Delta c)$ are the input differences, and $\Delta z_1, \Delta z_2$ are the output differences of A_1, A_2 respectively. The primary observation in [9] was that

when $(\Delta a, \Delta b) = (1, 0)$ and $(\Delta b, \Delta c) = (0, 1)$ then $\Delta z_1 = \Delta z_2 = b$. This implies that, for two correlated AND gates A_1 and A_2, when $(\Delta a, \Delta b, \Delta b) = (1, 0, 1)$, then both the output differences are 0 with probability 2^{-1} or 1 with probability 2^{-1}. Whereas, for two un-correlated AND gates this figure would have been 2^{-2}. Lemma 1 gives a separate perspective on the two correlated AND gates based on the Observations 2 and 3.

Lemma 1. *Let the input difference to two correlated AND gates be (Δa, Δb) and (Δb, Δc) respectively and corresponding output differences be Δz_1 and Δz_2 respectively. If $\Delta a = 1$, $\Delta b = 0$, $\Delta c = 1$, then* $\Pr[\Delta z_1 = \Delta z_2] = 2^{-1}$.

Proof. First of all, the value of Δz_1 is computed first. Thus, for $(\Delta a, \Delta b) = (1, 0)$, it can be concluded that $\Delta z_1 = b$ according to Observation 3. Also, for the second AND gate with $(\Delta b, \Delta c) = (0, 1)$, $\Delta z_2 = b$ (from Observation 2). Hence, we have, $\Pr[\Delta z_1 = \Delta z_2] = \Pr(b) = 2^{-1}$. □

Remark. It is worth mentioning that despite being one of first attempts In [9], the authors do not explicitly give a systematic way to capture the correlation between two AND gates. Moreover, the authors have not considered the Observations 2 and 3 in their refined model. In this work, these two observations along with Observation 1 are exploited to penetrate more number of rounds for NLFSR-based ciphers.

4 DEEPAND Modeling of NLFSR-based Ciphers

A NLFSR is a shift register whose input bit, often called a *feedback bit*, is a non-linear function of its previous state. In this section, we will first review some different classes of NLFSRs based on the number of AND gates that are used to define a non-linear feedback function. We will then state the explicit form of these NLFSRs. Finally, we will describe how DEEPAND leads to a general attack framework to capture correlations among single and multiple AND gates.

4.1 Case-1: Single AND Based NLFSR

Any n-bit cipher based on the NLFSR-based keyed permutation with single AND gate can be further classified into two cases. In each round of the cipher, the first one is to feed the the feedback bit using non-linear function to the most significant bit (msb) in the state and then shift each bit towards the least significant bit (lsb) (see Figure 1). Similarly, for the second one, compute the feedback bit and feed into the lsb and then shift each bit towards msb. We now give the explicit form of these two NLFSRs.

4.1.1 Computing Forward Differential Consider an n-bit NLFSR-based cipher $\overleftarrow{\mathcal{C}}$ with s^0 being its initial state value, where $s^0 = (s_0^0, s_1^0, \cdots, s_{n-1}^0)$. Then, for each round number $i, 1 \leq i \leq l$, the feedback bit f^i is computed first, in the following way:

$$f^i \leftarrow s_0^{i-1} \oplus s_{j_1}^{i-1} \oplus \cdots \oplus s_{j_m}^{i-1} \oplus s_{u_1}^{i-1} s_{v_1}^{i-1} \oplus K_{(i-1) \bmod |K|}.$$

where $0, j_1, \cdots, j_m$ are the tap bit positions of the NLFSR and u_1, v_1 $(u_1 < v_1)$ are the input bits to the AND gate. Then, the state bits in the next round (round $i+1$) are updated as follows:

$$s_j^i = \begin{cases} s_{j+1}^{i-1}, & \text{for } 0 \le j \le (n-2) \\ f^i, & \text{for } j = n-1 \end{cases}$$

Consider a similar cipher $\overrightarrow{\mathcal{C}}$, whose tap bits are the same as that of $\overleftarrow{\mathcal{C}}$. The only difference is that the bits are shifted in opposite direction as that of $\overleftarrow{\mathcal{C}}$ and in the feedback function s_{n-1}^{i-1} is XOR-ed instead of s_0^{i-1}. The cipher $\overrightarrow{\mathcal{C}}$ is called *reverse-fed* cipher of $\overleftarrow{\mathcal{C}}$. The feedback bit f^i for $\overrightarrow{\mathcal{C}}$ is computed as follows:

$$f^i = s_{j_1}^{i-1} \oplus \cdots \oplus s_{j_m}^{i-1} \oplus s_{n-1}^{i-1} \oplus s_{u_1}^{i-1} s_{v_1}^{i-1} \oplus K_{(i-1) \bmod |K|}.$$

and

$$s_j^i = \begin{cases} s_{j-1}^{i-1}, & \text{for } 1 \le j \le (n-1) \\ f^i, & \text{for } j = 0 \end{cases}$$

To find the differential trails for such ciphers $\overleftarrow{\mathcal{C}}$, $\overrightarrow{\mathcal{C}}$, the probability is only paid for the active AND gates through rounds. Thus, given an l round differential trail, the overall probability can be calculated by counting only the total number of active ANDs in the trail. Also, it is to be noted that, the whole state bits become unknown after n number of rounds. In another way, we can say that exactly $n-i$ number of state bits are still known for the initial i $(1 \le i \le n)$ rounds. Therefore, in chosen plaintext scenario, we can deterministically bypass some of the active AND gates by fixing the message bits for up to some initial i $(\le n)$ rounds. This characteristic of any NLFSR-based ciphers $\overleftarrow{\mathcal{C}}$,$\overrightarrow{\mathcal{C}}$ is described in the following lemma.

Lemma 2. *For cipher $\overrightarrow{\mathcal{C}}$, forward differential trail for the first (u_1+1) rounds is completely free. For the next (v_1-u_1) rounds, if the input differential to the AND gate is 0 and 1 (i.e., $\Delta s_{u_1} = 0, \Delta s_{v_1} = 1$) then the output of the AND gate can be determined with probability 1 (**conditionally free**). Similarly, for a cipher $\overleftarrow{\mathcal{C}}$, $(n-v_1)$ rounds are completely free and (v_1-u_1) rounds are conditionally free.*

Proof. As both the inputs to AND gate are known for the first (u_1+1) rounds, the output difference of the AND gate can be bypassed with probability 1. For the next (v_1-u_1) rounds, the u_1-th bit in the state, i.e., s_{u_1} is still known to us from the given input message. Therefore, at the intermediate rounds i $(u_1+1 < i \le v_1)$ if the input difference corresponding to the AND gate becomes $(0,1)$, i.e., $\Delta s_{u_1} = 0$ and $\Delta s_{v_1} = 1$, then by Observation 2 the output difference of the AND gate can be deterministically bypassed. The proof for the cipher $\overleftarrow{\mathcal{C}}$ follows a similar approach. □

Note that, in the chosen plaintext attack model (CPA), Lemma 2 can be exploited by carefully choosing the message bits. This, in turn, reduces the *degrees of freedom* of the message space.

4.1.2 Computing Backward Differential While computing the backward differential for a cipher $\overleftarrow{\mathcal{C}}$, the feedback function remains almost the same except only the index of the bits are changed. Consider that the initial state is t^0 and the intermediate state after the i^{th} round is t^i. Then the feedback bit, f^i for the i^{th} round is computed in the following way:

$$f^i \leftarrow t^{i-1}_{j_1-1} \oplus \cdots \oplus t^{i-1}_{j_m-1} \oplus t^{i-1}_{n-1} \oplus t^{i-1}_{u_1-1} t^{i-1}_{v_1-1} \oplus K_{(i-1) \bmod |K|}$$

and the state bits are updated as follows:

$$t^i_j = \begin{cases} t^{i-1}_{j-1}, & \text{for } 1 \le j \le (n-1) \\ f^i, & \text{for } j = 0. \end{cases}$$

Similarly, for cipher $\overrightarrow{\mathcal{C}}$, the feedback bit is computed as

$$f^i = t^{i-1}_{j_1+1} \oplus \cdots \oplus t^{i-1}_{j_m+1} \oplus t^{i-1}_0 \oplus t^{i-1}_{u_1+1} t^{i-1}_{v_1+1} \oplus K_{(i-1) \bmod |K|}$$

and

$$t^i_j = \begin{cases} t^{i-1}_{j+1}, & \text{for } 0 \le j \le (n-2) \\ f^i, & \text{for } j = n-1. \end{cases}$$

Lemma 3. *For cipher $\overrightarrow{\mathcal{C}}$, backward differential trail for first $(n - v_1 - 1)$ rounds is completely free. The next $(v_1 - u_1)$ rounds are conditinally free. Similarly, for cipher $\overleftarrow{\mathcal{C}}$, the first $(u_1 - 1)$ rounds are completely free whereas the next $(v_1 - u_1)$ rounds are conditionally free.*

Proof. The proof is quite similar to that of Lemma 2 □

4.2 Case-2: Multiple AND Based NLFSR

Consider an n-bit NLFSR-based block cipher $\overrightarrow{\mathcal{D}}$ with the initial state value as $s^0 = (s^0_0, s^0_1, \cdots, s^0_{n-1})$. At each round i, the feedback bit f^i is computed in the following way.

$$f^i \leftarrow s^{i-1}_{j_1} \oplus \cdots \oplus s^{i-1}_{j_m} \oplus s^{i-1}_{n-1} \oplus s^{i-1}_{u_1} s^{i-1}_{v_1} \oplus \cdots \oplus s^{i-1}_{u_h} s^{i-1}_{v_h} \oplus K_{i-1},$$

where

- k^{i-1} is the key bit used in the i^{th} round,
- $j_1, \cdots, j_m, n-1$ are the taps of the NLFSR,
- u_j, v_j are the inputs to the AND gate A_j such that $u_j < v_j \le n-1, 1 \le j \le h$,
- $j_1 < j_2 \implies u_{j_1} < u_{j_2}$.

Also, the state in the next round is updated in the following way.

$$s_j^i = \begin{cases} s_{j-1}^{i-1}, & \text{for } 1 \le j \le (n-1) \\ f^i, & \text{for } j = 0. \end{cases}$$

Lemma 4. *For a cipher $\overrightarrow{\mathcal{D}}$, in the forward differential, the output of gate A_j is deterministic for the first (u_j+1) rounds. For the next (v_j-u_j) rounds, the output of the AND gate is conditionally free. Similarly, for a cipher $\overleftarrow{\mathcal{D}}$, the* reverse-feed *cipher of $\overrightarrow{\mathcal{D}}$, the output of gate A_j is deterministic for the first $(n-v_j)$ rounds and conditionally free for the next (v_j-u_j) rounds.*

Proof. For cipher $\overrightarrow{\mathcal{D}}$, as $s_{u_j}^i$ and $s_{v_j}^i$ are known for $0 \le i \le u_j$, so ΔA_j can be deterministically computed for the first (u_j+1) number of rounds as both inputs to the AND gate are known.

Suppose, during the intermediate rounds, $s_{v_j}^i$ is known and $s_{u_j}^i$ is unknown for $u_j+1 \le i \le v_j$ (round number u_j+2 to v_j+1). If $\Delta s_{v_j}^i = 0$ and $\Delta s_{u_j}^i = 1$, then by Observation 3, $\Delta A_j = s_{v_j}^i$. Hence, for round u_j+1 to v_j, ΔA_j can be determined with probability 1 when such conditions are met.

For cipher $\overleftarrow{\mathcal{D}}$, $s_{u_j}^i$ and $s_{v_j}^i$ are known for $0 \le i \le (n-v_j-1)$. Hence, ΔA_j can be determined completely free for first $(n-v_j)$ rounds. $s_{u_j}^i$ is known and $s_{v_j}^i$ is unknown for $(n-v_j) \le i \le (n-u_j-1)$ (round number $(n-v_j+1)$ to $(n-u_j)$). If $\Delta s_{v_j}^i = 1$ and $\Delta s_{u_j}^i = 0$, then by Observation 2, $\Delta A_j = s_{u_j}^i$. Therefore, for next (v_j-u_j) rounds, ΔA_j can be determined with probability 1 when such conditions are met. □

In the same fashion, computing the backward differential, the feedback bit f^i for i^{th} round is computed as

$$f^i \leftarrow t_{l_1+1}^{i-1} \oplus \cdots \oplus t_{l_m+1}^{i-1} \oplus t_0^{i-1} \oplus t_{u_1+1}^{i-1} t_{v_1+1}^{i-1} \oplus \cdots \oplus t_{u_h+1}^{i-1} t_{v_h+1}^{i-1} \oplus k'^{i-1}$$

and the state in the next round is updated as

$$t_j^i = \begin{cases} t_{j-1}^{i-1}, & \text{for } 0 \le j \le (n-2) \\ f^i, & \text{for } j = n-1. \end{cases}$$

Lemma 5. *For cipher $\overrightarrow{\mathcal{D}}$, in the backward differential, the output of gate A_j is deterministic for first $(n-v_j-1)$ rounds. For the next (v_j-u_j) rounds, the output of the gate is conditionally free. Similarly, for a cipher $\overleftarrow{\mathcal{D}}$, the* reverse-feed *cipher of $\overrightarrow{\mathcal{D}}$, in the backward differential the output of gate A_j is deterministic for first (u_j) rounds and conditionally free for next (v_j-u_j) rounds.*

Proof. For cipher $\overrightarrow{\mathcal{D}}$, as $t_{u_j+1}^i$ and $t_{v_j+1}^i$ are known for $0 \le i \le n-v_j-2$, so ΔA_j can be deterministically computed for first $(n-v_j-1)$ number of rounds as both inputs to the AND gate are known.

$t^i_{v_j+1}$ is known and $t^i_{u_j+1}$ is unknown for $n - v_j - 1 \leq i \leq n - u_j - 2$ (round number $n-v_j$ to $n-u_j-1$). If $\Delta t^i_{u_j+1} = 0$ and $\Delta t^i_{v_j} = 1$, then by Observation 2, $\Delta A_j = t^i_{u_j+1}$. Hence, for round $n - v_j$ to $n - u_j - 1$, ΔA_j can be determined with probability 1 when such conditions are met.

In similar way, it can be proved for $\overleftarrow{\mathcal{D}}$. □

4.3 Generalization of Chained ANDs

Consider an n-bit cipher $\mathcal{C}$ with $(s_{u_1}, s_{u_2}), (s_{u_2}, s_{u_3})$ and $(\Delta s_{u_1} = 1, \Delta s_{u_2} = 0), (\Delta s_{u_2} = 0, \Delta s_{u_3} = 1)$ are respectively two sequential inputs and their differences to the AND gate. Suppose we have differential trail and at the round i, we see that the input difference $\Delta s_{u_1} = 1, \Delta s_{u_2} = 0$ happens at the AND gate and Δz be the coresponding output difference. Then, according to Observation 3, the internal state bit s_{u_2} will be revealed due to the relation $\Delta z = s_{u_2}$. Thus, after the $(u_2 - u_1 - 1)$ number of rounds, i.e., at the round $i + (u_2 - u_1 - 1)$, $\Delta s_{u_2} = 0, \Delta s_{u_3} = 1$ becomes the input difference to the AND gate. In this case, by Observation 2, this active AND gate will be freely bypassed as we know the bit value s_{u_2}. Therefore, if the subsequent input differences to the AND gate are $1, 0, 1$ then instead of paying the probability of $\frac{1}{4}$, we only have to pay the probability of $\frac{1}{2}$. In another way, we can say that when this subsequent $1, 0, 1$ bit difference arise in the AND gate, we will count it as one active AND. Because, out of two subsequent active ANDs, we only pay the probability for the first one (i.e., when $\Delta s_{u_1} = 1, \Delta s_{u_2} = 0$) whereas the second (where $\Delta s_{u_2} = 0, \Delta s_{u_3} = 1$) one will pass with probability 1.

In the refined modeling paper [9] introduced for TinyJAMBU, the authors added some extra constraints in the simple MILP model and recorded all the two subsequent ANDs with $1, 0, 1$ bit differences which helps to increase the overall probability of the differential trail. We named this kind of two subsequent ANDs with $1, 0, 1$ bit differences as Chained AND Bit Pattern (BAND). Now, if we consider a NLFSR with multiple ANDs-based cipher, then there might arise more than two subsequent ANDs with various bit difference patterns that might significantly increase the overall probability of the trail and we named it as Multiple AND Bit Pattern (MAND). Before going to define it, we give one example to show how MAND increases the probability in the trail.

Example 2. Suppose, we have an n-bit cipher $\overrightarrow{\mathcal{D}}$ with two ANDs, where $n = 32$ and $(3, 8), (10, 12)$ are the two different AND's input positions in the NLFSR state. At the round i, we assume that a particular bit difference $\Delta s_8 = 1$, $\Delta s_5 = 1$, $\Delta s_3 = 0$, $\Delta s_1 = 1$, and $\Delta s_{-2} = 0$ happens in the state. Also, we choose the bit difference 0 at the third position in the state as a pivot. In the subsequent rounds, this pivot will activate some related AND gates, and then it helps to freely pass some subsequent ANDs in the following way.

1. At round i, since $\Delta s_8 = 1, \Delta s_3 = 0$ happens, we get the information of the state bit at the pivotal position according to Observation 2.

2. Then, at the round $i+7$, the pivot goes to the bit position 10 and activate the second AND gate as $\Delta s_{12} = 1, \Delta s_{10} = 0$. Thus, according to the Observation 3, this active AND will be freely passed.
3. Similarly, when the pivot goes to the 12-th position in the state at the round $i+9$, the AND will be passed detrministically according to the Observation 2.

The above steps are summarized in the Table 4a. In this example, we have to only pay the probability of 2^{-1} instead of 2^{-3}, as the total number of active ANDs subject to the pivot is 3.

Table 4: Examples of MAND and BAND

(a) An Example of MAND

Round	NLFSR State Bit Positions[1]						
	Δs_{12}	Δs_{10}	Δs_8	Δs_5	Δs_3	Δs_1	Δs_{-2}
i	0	0	1	1	0	1	0
$i+5$	0	1	0	-	0	-	-
$i+7$	1	0	1	0	-	-	-
$i+9$	0	1	-	-	-	-	-

(b) An Example of BAND

Round	NLFSR State Bit Positions[2]		
	Δs_{21}	Δs_{24}	Δs_{27}
i	1	0	1
$i+3$	-	1	0

Let us denote Δs_j^i to be the state difference Δs_j at round i and s_j^i to be the state value s_j at round i. Also, for ciphers like $\overrightarrow{\mathcal{C}}$ and $\overrightarrow{\mathcal{D}}$, we use $\Delta s_{u_1}^i$ to be the the pivotal bit difference at the position u_1, the first AND bit position. We now furnish the formal definitions of BAND and MAND for ciphers $\overrightarrow{\mathcal{C}}$ and $\overrightarrow{\mathcal{D}}$ respectively. They can be defined similarly for ciphers like $\overleftarrow{\mathcal{C}}$ and $\overleftarrow{\mathcal{D}}$.

Definition 1 (Bi-AND Bit Pattern - BAND). *Consider the cipher $\overrightarrow{\mathcal{C}}$ with (u_1, v_1) as its input position of the AND gate. BAND of a pivotal bit difference $(\Delta s_{v_1}^i = 0)$ is denoted by $\mathcal{B}_i^{\overrightarrow{\mathcal{C}}}$ and is defined as a bit string in the following way.*

$$\mathcal{B}_i^{\overrightarrow{\mathcal{C}}} = l_1 \overbrace{\Delta s_{u_1}^i}^{\text{pivot}} r_1, \text{ where } \begin{cases} l_1 = \Delta s_{u_1}^i & \text{when } \Delta s_{u_1}^i \text{ is at } u_1 \\ r_1 = \Delta s_{u_1}^{i+(v_1-u_1)} & \text{when } \Delta s_{u_1}^i \text{ is at } v_1 \end{cases}$$

Example 3. Consider an NLFSR-based block cipher $\overrightarrow{\mathcal{C}}$ with $n = 32$ and $(24, 27)$ as the inputs to the AND gate A_1. Let us assume that, at round $i(> 42)$, particular

[1] The bit values in $\Delta s_8, \Delta s_5, \Delta s_3, \Delta s_1$, and Δs_{-2} are shown in orange, green, red, violet, and brown colors respectively.

[2] The bit values in $\Delta s_{21}, \Delta s_{24}$, and Δs_{27} are shown in blue, red, and green colors respectively.

bit differences of $\Delta s_{21} = \mathbf{1}$, $\Delta s_{24} = \mathbf{0}$, and $\Delta s_{27} = \mathbf{1}$ occur in the state (see Table 4b). Then the BAND of the pivot $\Delta s^i_{24} = \mathbf{0}$, $\mathcal{B}^{\vec{\mathcal{C}}}_i$ is given as below.

$$\begin{aligned}\mathcal{B}^{\vec{\mathcal{C}}}_i = l_1 \boxed{\Delta s^i_{24}} r_1 &= \Delta s^{i+3}_{24} \boxed{\Delta s^i_{24}} \Delta s^i_{27} \\ &= \Delta s^i_{21} \boxed{\Delta s^i_{24}} \Delta s^i_{27}, \quad [\because \Delta s^{i+a}_b = \Delta s^i_{b-a}]\end{aligned}$$

Definition 2 (Multiple AND Bit Pattern - MAND). *Consider the cipher $\vec{\mathcal{D}}$ with $(u_1, v_1), \cdots, (u_h, v_h)$ denoting respectively input positions to h number of AND gates. The MAND of a pivotal bit difference $(\Delta s^i_{u_1} = 0)$ is denoted by $\mathcal{M}^{\vec{\mathcal{D}}}_i$ and is defined as a $(2h+1)-$bit string in the following way:*

$$\mathcal{M}^{\vec{\mathcal{D}}}_i = l_h\ l_{h-1}\ \cdots\ l_1\ \overbrace{\Delta s^i_{u_1}}^{\text{pivot}}\ r_1\ \cdots\ r_{h-1}\ r_h, \text{ where, } \exists \text{ some } p \in \{1, \cdots, h\}$$
$$\text{such that } \begin{cases} l_p = \Delta s^{i+(u_p-u_1)}_{v_p} & \text{when } \Delta s^i_{u_1} \text{ is at } u_p \\ r_p = \Delta s^{i+(v_p-u_1)}_{u_p} & \text{when } \Delta s^i_{u_1} \text{ is at } v_p \end{cases}$$

When there is exactly a single $p \in \{1, \cdots, h\}$ such that $l_p = r_p = 1$, $\mathcal{M}^{\vec{\mathcal{D}}}_i$ collapses to a BAND which can hence be interpreted as a specific instance of a MAND. With the above formalisms in place, we can now revisit Example 2 where the MAND of the pivot $\Delta s^i_3 = \mathbf{0}$ can be captured as below.

$$\begin{aligned}\mathcal{M}^{\vec{\mathcal{D}}}_i = l_2 l_1 \boxed{\Delta s^i_3} r_1 r_2 &= \Delta s^{i+9}_{10} \Delta s^{i+5}_3 \boxed{\Delta s^i_3} \Delta s^i_8 \Delta s^{i+7}_{12} \\ &= \Delta s^i_1 \Delta s^i_{-2} \boxed{\Delta s^i_3} \Delta s^i_8 \Delta s^i_5, \quad [\because \Delta s^{i+a}_b = \Delta s^i_{b-a}]\end{aligned}$$

We can demonstrate that the probability of a particular trail in an AND-based cipher ($\vec{\mathcal{D}}$) can be significantly increased due to the occurrence of MANDs. To detect the MANDs, we need to introduce variables that represent the output differences of the AND gates in the intermediate rounds. For $p \in \{1, \cdots, h\}$, we define $\Delta A^{i,u_p}_p$ and $\Delta A^{i,v_p}_p$ as the output differences of AND gate A_p when the pivotal bit difference $\Delta s^{u_1}_i$ moves to positions u_p and v_p, respectively. When certain MANDs occur at the intermediate rounds (out of a total of $2h+1$ bit patterns), we can establish relationships among $s^i_{u_1}$, $\Delta A^{i,u_p}_p$, and $\Delta A^{i,v_p}_p$. These relationships can help us understand how the occurrence of MANDs affects the trail probability. We have already established how BAND is a special case of MAND. The following lemma captures the behavior of BAND with regards to variables $\Delta A^{i,u_p}_p$, and $\Delta A^{i,v_p}_p$ introduced above. Later we use the notion of $\mathcal{M}^{\vec{\mathcal{P}}}_i$-weight in subsequent lemmas to highlight the gain in trail propagation probability that ensues due to MANDs.

Lemma 6. *Consider a BAND with $\mathcal{B}^{\vec{\mathcal{C}}}_i = l_1\ \Delta s^i_{u_1}\ r_1$. If $l_1 = r_1 = 1$, then $\Delta A^{i,u_1}_1 = \Delta A^{i,v_1}_1$.*

Proof. According to the Observation 3, if $l_1 = 1$ and $\Delta s^i_{u_1} = 0$, then $\Delta A^{i,u_1}_1 = s^i_{u_1}$. Similarly, as $r_1 = 1$ and $\Delta s^i_{u_1} = 0$, we have $\Delta A^{i,v_1}_1 = s^i_{u_1}$. Hence, we can conclude that $\Delta A^{i,u_1}_1 = \Delta A^{i,v_1}_1$. □

Definition 3 (MAND-weight). *The weight of a MAND $\mathcal{M}^{\vec{\mathcal{P}}}_i$, denoted by $wt(\mathcal{M}^{\vec{\mathcal{P}}}_i)$ captures its Hamming-weight.*

Lemma 7. *Consider a MAND with $\mathcal{M}^{\vec{\mathcal{P}}}_i = l_h \cdots l_1 \ \Delta s^i_{u_1} \ r_1 \cdots r_h$ and $wt(\mathcal{M}^{\vec{\mathcal{P}}}_i) = p + q$. For $\{w_1, \cdots, w_p\}$ and $\{y_1, \cdots, y_q\} \subset \{1, \cdots, h\}$,*

$$\begin{aligned} & l_{w_1} = \cdots = l_{w_p} = r_{y_1} = \cdots = r_{y_q} = 1 \\ \implies & \Delta A^{i,u_{w_1}}_{w_1} = \cdots = \Delta A^{i,u_{w_p}}_{w_p} = \Delta A^{i,v_{y_1}}_{y_1} = \cdots = \Delta A^{i,v_{y_q}}_{y_q} \end{aligned}$$

Proof. By Observation 3, if $l_{w_g} = 1$ and $\Delta s^i_{u_1} = 0$ then $\Delta A^{i,u_{w_g}}_{w_g} = s^i_{u_1}$ holds $\forall g \in \{1, \cdots, p\}$. Similarly, as $r_{y_g} = 1$ and $\Delta s^i_{u_1} = 0$, $A^{i,u_{y_g}}_{y_g} = s^i_{u_1}$ holds $\forall g \in \{1, \cdots, q\}$. Hence, we can conclude that $\Delta A^{i,u_{w_1}}_{w_1} = \cdots = \Delta A^{i,u_{w_p}}_{w_p} = \Delta A^{i,v_{y_1}}_{y_1} = \cdots = \Delta A^{i,v_{y_q}}_{y_q}$. □

Lemma 8. *Let $wt(\mathcal{M}^{\vec{\mathcal{P}}}_i) = m$ and $m \geq 2$. Then the subsequent output differences of m active AND gates can be restricted to probability 2^{-1} instead of 2^{-m}.*

Proof. As $wt(\mathcal{M}^{\vec{\mathcal{P}}}_i) = m$, then Lemma 7 implies that output differences of m AND gates should be equal to $s^i_{u_1}$. Thus the output differences are correlated and the joint propagation probability increases from 2^{-m} to 2^{-1}. □

4.4 Experimental Evidence of MAND

The effect of MAND is observed in the 60-round related-key differential trail of KATAN48. The trail is listed in Table 15 with input difference 0x820031400000 and output difference 0x00018000c000.

The feedback function $f_b(L_2)$ of KATAN48 consists of two AND gates. $L_2[6]$ and $L_2[15]$ are inputs to one AND gate whereas $L_2[13]$ and $L_2[21]$ are inputs to another AND gate. Using the NLFSR description from Section 4.2, the following values can be fixed.

$$u_1 = 6 \qquad v_1 = 15 \qquad u_2 = 13 \qquad v_2 = 21$$

Now we find the MAND with respect to the pivot $\Delta s^{103}_{u_1} (= \Delta s^{103}_6)$. In particular, we are finding the expression for $\mathcal{M}^{\vec{\mathcal{P}}}_{103}$. ($\vec{\mathcal{P}} =$ KATAN48). From Definition 2,

$$\mathcal{M}^{\vec{\mathcal{P}}}_{103} = l_2 l_1 s^{103}_6 r_1 r_2$$

The values for l_2, l_1, r_2, r_1 are needed to be computed. Again from Definition 2, in this case $p \in \{1, 2\}$. Thus,

$$l_2 = \Delta s_{v_2}^{103+(u_2-u_1)} = \Delta s_{21}^{103+(13-6)} = \Delta s_{21}^{110} = 0$$
$$r_2 = \Delta s_{u_2}^{103+(v_2-u_1)} = \Delta s_{13}^{118} = 1$$
$$l_1 = \Delta s_{v_1}^{103+(u_1-u_1)} = \Delta s_{15}^{103} = 0$$
$$r_1 = \Delta s_{u_1}^{103+(v_1-u_1)} = \Delta s_{6}^{112} = 1$$

Hence, $\mathcal{M}_{103}^{\vec{\mathcal{D}}} = 00011$. Now using Lemma 7, we have $y_1 = 1$ and $y_2 = 2$ and the following,

$$\Delta A_1^{103,v_1} = \Delta A_2^{103,v_2} \implies \Delta A_1^{103,15} = \Delta A_2^{103,21}$$

The above equality can also be verified from the trail given in Fig. 5 (in both the cases, the key difference is 0). Now, as $wt(\mathcal{M}_{103}^{\vec{\mathcal{D}}}) \geq 2$, thus from Lemma 8 it can be concluded that MAND is able to deliver a probabilistic advantage. *Note that, the above pattern 00011 can only be captured through MAND. BAND will* ***not*** *be able to capture such patterns.*

102	00000000000000000000000000000000	00000000000000000000
103	00000000000000000000000000000000	00000000000000000000
104	00000000000000000000000000000000	00000000000000000000
105	10000000000000000000000000000000	00000000000000000000
106	11000000000000000000000000000000	00000000000000000000
107	01100000000000000000000000000000	00000000000000000000
108	00110000000000000000000000000000	00000000000000000000
109	00011000000000000000000000000000	00000000000000000000
110	00001100000000000000000000000000	00000000000000000000
111	00000110000000000000000000000000	00000000000000000000
112	00000011000000000000000000000000	00000000000000000000
113	00000001100000000000000000000000	00000000000000000000
114	00000000110000000000000000000000	00000000000000000000
115	00000000011000000000000000000000	00000000000000000000
116	00000000001100000000000000000000	00000000000000000000
117	00000000000110000000000000000000	10000000000000000000
118	00000000000011000000000000000000	11000000000000000000
119	00000000000001100000000000000000	01100000000000000000
120	00000000000000110000000000000000	00110000000000000000

Fig. 5: Experimental demonstration of a MAND occurrence. The figure shows the last few rounds trail of the 60-round (120 iterations) KATAN48 related-key distinguisher. In the figure, left, middle and right columns refer to the iteration number, bit-differences in L_2 and L_1 register respectively. In the L_2 register, the red-colored bits denote the position 6, 13, 15 and 21 (starting from left).

In the next section, we showcase, how the advantage that MAND provides can ve leveraged in the DEEPAND modeling of NLFSR based ciphers using MILP.

5 MILP Based DEEPAND Modeling for NLFSR

For a given differential trail in NLFSR based ciphers, the probability is calculated by counting the total number of active AND gates in each round. The objective is to find the optimal trail with the minimum number of active AND gates in a fixed number of rounds. In the simple MILP model, the goal is to minimize the number of non-zero input differences to the AND gates in each round. The authors of [9] studied the impact of AND gates on the trail probability for the single AND-based NLFSR cipher TinyJAMBU. They found that subsequent AND gates may depend on each other and form what we in the current work defined as a BAND, which has a significant effect on the trail probability. To capture such a BAND, they proposed a refined MILP model for TinyJAMBU. However, as per our investigations, it has been observed that further refinement is possible and a generalized model can be devised to extend the findings to a class of NLFSR based ciphers. The refined model for capturing BANDs is described in Section 5.1. In Section 5.2, we describe how to capture MAND for multiple AND based NLFSR ciphers, which automatically includes BAND.

5.1 MILP Modeling of BAND

To model $\vec{\mathcal{C}}$ leveraging the BAND $\mathcal{B}_i^{\vec{\mathcal{C}}}$, we use γ_i to capture the correlation among two subsequent active AND gates. For each round, we compute γ_i as $\gamma_i = l_1 \ \overline{\Delta s^i_{u_1}} \ r_1$. According to Lemma 6, we have $\Delta A_1^{i,u_1} = \Delta A_1^{i,v_1}$. Thus for the pivot postion at u_1 in the consecutive rounds of the state, the following constraints will be added to the MILP model to capture the correlation in BAND.

$$\gamma_i = l_1 \ \overline{\Delta s^i_{u_1}} \ r_1, \qquad \Delta A_1^{i,u_1} - \Delta A_1^{i,v_1} \leq 1 - \gamma_i, \qquad \Delta A_1^{i,v_1} - \Delta A_1^{i,u_1} \leq 1 - \gamma_i$$

5.2 MILP Modeling of MAND

The number of valid patterns of MAND, which captures the dependency among the output differences of subsequent active AND gates, is described in the following Lemma 9.

Lemma 9. *The number of valid patterns (λ) of a MAND $\mathcal{M}_i^{\vec{\mathcal{D}}}$ of an NLFSR-based cipher $\vec{\mathcal{D}}$ with h AND gates is equal to* $\sum_{m=2}^{2h} \binom{2h}{m} = 4^h - 2h - 1$.

Proof. Consider a MAND with $wt(\mathcal{M}_i^{\vec{\mathcal{D}}}) = m$. There are $\binom{2h}{m}$ valid patterns of $\mathcal{M}_i^{\vec{\mathcal{D}}}$ which shows the dependency between m subsequent active AND gates. By Lemma 8, for a MAND, if $m \geq 2$, then we have shown a dependency between the output differences of AND gates. Therefore, the total number of valid MAND will be $\binom{2h}{2} + \binom{2h}{3} + \cdots \binom{2h}{2h} = 4^h - 2h - 1$. □

For modeling the dependency among the subsequent active AND gates, the approach is quite similar to the model given in [9]. To do so, first, a constraint is used to identify which AND gates are correlated and then pairs of AND gates are considered to model the dependency between them. So, to capture any bit difference pattern in the MAND with $m \geq 2$, we have added some extra constraints corresponding to the chained active AND gates in the simple MILP modeling. As the MAND $\mathcal{M}_i^{\vec{\mathcal{P}}}$ has λ different valid patterns, we take γ_z, $1 \leq z \leq \lambda$ to capture the correlation among $wt(\mathcal{M}_i^{\vec{\mathcal{P}}})$ number of active AND gates.

Thus for the pivot postions at u_1 in the consecutive rounds i of the state, we have λ number of γ_z and compute them in the following way.

$$\gamma_z = l_{w_1} \cdots l_{w_p} \overline{l_{w_1'}} \cdots \overline{l_{w_{p'}'}} \; \overline{\Delta s_{u_1}^i} r_{y_1} \cdots r_{y_q} \overline{r_{y_1'}} \cdots \overline{r_{y_{q'}'}}$$

$$\text{Where, } \begin{cases} l_{w_1} = \cdots = l_{w_p} = r_{y_1} = \cdots = r_{y_q} = 1 \\ l_{w_1'} = \cdots = l_{w_{p'}'} = r_{y_1'} = \cdots = r_{y_{q'}'} = 0 \\ \text{Such that } \begin{cases} \{w_1, \cdots, w_p\} \cup \{w_1', \cdots, w_{p'}'\} = \{u_1, \cdots, u_h\}, \\ \{w_1, \cdots, w_p\} \cap \{w_1', \cdots, w_{p'}'\} = \varnothing, \\ \{y_1, \cdots, y_q\} \cup \{y_1', \cdots, y_{q'}'\} = \{v_1, \cdots, v_h\}, \\ \{y_1, \cdots, y_q\} \cap \{y_1', \cdots, y_{q'}'\} = \varnothing \end{cases} \end{cases}$$

Lemma 7 implies that $\Delta A_{w_1}^{i,v_{w_1}} = \cdots = \Delta A_{w_p}^{i,v_{w_p}} = \Delta A_{y_1}^{i,v_{y_1}} = \cdots = \Delta A_{y_q}^{i,v_{y_q}}$. Therefore, for each of λ valid bit difference patterns of a MAND, the correlation is captured by the constraints given in Table 5. There constraints constitute the DEEPAND model for MILP that is used to find the better differentials for KATAN and TinyJAMBU leading to improved attacks on both the lightweight ciphers which are discussed in the subsequent sections.

Table 5: MILP Constraints Pertaining to DEEPAND

$\gamma_z = l_{w_1} \cdots l_{w_p} \overline{l_{w_1'}} \cdots \overline{l_{w_{p'}'}} \; \overline{\Delta s_{u_1}^i} r_{y_1} \cdots r_{y_q} \overline{r_{y_1'}} \cdots \overline{r_{y_{q'}'}}$	
$\left.\begin{array}{l} \Delta A_{w_t}^{i,u_{w_t}} - \Delta A_{w_x}^{i,u_{w_x}} \leq 1 - \gamma_z, \\ \Delta A_{w_x}^{i,u_{w_x}} - \Delta A_{w_t}^{i,u_{w_t}} \leq 1 - \gamma_z, \end{array}\right\}$	$1 \leq t < x \leq p$
$\left.\begin{array}{l} \Delta A_{w_t}^{i,u_{w_t}} - \Delta A_{y_x}^{i,v_{y_x}} \leq 1 - \gamma_z, \\ \Delta A_{y_x}^{i,v_{y_x}} - \Delta A_{w_t}^{i,u_{w_t}} \leq 1 - \gamma_z \end{array}\right\}$	$1 \leq t \leq p,\ 1 \leq x \leq q$

6 Attacks on TinyJAMBU

The DEEPAND model has been applied to mount attacks on variants of *keyed* permutation $\mathcal{P}_l$, $\hat{\mathcal{P}}_l$ of TinyJAMBU. We start with a brief discussion of the rele-

vant previous attacks before sharing the results obtained in this work to give a perspective on the degree of improvement.

6.1 Attacks on Keyed Permutation $\mathcal{P}_l$

In their security analysis of the mode, the designers consider $\mathcal{P}_l$ to be an ideal keyed-permutations which means under a chosen plaintext attack, $\mathcal{P}_l$ cannot be distinguished from a random permutation. This gives us a motivation to evaluate the security of $\mathcal{P}_l$ against differential cryptanalysis as a stand-alone keyed-permutation. Furthermore, based on our proposed DEEPAND model, we show that the keyed permutations $\mathcal{P}_l$ and $\hat{\mathcal{P}}_l$ do not behave as a pseudo-random permutations.

6.1.1 MILP Modeling for Finding Differential Trail As the design of TinyJAMBU is similar to the cipher described in Section 4.1, from Lemma 2 it can be concluded that the first $(128-85-1)=42$ rounds are *completely* free and the next $(85-70)=15$ rounds are *conditionally* free. For the rest number of rounds refined modeling [9] is employed. It is worth mentioning that our findings with complete and conditionally free rounds lead to improvements of the results reported in [9].

To find the differential characteristics of $\mathcal{P}_l$, in addition to the refined model, the Observation 1 and Observation 2 are employed to improve the probability. By Lemma 2, it can be concluded that the first $(128-85-1)=42$ rounds is completely free, but some of the next $(85-70)=15$ rounds are conditionally free when a particular difference pattern $(\Delta s_{70}, \Delta s_{85}) = (0,1)$ occurs in the input to the AND gate and s_{70} is completely known. This conditional free scenario is demonstrated in Table 6.

Consider the bits 70 and 85 in round number 43 to 57 of the trail given in Table 7. It is evident from the table that in round 49 and 52, $\Delta s_{70}^{49} = \Delta s_{70}^{52} = 0$ and $\Delta s_{85}^{49} = \Delta s_{85}^{52} = 1$. As s_{70}^{49} and s_{70}^{52} are known, the output difference of the corresponding AND gate is deterministic. Hence, this gives a factor of

Table 6: Part of differential trail of TinyJAMBU showing the effect of Observation 2.

#Rnd	$\Delta s_{70\cdots 85}$	Conditionally Free
42	0000000000000000	No
43	0000000000000000	No
⋮	⋮	⋮
48	0000000000000000	No
49	0000000000000001	Yes
50	0000000000000010	No
51	0000000000000100	No
52	0000000000001001	Yes
⋮	⋮	⋮
57	0000000100100000	No

2^2 advantage in the probability. Notice that, although it gives a factor of 2^2 advantage in the probability, parallely it also decreases the message space by the factor of 2^2. However, in general, in both the free and conditionally free cases, the trail probability can be increased by fixing some of the input message bit values. So, for the differential attack, we need a trade-off between the probability and the message space (the data complexity of the attack).

Table 7: Type 4 Differential Trails of $\mathcal{P}_{384}$ with Probability 2^{-14}

Input:	$\Delta S_{127\cdots0}$	0x00000000	0x88040000	0x00000248	0x02000043
	$\Delta S_{255\cdots128}$	0x00000000	0x80000000	0x00010000	0x00000012
	$\Delta S_{383\cdots256}$	0x00000000	0x80000000	0x00000000	0x00000000
Output:	$\Delta S_{511\cdots384}$	0x04080000	0x80004000	0x00010200	0x00000010

Discussion It should be noted that the use of a single AND gate in TinyJAMBU means that the dependencies between the AND gates (BAND) will remain the same. Our analysis took into account the keyed-permutation of TinyJAMBU, so these conditions will remain unaltered. A similar type of differential analysis was performed in [10] using a refined MILP model, which showed that the first 43 rounds are free when both inputs to the AND gate are known. Additionally, we have shown that even when only one input bit of the AND gate is known, the output difference of the AND gate can be deterministic (for rounds 43 to 57). This property was not captured in previous works [9,10], but we have identified it as the underlying factor behind the DEEPAND model. This same property leads to the modeling of the correlation among multiple AND gates when used in a block cipher like KATAN. If we compare our model with [9], we need to omit the initial free rounds and our model will be similar to theirs. However, if we want to take advantage of the known plaintext scenario, then our model can be better or at least as good as that of [10].

6.1.2 Cluster Differential Trail of $\mathcal{P}_{384}$ By employing the DEEPAND model in MILP, we are able to find better differential trails. A comparison of these three models for both Type-IV and Type-I differences with respect to different rounds is summarized in Table 8. For 320 rounds, our model gives a differential trail with probability 2^{-8} which is much better than previously reported results. For $\mathcal{P}_{384}$, a Type-IV differential trail with probability 2^{-14} is found. The trail is shown in Table 7. We obtained 4 differential trails with the same input and output difference as shown in Table 7 each with probability 2^{-14}, 2^{-15}, 2^{-16} and 2^{-17}. Thus the overall probability for the differential trail is $2^{-13.17}$.

Also, using the DEEPAND model, we have found a Type-III differential trail[5] of $\mathcal{P}_{384}$ with probability 2^{-71}. The input and output differences are given in

[5]We have found a 384 round Type-III differential trails with probability 2^{-71} by running our DEEPAND model. Meanwhile, we don't know why we did not get this trail

Table 8: Best Results for Type-IV and Type-I Trails of TinyJAMBU Correspond to Different MILP Models. "?" denotes that the solver has not stopped. Here each entry equals $-\log_2(\text{Trail}_{\text{Probability}})$

Number of Rounds	Simple Model [1]		Refined Model [9]		DEEPAND Model	
	Type-IV	Type-I	Type-IV	Type-I	Type-IV	Type-I
128	2	6	2	6	0	5
192	4	13	4	12	2	11
256	8	22	8	20	5	19
320	13	33	12	29	8	28
384	–	45	19	41	14	40?
480	–	–	29?	–	22	–
640	–	88	53?	–	42?	79?
1024	–	–	–	–	108?	–

Table 9 that consists of total 84 active active AND gates among which 6 gates are completely free, 0 gates are conditionally free, and 13 gates are correlated. Therefore to satisfy this Type-III trail with probability 2^{-65}, we need to fix precisely 6 bits in the input message. As a result, the message space will become reduced from 2^{128} to 2^{122}. We then evaluated its probability by finding multiple differential trails with the same input and output difference, given in Table 9. We found 50 distinct trails with probability 2^{-70} or more, whose distribution is listed in Table 10. By taking account of all these distinct trails, the overall probability to satisfy this Type-III trail will become $2^{-61.88}$

6.1.3 Differential Trail of $\mathcal{P}_{640}, \mathcal{P}_{1024}$ The MILP model developed in this work has also been applied on the keyed permutations $\mathcal{P}_{640}$ and $\mathcal{P}_{1024}$ to find the best Type-IV differential trail. For, $\mathcal{P}_{640}$, we have found Type-IV and Type-I differential trails with probabilities of 2^{-42} and 2^{-79} respectively. We also searched

Table 9: Differential Trails of the TinyJAMBU Keyed Permutation $\mathcal{P}_l$

Keyed Permutation	Differential Trail		
	Type	probability	Masks
$\mathcal{P}_{384}$	Type-III	2^{-65}	Input Difference: 0x048a2000 0x00000000 0x00000000 0x00000000 Output Difference: 0x40800441 0x00000000 0x00000000 0x00000000
$\mathcal{P}_{640}$	Type-III	2^{-93}	Input Difference: 0xc3804381 0x00000000 0x00000000 0x00000000 Output Difference: 0x00000100 0x00000000 0x00000000 0x00000000
$\mathcal{P}_{640}$	Type-IV	2^{-42}	Input Difference: 0x00000204 0x10000080 0x00412000 0x01020800 Output Difference: 0x20409200 0x88000480 0x00001020 0x00024001
$\mathcal{P}_{1024}$	Type-IV	2^{-108}	Input Difference: 0x00308080 0x00002129 0x00000808 0x00420000 Output Difference: 0x40110000 0x02040920 0x00800048 0x00000102

by using the implementation provided in [9]. One possible reason is that in both models the MILP solver did not stop to provide the best trails. In conclusion, this is not an advantage of the DEEPAND model but perhaps was not captured in [9].

Table 10: Multiple Type-III Paths and Their Probabilities of $\mathcal{P}_{384}$

Probability	2^{-65}	2^{-66}	2^{-67}	2^{-68}	2^{-69}	2^{-70}
Number of Trails	3	3	7	10	13	14

for the best Type-III trail of $\mathcal{P}_{640}$ and were able to find a trail with probability 2^{-93} (see Table 9). However, for $\mathcal{P}_{1024}$, we could only find a differential trail with probability 2^{-108}. Note that the solver is unable to find the best trails due to a higher number of rounds in both the permutations $\mathcal{P}_{640}$,$\mathcal{P}_{1024}$.

6.1.4 Related-key Differential Trail of $\mathcal{P}^{128}_{1024}$, $\mathcal{P}^{192}_{1152}$, and $\mathcal{P}^{256}_{1280}$

The designers have mentioned that if two related keys are available, then TinyJAMBU has the sliding property which can be prevented by adding the frame bits to the state. Although, for the keyed permutations $\mathcal{P}_l$ in the TinyJAMBU mode, the related-key differential attack is less practical compare to the single-key differential attack, we have applied our DEEPAND model for the keyed permutations $\mathcal{P}^{128}_{1024}$, $\mathcal{P}^{192}_{1152}$, and $\mathcal{P}^{256}_{1280}$ in the related key setting and found trails which are summarized in Table 11.

Table 11: Related-key Differential Trails of the TinyJAMBU Keyed Permutations $\mathcal{P}^{128}_{1024}$, $\mathcal{P}^{192}_{1152}$, and $\mathcal{P}^{256}_{1280}$

Keyed Permutation	Differential Trail			
	Type	probability	Masks	
$\mathcal{P}^{128}_{1024}$	Type-IV	2^{-14}	Input Difference:	0x00000000 0x00000000 0x00000004 0x00000000
			Output Difference:	0x00000000 0x00000000 0x00000004 0x00000000
			Key Difference:	0x20000000 0x00020000 0x00000000 0x00000000
$\mathcal{P}^{192}_{1152}$	Type-IV	2^{-10}	Input Difference:	0x00000000 0x00000000 0x00000000 0x20000000
			Output Difference:	0x00000000 0x00000000 0x00000000 0x20000000
			Key Difference:	0x01000000 0x00001000 0x00000000 0x20000000
				0x00000000 0x20000000
$\mathcal{P}^{256}_{1280}$	Type-IV	2^{-8}	Input Difference:	0x00000004 0x00000000 0x00000000 0x10000000
			Output Difference:	0x00000000 0x00000000 0x00000000 0x10000000
			Key Difference:	0x00800004 0x00000800 0x00000000 0x10000000
				0x00000000 0x00000000 0x00000000 0x10000000

6.2 Fixing Saha *et al.*'s Forgery Attack [9]

In this subsection, we show that the forgery attack furnished in [9] has a flaw which makes it ineffective. To be precise, the flaw originates from the lack of entropy or degrees of freedom in generating sufficient messages to create a favorable event for the forgery. We restate the attack in order to highlight flaw in the arguments furnished in [9] followed by our fix. In their work Saha *et al.* discuss

the forgery attack that can occur during the nonce setup or data processing phase. The attack involves injecting a 32-bit difference Δ_i into the i-th input block and then cancelling the state differences by injecting another 32-bit state difference Δ_{i+1} into the $(i+1)$-th input block, which maps to Type-III difference. The attack is based on the existence of a differential trail that maps the state difference $(\Delta_i||0^{96})$ to $(\Delta_{i+1}||0^{96})$ through $\mathcal{P}_l$ with probability p.

There are two types of attacks mentioned in the paper. The first one is called the "*probabilistic nonce-reuse almost universal forgery*" where the length of the associated data must be at least two blocks. The attacker repeatedly makes queries to the encryption oracle with the same nonces to observe the tag T. If the observed tag $T^{'}$ is matched with the tag T, the attacker succeeds in making a forgery. This attack breaks the 64-bit security if the differential trail $\Delta_i \rightarrow \Delta_{i+1}$ of $\mathcal{P}_l$ has a probability $p \geq 2^{-64}$. The second attack is called the "*nonce-respect almost universal forgery with reforgeability*" where the attacker can choose the first 64 bits (out of 96 bits) of the nonce $N = N_0||N_1||N_2$, and can make a forgery for any (A, M) immediately after finding N and T that satisfy the nonce-respect requirement. The attacker repeatedly makes queries to the encryption oracle with different nonces to observe the tag T. If the observed tag $T^{'}$ is equal to T, the attacker succeeds in making a forgery. The success probability of this attack is $D \times p$, where D is the number of distinct nonces examined by the attacker. Once the attacker finds a collision, they can obtain a valid tag for any (A, M) by choosing the last 32 bits of the nonce arbitrarily.

Now to satisfy the trail $\Delta_i \rightarrow \Delta_{i+1}$ for $\mathcal{P}_l$, the number of distinct state pairs (D) should be at least $\frac{1}{p}$. In another way, we can say that the expected number of state pairs to satisfy a given trail $\Delta_i \rightarrow \Delta_{i+1}$ will be $D \times p$. For the second forgery attack, by choosing different N_0, the number of distinct state pairs $(S, S^{'})$ with $S \oplus S^{'} = \Delta_i$ at the processing of the first nonce block will be $D = 2^{31}$. Note that, in this case, varying N_1 does not have any effect to increase the number of state pairs (D). In [9], the authors found a differential trail $\Delta_i \rightarrow \Delta_{i+1}$ for $\mathcal{P}_{338}$ with probability $p = 2^{-62.68}$ (by considering multiple paths). Thus, for $\mathcal{P}_{338}$, using the second forgery attack, the attacker can find a state collision after exhausting the first two nonce blocks with probability $2^{31} \times 2^{-62.68} \approx 2^{-31.68} (\ll 1)$. Therefore, for $\mathcal{P}_{338}$, the proposed attack ***cannot*** *effectively find a state collision* to break the 64-bit authentication security, i.e., the probability to make a state collision at the first two nonce processing blocks will be $2^{-31.68}$ even though the attacker can make $2^{31} \times 2^{31} (= 2^{62})$ number of Q1 and Q2 queries.

In order to carry out a forgery attack in the nonce-respect scenario, the attacker needs to perform two queries repeatedly:

Q1: The attacker makes a query to the encryption oracle with inputs $(N_0||N_1||N_2, A^*, M^*)$ in order to observe the tag T.

Q2: The attacker makes a related query to the encryption oracle with inputs $(N_0||N_1 \oplus \Delta_i||N_2 \oplus \Delta_{i+1}, A^*, M^*)$ in order to achieve a successful forgery if the observed tag $T^{'}$ is equal to T.

In this scenario, the number of chosen state pairs at the input of the second nonce block for $\mathcal{P}_l$ would be $2^{32} \times 2^{31} = 2^{63}$. This means that if a given trail

$\Delta_i \rightarrow \Delta i+1$ has a probability $p \geq 2^{-63}$, then after making queries of Q1 and Q2 for all nonces N_0, N_1, it is expected that there will be at least one state collision at the third nonce block position, which will immediately lead to the forgery. Additionally, if $N_0||N_1||N_2$ and $N_0||N_1 \oplus \Delta_i||N_2 \oplus \Delta_{i+1}$ are two 96-bit nonces that result in a state collision, then the attacker can choose the last 32 bits of nonce $\tilde{N_2}$ ($\neq N_2, N_2 \oplus \Delta_{i+1}$) arbitrarily to obtain a tag T for $(N_0||N_1||\tilde{N_2}, A^*, M^*)$ through an encryption query. Then T will also be valid for $(N_0||N_1 \oplus \Delta_i||\tilde{N_2} \oplus \Delta_{i+1}, A^*, M^*)$ implying a forgery.

According to our analysis using the DEEPAND model for $\mathcal{P}_{384}$, we discovered a differential trail with a probability[6] of 2^{-65} when an attacker has the ability to manipulate 6 bits in the input message during encryption. After taking into account multiple paths for this trail, the probability increases to $2^{-61.88}$. However, in this forgery attack, the attacker has no control over the initial bits in the message and cannot freely bypass some initial AND gates. Therefore, by not considering the manipulation of the message bits at the initial 57 rounds of Tiny-JAMBU state, the overall probability decreases to $2^{-67.88}$, which is higher than the original estimations made by Saha *et al.* and the designers. Our DEEPAND model analysis for $\mathcal{P}_{384}$ suggests that the security margin against differential cryptanalysis is less than 4 bits.

7 Attacks on KATAN

In this section, to find the best differential trails of any rounds in the KATAN ciphers, we will show that how the DEEPAND model efficiently captures the correlated ANDs and significantly increase their trail probability. First, we will show that the differential characteristics using our DEEPAND model for some initial rounds of KATAN give a much better probability than the designer's claims in [5]. Then, we show that the related key boomerang attack on KATAN in [6] can also be improved by employing this new model.

7.1 Improved Differential Cryptanalysis of KATAN

7.1.1 MILP Modeling of Free Rounds. In KATAN, there are three AND gates where the tuples $(y_3, y_4), (y_5, y_6)$, and (x_3, x_4) represent the input bit-positions to the AND gates A_1, A_2, and A_3 respectively. Then by Lemma 4, the differential output of the gates A_1, A_2 and A_3 in the forward differential trail are deterministic for the first (y_4+1), (y_6+1), and (x_4+1) rounds respectively. Also, they are conditionally free from the round number (y_4+2) to (y_3+1), (y_6+2) to (y_5+1), and (x_4+2) to (x_3+1) respectively.

Similarly, by Lemma 5, it can be concluded that in the backward differential trail, the output differences of the gates A_1, A_2 and A_3 are deterministic for the first $(n-y_3-1)$ rounds, first $(n-y_5-1)$ rounds and first $(n-x_3-1)$ rounds

[6] The attack scenario does not allow for the attacker to control the input message bits in the encryption process, thus we have taken into account the cost of fixing the message bits by multiplying the overall probability by 2^{-6}.

respectively and conditionally free from round number $(n - y_3)$ to $(n - y_4 - 1)$, $(n - y_5)$ to $(n - y_6 - 1)$ and $(n - x_3)$ to $(n - x_4 - 1)$ respectively. Here n denotes the state-size of KATAN.

7.1.2 Modeling the Dependency Between AND Gates For KATAN, there is only one AND gate in the L_1 register. In this case, a BAND can happen during intermediate rounds. To capture all the BANDs in rounds, we have to track the BAND for each round and then we add the respective constraints according to the MILP model discussed in Section 5.2.

In L_2 register, there are two AND gates and the dependency between two different AND gates is not captured in the refined model. Consider a bit s_3^i in register L_2. For KATAN32, the MAND of the pivotal difference $\Delta s_3^i = 0$ is

$$\mathcal{M}_i^{\text{KATAN32}} = \Delta s_{10}^{i+9} \Delta s_3^{i+5} \boxed{\Delta s_3^i} \Delta s_8^i \Delta s_{12}^{i+7} = \Delta s_1^i \Delta s_{-2}^i \boxed{\Delta s_3^i} \Delta s_8^i \Delta s_5^i$$

Now by Lemma 9 there are $\binom{4}{4} + \binom{4}{3} + \binom{4}{2} = 11$ patterns for which output differential of several AND computations are inter-related. The MAND and its corresponding differential bit patterns with refined probabilities are shown in Table 12.

Table 12: MAND of Δs_3^i and the corresponding differential value of related bits.

MAND	Δs_8^i	Δs_5^i	Δs_3^i	Δs_1^i	Δs_{-2}^i	Naive Prob.	Improved Prob.
11 0 11	1	1	0	1	1	2^{-4}	2^{-1}
11 0 10	1	1	0	1	0	2^{-3}	2^{-1}
11 0 01	1	1	0	0	1	2^{-3}	2^{-1}
10 0 11	1	0	0	1	1	2^{-3}	2^{-1}
01 0 11	0	1	0	1	1	2^{-3}	2^{-1}
11 0 00	1	1	0	0	0	2^{-2}	2^{-1}
10 0 10	1	0	0	1	0	2^{-2}	2^{-1}
01 0 10	0	1	0	1	0	2^{-2}	2^{-1}
10 0 01	1	0	0	0	1	2^{-2}	2^{-1}
01 0 01	0	1	0	0	1	2^{-2}	2^{-1}
00 0 11	0	0	0	1	1	2^{-2}	2^{-1}

7.1.3 DEEPAND Based New Differential Trails for KATAN In [5], the designers have claimed that for 42-round KATAN32, the best differential characteristic has probability 2^{-11}. However, for the initial 42 rounds, the DEEPAND MILP model is able to find two *identical* differential trails with probability 2^{-7}.

For 43-round KATAN48 and 37-round KATAN64, the best differential trail, as claimed by the designers, can be found with probability 2^{-18} and 2^{-20} respec-

tively whereas for both variants our model finds differential trails with probability 2^{-14}.

Table 13: Differential Properties of KATAN Variants. #R ← number of rounds, fr ← number of ANDs to be freely passed, C_{fr} ← number of conditionally free ANDs, C_A → number of correlated ANDs, t ← number of required ANDs where probability should be paid, p_α and p → refer to probabilities with and without bit-fixing.

Cipher	#R	Active Gates				Difference		Probability	
		t	fr	C_{fr}	C_A	Input	Output	p_α	p
KATAN32	42	7	4	1	0	0x08020040	0x00200420	2^{-7}	2^{-11}
	74	29	2	0	0	0x0000c010	0x40880101	2^{-29}	2^{-31}
	81†	29	5	0	4	0x10802004	0x00000800	2^{-29}	2^{-34}
KATAN48	43	14	10	0	0	0x000008442c10	0x040000000229	2^{-14}	2^{-24}
KATAN64	37	17	3	2	0	0x4000002001000800	0x0444200000001000	2^{-17}	2^{-20}

†Note that, this trail has the probability 2^{-34} if we do not consider any message bit fixing. So, this 81-round trail can not be verified because the message space for KATAN32 is 2^{32}.

7.2 Related Key Differential Attack

In the related-key setting, the DEEPAND model was applied to the KATAN32 cipher, and the best trail probabilities for various rounds are summarized in Table 14. This model outperforms previous simple and refined models in capturing multiple correlated ANDs. These correlated ANDs not only increase the trail probability but also aid in finding longer differential trails. As a result, this model can be used to identify better related-key differential trails for the KATAN48 and KATAN64 ciphers compared to the simple and refined models.

7.2.1 Improving Isobe *et al.*'s Related Key Boomerang Attack [6]

The related-key boomerang attack is a combination of the boomerang attack and the related-key differential attack. Such attacks are useful to build a distinguishers when it consists of two shorter differential trails with high probabilities.

In [6] for KATAN32 ($= E_1 \circ E_0$), the authors devise a 140-round boomerang distinguisher, where both E_0 and E_1 have 70-rounds. Based on their efficient differential characteristics search for both E_0 and E_1, the authors provided maximum probability differential characteristics of each set in [6, Table 5,6]. In the construction of the boomerang distinguisher, the authors choose a differential characteristic of E_0 corresponding to the set 8 [6, Table 4] with probability 2^{-9}

[7]Note that, for larger rounds, the DEEPAND model could not find the best trails due to too many constraints in the model.

Table 14: Related-key Differential Properties of KATAN. #R ← number of rounds, fr ← number of ANDs to be freely passed, C_{fr} ← number of conditionally free ANDs, C_A → number of correlated ANDs, t ← number of required ANDs where probability should be paid, p_α and p → refer to probabilities with and without bit-fixing.

Cipher	#R	Active Gates				Difference[7]			Probability	
		t	fr	C_{fr}	C_A	Input	Output	Key	p_α	p
KATAN32	60	3	0	0	0	$0x00004000$	$0x00b80084$	$\Delta k[9,39,50,54,64]=1$	2^{-3}	2^{-3}
		7	0	0	0	$0x00042000$	$0x00880801$	$\Delta k[1,11,53,64,68,78]=1$	2^{-7}	2^{-7}
	70	6	1	0	0	$0x80031000$	$0x01200400$	$\Delta k[0,3,5,13,55,70,72]=1$	2^{-6}	2^{-7}
		4	3	0	0	$0xa4020010$	$0x00b80084$	$\Delta k[3,4,7,10,17,29,59,70,74]=1$	2^{-4}	2^{-7}
	84	16	0	1	1	$0xa0048000$	$0x01180263$	$\Delta k[1,4,23,31,42,61]=1$	2^{-16}	2^{-17}
KATAN48	50	0	7	0	0	$0x000000301800$	$0x000180000000$	$\Delta k[17]=1$	2^{0}	2^{-7}
	59	6	3	0	0	$0x000003018000$	$0x000000001460$	$\Delta k[13]=1$	2^{-6}	2^{-9}
		6	2	3	0	$0x820031400000$	$0x000060003000$	$\Delta k[5,24]=1$	2^{-6}	2^{-11}
	60	7	2	3	1	$0x820031400000$	$0x00018000c000$	$\Delta k[5,24]=1$	2^{-7}	2^{-12}
		6	14	0	0	$0xdb0000643018$	$0x180000000005$	$\Delta k[6,25]=1$	2^{-6}	2^{-20}
KATAN64	56	11	4	0	0	$0x0000001c00e00000$	$0x000020000001cce0$	$\Delta k[11]=1$	2^{-11}	2^{-15}
	57	13	3	0	1	$0x0000004801c00000$	$0x00000380001c0e00$	$\Delta k[1,7,20,26]=1$	2^{-13}	2^{-16}

and of E_1 for the set 10 [6, Table 4] with probability 2^{-8}. Thus the the probability to form a simple boomerang will be $(2^{-9})^2 \times (2^{-8})^2 = 2^{-34}$. Whereas for KATAN32, the attacker only has 2^{31} number of input message pairs with a fixed difference. To reduce the data complexity for this boomerang attack, the authors have considered multiple trails with the same input and output difference. As a result, the overall probability for the trails in E_0 and E_1 improves to $2^{-7.1}$ and $2^{-6.5}$ respectively. Therefore by combining these two differential characteristics, the overall probability of the above 140-round related-key boomerang distinguisher is increased to $(2^{-7.1})^2 \times (2^{-6.5})^2 = 2^{-27.2}$.

Table 15: Sets of key difference considered in [6].

Set	0	1	2	3	4	5	6	7	8	9	10
Key Difference	0,19	1,20	2,21	3,22	4,23	5,24	6,25	7,26	8,27	9,28	10,29
Plaintext Difference	$L_2[9]$ $L_1[12]$	$L_2[18]$ $L_1[2,7,12]$	$L_2[8]$ $L_1[11]$	$L_2[17]$ $L_1[1,6,11]$	$L_2[7,18]$ $L_1[10]$	$L_2[16]$ $L_1[0,5,10]$	$L_2[6,17]$ $L_1[9]$	$L_2[15,18]$ $L_1[4,9]$	$L_2[5,16]$ $L_1[8]$	$L_2[14,17]$ $L_1[3,8]$	$L_2[4,15]$ $L_1[7,12]$

Using the DEEPAND model, we have verified all the trails corresponding to the differential characteristics of each set in [6, Table 5]. For set 0 and set 10, we have respectively identified three and one correlated AND gates in the trails

Table 16: Verified Related-key Boomerang Distinguisher of KATAN32 . In hexadecimal notation, the most significant bit (MSB) is placed on the right side and the least significant bit (LSB) is located on the left side.

KATAN32					
No.	Prob.	Input Diff	Output Diff	Key Difference	
		Upper	Lower	Upper	Lower
1.	2^{-22}	$0x00026000$	$0x48008b00$	$0xa0800000000002001504$	$0x52c0a267036154fc4c36$

with probabilities 2^{-12} and 2^{-10} (without considering free and conditionally free AND gates). Whereas, according to their search strategy, the trail probabilities for set 0 and set 10 are 2^{-15} and 2^{-12}. For other sets, the DEEPAND model did not find any extra advantage in the trails. Moreover, if we do not consider the predefined sets in Table 15, the DEEPAND model is able to find much better 70-round trails of probability 2^{-7} ($> 2^{-9}$). So, by choosing two trails of probabilities 2^{-7}, 2^{-7} for both E_0, E_1, we can form a boomerang distinguisher with probability $(2^{-7})^2 \times (2^{-7})^2 = 2^{-28}$. Also, in the similar fashion, we can further reduce the data complexity of this 140-round boomerang attack by choosing the multiple differentials correspond to the same input/output difference. For the first boomerang in Table 16, the input and key difference of E_0 is represented by 4 trails of probability 2^{-7}, 8 trails of probability 2^{-8}, 16 trails of probability 2^{-9}, and 32 trails of probability 2^{-10}. Similarly, the output and key difference of E_1 is represented by 4 trails of probability 2^{-7}, 8 trails of probability 2^{-8}, and 32 trails of probability 2^{-9}. The overall probabilities of E_0 and E_1 are approximately $2^{-5.52}$ and $2^{-5.5}$, respectively. The overall probability of the boomerang distinguisher can be calculated as $(2^{-5.52})^2 \times (2^{-5.5})^2 = 2^{-22.04}$ which is greater than $2^{-27.2}$. Note that for the distinguishers given in Table 16, we have not considered any message-bit fixing in order to take advantage of the cluster of trails.

8 Conclusion

In this work, we have developed DEEPAND, a new generalized MILP model to capture the first-order correlation in single/multiple AND-based (NLFSR) ciphers. The model is developed primarily on the basis of three Observations 1, 2, 3 and introduces the notion of conditionally free rounds. In this model, it is shown that there can be dependencies among multiple AND gates in NLFSR-based ciphers. To capture the dependencies in a proper way, BAND has been introduced. In addition, it is also shown that if one of the inputs of AND gate is known, then for certain values of input differences of the AND gate, the output difference is deterministic. Using the DEEPAND model, we have primarily investigated the differential properties of the TinyJAMBU's keyed permutations. For the full-round of $\mathcal{P}_{1024}$, we found a differential trail (Type-IV) with probability 2^{-108} highlighting its non-ideal nature. For $\mathcal{P}_{640}$, the figure is 2^{-42}. For KATAN, we report the

best differential trail (verified[7]) for 42-rounds with a practical probability of 2^{-7} breaking the designer's claim. We have also bettered the related-key boomerang attack by Isobe *et al.* using DEEPAND. Finally, the designer's differential trail for 43-round KATAN48 and 37-round KATAN64 is also improved showing the widespread applicability of the new model DEEPAND. Finally, DEEPAND model developed in this work appears like an effective tool to probe into the correlations that develop during the differential propagation and warrants further investigation.

References

1. Hongjun Wu and Tao Huang: TinyJAMBU: A Family of Lightweight Authenticated Encryption Algorithms (Version 2), nIST LWC Finalist, 2021
2. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) s-boxes to optimize probability of differential characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017)
3. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. J. Cryptol. **4**(1), 3–72 (1991)
4. Boura, C., Coggia, D.: Efficient MILP modelings for sboxes and linear layers of SPN ciphers. IACR Trans. Symmetric Cryptol. **2020**(3), 327–361 (2020)
5. De Cannière, C., Dunkelman, O., Knežević, M.: Katan and ktantan — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2009. pp. 272–288. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
6. Isobe, T., Sasaki, Y., Chen, J.: Related-Key Boomerang Attacks on KATAN32/48/64. In: Boyd, C., Simpson, L. (eds.) Information Security and Privacy. pp. 268–285. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Information Security and Cryptology, Inscrypt 2011. Lecture Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2011)
8. National Institue of Standards and Technology: Lightweight Cryptography. Tech. rep. (Aug 27 2018)
9. Saha, D., Sasaki, Y., Shi, D., Sibleyras, F., Sun, S., Zhang, Y.: On the Security Margin of TinyJAMBU with Refined Differential and Linear Cryptanalysis. IACR Transactions on Symmetric Cryptology **2020**(3), 152–174 (Sep 2020)
10. Sibleyras, F., Sasaki, Y., Todo, Y., Hosoyamada, A., Yasuda, K.: Birthday-bound slide attacks on tinyjambu's keyed-permutations for all key sizes. In: Cheng, C., Akiyama, M. (eds.) Advances in Information and Computer Security - 17th International Workshop on Security, IWSEC 2022, Tokyo, Japan, August 31 - September 2, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13504, pp. 107–127. Springer (2022). https://doi.org/10.1007/978-3-031-15255-9_6, https://doi.org/10.1007/978-3-031-15255-9_6
11. Sun, S., Hu, L., Song, L., Xie, Y., Wang, P.: Automatic security evaluation of block ciphers with s-bp structures against related-key differential attacks. In: Lin, D., Xu, S., Yung, M. (eds.) Information Security and Cryptology Inscrypt 2013.

[7] All practically verifiable claims have been experimentally cross-checked and the results have been shared as supplementary material along with source-codes.

12. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: 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, Paper 2014/747 (2014)
13. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In: Advances in Cryptology - ASIACRYPT 2014