IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2022, No. 2, pp. 161–200. DOI:10.46586/tosc.v2022.i2.161-200

Revisiting the Extension of Matsui's Algorithm 1 to Linear Hulls: Application to TinyJAMBU

Muzhou Li[1,2], Nicky Mouha[3], Ling Sun[1,2] and Meiqin Wang[1,2,4](✉)

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[2] School of Cyber Science and Technology, Shandong University, Qingdao, China
[3] Strativia, Largo, MD, USA
[4] Quan Cheng Shandong Laboratory, Jinan, China
muzhouli@mail.sdu.edu.cn, nicky@mouha.be, {lingsun,mqwang}@sdu.edu.cn

Abstract. At EUROCRYPT '93, Matsui introduced linear cryptanalysis. Both Matsui's Algorithm 1 and 2 use a linear approximation involving certain state bits. Algorithm 2 requires partial encryptions or decryptions to obtain these state bits after guessing extra key bits. For ciphers where only part of the state can be obtained, like some stream ciphers and authenticated encryption schemes, Algorithm 2 will not work efficiently since it is hard to implement partial encryptions or decryptions. In this case, Algorithm 1 is a good choice since it only involves these state bits, and one bit of key information can be recovered using a single linear approximation trail. However, when there are several strong trails containing the same state bits, known as the linear hull effect, recovering key bits with Algorithm 1 is infeasible. To overcome this, Röck and Nyberg extended Matsui's Algorithm 1 to linear hulls. However, Röck and Nyberg found that their theoretical estimates are quite pessimistic for low success probabilities and too optimistic for high success probabilities. To deal with this, we construct new statistical models where the theoretical success probabilities are in a good accordance with experimental ones, so that we provide the first accurate analysis of the extension of Matsui's Algorithm 1 to linear hulls. To illustrate the usefulness of our new models, we apply them to one of the ten finalists of the NIST Lightweight Cryptography (LWC) Standardization project: TinyJAMBU. We provide the first cryptanalysis under the nonce-respecting setting on the full TinyJAMBU v1 and the round-reduced TinyJAMBU v2, where partial key bits are recovered. Our results do not violate the security claims made by the designers.

Keywords: Matsui's Algorithm 1 · Linear Hull · TinyJAMBU

1 Introduction

The linear cryptanalysis of block ciphers, originally proposed by Matsui [Mat93], uses a strong linear approximation $u \cdot x \oplus w \cdot \mathcal{E}_K(x) = v \cdot K$ between certain bits of the plaintext x, the ciphertext $\mathcal{E}_K(x)$, and the key K, where $a \cdot b = \oplus_{j=0}^{n-1} a_j b_j$ is the inner product of the two n-bit values a and b. Here, u and w are called the input and the output masks, respectively.

To evaluate how strong the approximation is, the correlation $c = 2p - 1$ was introduced, where p is the probability that the approximation holds. We say that an approximation is strong if the absolute value of the correlation is large. In order to find such strong approximations, Matsui aimed to find a linear trail by chaining approximations from round to round over the cipher and estimate the total correlation using the Piling-Up Lemma.

With such approximations, and given a sufficient amount of data, key bits can be recovered by comparing the correlation of $u \cdot x \oplus w \cdot \mathcal{E}_K(x)$ evaluated under known

Received: 2022-03-01 Accepted: 2022-05-01 Published: 2022-06-10

plaintext-ciphertext pairs $(x, \mathcal{E}_K(x))$ with the value of the correlation obtained by Matsui's Algorithms 1 and 2. However, attacks based on this approach make many assumptions that, while successful in the case of the Data Encryption Standard (DES), fail to hold for other ciphers due to strong linear hull effects.

Daemen *et al.* [DGV94] found that there may exist several trails that share the same input and output masks and give non-negligible contributions to the total correlation of $u \cdot x \oplus w \cdot \mathcal{E}_K(x)$. The set of all trails that share the same u and w was called the linear hull by Nyberg [Nyb94]. Hence, Matsui's algorithms will not work as expected in this case since correlation of $u \cdot x \oplus w \cdot \mathcal{E}_K(x)$ is determined by all trails in the hull, rather than by a single trail. Moreover, the correlation of a linear hull is related to the value of the key. For instance, the sign of the correlation of a linear trail for key-alternating ciphers varies with the value of the key and thus leads to different values of the correlation of this hull.

Several approaches have been proposed to deal with the key dependency of correlations of these trails. Nyberg [Nyb94] showed that the squared total correlation of the hull is equal to the average of the squared total correlation over the keys. However, there can be keys that give correlations with negligible magnitude so that the linear distinguisher is not effective, as pointed out by Murphy [Mur12]. For a fixed key, the correlation of the (composition of the) linear hull can be estimated by evaluating the correlations of all strong trails that belong to this hull under the same key. Cho [Cho10] showed how this can lead to improved linear attacks on PRESENT. Besides these approaches, there has been some interest in deducing key information from the value of the observed correlations [AR16, CS11, HN11, NH07, RN13].

To obtain more key information, attacks often follow the approach of Matsui's Algorithm 2 by guessing key bits involved outside the hull. However, for certain stream ciphers or authenticated encryption schemes where only partial input or output bits of state can be obtained, Matsui's Algorithm 2 will not work efficiently since it is hard to proceed with partial encryptions or decryptions. In this case, Matsui's Algorithm 1 can be used since it only uses these partial input and output bits of the state.

For ciphers with a strong linear hull effect, key recovery attacks based on Matsui's Algorithm 1 are complicated. To deal with this problem, Röck and Nyberg [RN13] proposed the linear hull version of Matsui's Algorithm 1 which can recover more information about the key. It uses the fact that the correlation of a linear hull can be the same for different keys in key-alternating ciphers. Then, the whole key space can be divided into disjoint key classes according to the corresponding correlations. If the key space can be divided in such a way, we may identify in which key class the right one lies according to the observed correlation after collecting a sufficient amount of data.

Röck and Nyberg [RN13] introduced a decision function that takes the observed correlation as input and outputs the guessed key class based on the maximum likelihood estimate (MLE). Note that there can be cases where the decision function takes a wrong class as output. As shown in [RN13], the error probability for such cases is related not only to the data complexities but also the key class that the right one lies in. In other words, when mounting attacks, this error probability will vary with the value of the right key. To theoretically decide how much data should be collected in their attack, Röck and Nyberg [RN13] adopted the approach of evaluating the average error probabilities over all keys as the total error probability of the attack. When deducing the relation between data complexities and total error probabilities, they equally treated the error probability for each wrong decision, and then computed the amount of data gathered using their statistical model in each case. The data complexity is then evaluated as the upper bound of these data complexities computed for each wrong decision.

To recover more key information, they also proposed statistical models under the basic and multiple related-key settings, where the differences of the correlations under a pair of related keys and under multiple related keys are considered, respectively. However,

Table 1: Comparison of Statistical Models involved in this paper. As our experiments show in Appendix F, our MLE-based model is slightly more precise but much slower to compute, compared to the threshold-based model.

Statistical Model	Röck and Nyberg	**This Paper** **Threshold-Based**	 **MLE-Based**
Methodology	asymptotic estimate of upper bound of data complexity	**CDF of accurately approximated distribution of data complexity**	
Decision Function	MLE	**threshold**	**MLE**
Absolute Error	93.75 % (Fig. 4)	**2.19 % (Fig. 3)**	**1.9 % (Fig. 5)**
Reference	[RN13]	**Sect. 3.1**	**Sect. 3.2**

Röck and Nyberg observed in their experimental verification of these three statistical models on PRESENT [BKL+07] that the relation between the data complexities and the error probabilities is not accurately described. To be more specific, the data complexities predicted by the theoretical models are quite pessimistic for low success probabilities, and too optimistic for high success probabilities. We found that the reason for such an inaccuracy comes from their methodology of deducing the relation, where the error probability is evaluated using an asymptotic estimate of the upper bound of the data complexities.

We will construct two kinds of new statistical models that accurately describe the relation between the data complexities and the error probabilities. The accuracy benefits from a new methodology to deduce the error probability using the cumulative distribution function (CDF) of the accurately approximated distribution of the statistic related to the data complexities. With this new methodology, two different kinds of decision functions are adopted. The first one is based on the *maximum likelihood estimate* (MLE) that was also exploited by Röck and Nyberg [RN13]; the second one is based on *threshold* values, which is slightly less precise but much easier to handle. A detailed comparison of these two statistical models along with the one proposed by Röck and Nyberg [RN13] is depicted in Table 1. The contributions of this paper are as follows:

New Statistical Models, Methodology, and Decision Function. In Sect. 3, we propose several new statistical models following the new methodology for all three attack settings considered in [RN13], which Röck and Nyberg refer to as the direct attack, the basic related-key, and the multiple related-key settings. Moreover, we build two different models under each setting according to whether the data collected are distinct or not. In all these attack settings, we are given all correlations for this linear hull and the corresponding key classes. Then, we have to decide in which key class the right one lies by comparing the statistical value with the theoretical correlation values. There are two kinds of decision functions we adopted: the one based on the maximum likelihood estimate (MLE) and the one based on threshold values. Since the MLE-based decision function has already been introduced by Röck and Nyberg [RN13], we omit the description of this kind of decision function here, and focus on the threshold-based one.

Take the statistical model for the direct attack setting where the threshold-based decision function is exploited as an example. Assume that there are q possible correlations, and let $C(K) = \{c_0, c_1, \dots, c_{q-1}\}$ denote the set containing them where $c_i < c_{i+1}$ for all $0 \le i \le q-2$. Let $\mathcal{K}(c_i)$ be the set that consists of all keys under which the correlation of the linear hull is c_i. Given a sufficient amount of data, it is likely that the observed correlation $\widehat{c}$ is close to the correlation evaluated under the right key. The statistical behavior of $\widehat{c}$ can be approximated by a normal distribution and there are q normal distributions with different expectations. To deal with the decision problem related to multiple distributions, we adopt the threshold-based decision function where the average value of every two adjacent correlations is taken as the threshold. In other words, $\mathcal{K}(c_i)$ is regarded as the right key class if $\widehat{c}$ fulfills $(c_{i-1} + c_i)/2 < \widehat{c} \leq (c_i + c_{i+1})/2$. As for the case when $i = 0$ and $q - 1$, we only have to compare $\widehat{c}$ with $(c_0 + c_1)/2$ and $(c_{q-2} + c_{q-1})/2$, respectively.

Such a decision strategy is inspired by several previous works [BLNW12, BW12, BBR+13, WCC+16] where a decision between two distributions has to be made. The probability of making a wrong decision can then be accurately evaluated with the cumulative distribution function (CDF) of the distribution of the statistic. To show the impact of this new methodology, we also applied it to the MLE-based decision function adopted by Röck and Nyberg [RN13] in Sect. 3. A detailed comparison of these two different kinds of statistical models is shown in Appendix F.

Our experiments on the 256-round permutation of TinyJAMBU confirm the error probabilities predicted theoretically, and show that the statistical models introduced by Röck and Nyberg [RN13] are far from accurate. More specifically, we find that the maximum absolute value of the theoretical probability minus the experimental probability of our models is 2.19 % (threshold-based) or 1.9 % (MLE-based), compared to 93.45 % for their models. We refer to Figs. 3, 4, and 12 for detailed comparisons.

Cryptanalysis of TinyJAMBU. TinyJAMBU is a family of Authenticated Encryption with Associated Data (AEAD) algorithms. In March 2021, the updated version TinyJAMBU v2 was selected as one of the ten finalists of the NIST LWC Standardization project [Nat21].

In [SSS+20], Saha *et al.* introduced the first third-party cryptanalysis on round-reduced TinyJAMBU v1 in the nonce-misuse setting. Later in [TSY+21], Teng *et al.* gave the first partial key recovery attacks on round-reduced TinyJAMBU v1 in the nonce-respecting setting, however, their attacks can only be applied on the cipher with the 128-bit key.

In Sect. 4, we provide partial key recovery attacks in the nonce-respecting setting which are suitable for all key lengths (*i.e.,* 128, 192, and 256 bits). These attacks are on the full TinyJAMBU v1 and the round-reduced TinyJAMBU v2 by respectively using 384-round and 387-round linear hulls in the tag generation phase with our proposed statistical models. Note that they are the first cryptanalysis results in the nonce-respecting setting on the full TinyJAMBU v1 and the round-reduced TinyJAMBU v2. A comparison between our attacks and the above two works is given in Table 2. The security claims made by the designers are not violated by our results.

To allow our results to be reproduced, all source code along with detailed instructions for all experiments in this paper is available at: https://github.com/MuzhouLi/attack_tinyjambu_code.

2 Preliminaries

2.1 Linear Trails and Linear Hulls of Key-Alternating Ciphers

Denote $\mathbb{F}_2^n$ as the space of n-dimensional binary vectors over $\mathbb{F}_2 = \{0, 1\}$. Then we can denote $\mathcal{E}_K(x)$ as the block cipher encryption of the plaintext $x \in \mathbb{F}_2^n$ under the κ-bit master key $K \in \mathbb{F}_2^\kappa$.

In this paper, we only consider key-alternating iterative ciphers. The concept of key-alternating ciphers was proposed by Daemen and Rijmen [DR02]. It forms a special but

Table 2: Summary of attacks on TinyJAMBU.

	TinyJAMBU v1			TinyJAMBU v2
Attack Phase	Nonce Setup, AD Processing	Initialization & Encryption	**Tag Generation**	**Tag Generation**
Attacked/Total Rounds	338/384	2604/3200	**384/384**	**387/640**
Nonce-Respecting	✗	✓	✓	✓
Data Complexity	$2^{62.68}$	2^{14}	$\geq \mathbf{2^{96.8}}$	$\geq \mathbf{2^{96.8}}$
Success Probability	$\approx 63\,\%$	N/A	$\geq$ **82 %**	$\geq$ **82 %**
Attack Type	Forgery	Partial Key Rec. (1-bit key)	**Partial Key Rec. ($\geq$ 7-bit key)**	**Partial Key Rec. ($\geq$ 7-bit key)**
Attack Method	Differential	Cube	**Linear Hull**	**Linear Hull**
Supported Key Length	128, 192, 256	128	**128, 192, 256**	**128, 192, 256**
Reference	[SSS+20]	[TSY+21]	**Sect. 4**	**Sect. 4**

important subset of modern block ciphers. Almost all Substitution-Permutation Networks (SPNs) and some Feistel ciphers are key-alternating.

Let k_i represent the n-bit round key in round i of an iterative block cipher with $1 \leq i \leq r$. Then for a key-alternating cipher $\mathcal{E}_K(x)$, k_i is XORed with the output state of the i-th round function f_i. Additionally, the initial round key k_0 is XORed with the plaintext before the first round. All round keys k_i with $0 \leq i \leq r$ are generated from K by means of the key schedule.

Assuming that there is a linear trail θ of an r-round key-alternating cipher, the input mask of round i is θ_{i-1} and the output mask is θ_i with $1 \leq i \leq r$. The inner product of binary vectors is defined as $u \cdot x = \bigoplus_{j=0}^{n-1} u_j x_j$ where x_0 is the rightmost bit of x. Then the correlation of the i-th round can be defined as

$$C_{\theta_{i-1},\theta_i} = 2\Pr[\theta_{i-1} \cdot x \oplus \theta_i \cdot f_i(x) = 0] - 1$$

where $f_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is the i-th round function. The correlation of the linear trail θ under K for a key-alternating cipher [DGV94] is

$$C_\theta(K) = (-1)^{\theta_0 \cdot k_0} \prod_{j=1}^{r} (-1)^{\theta_i \cdot k_i} C_{\theta_{i-1},\theta_i} = (-1)^{\theta_0 \cdot k_0 \oplus \cdots \oplus \theta_r \cdot k_r} \prod_{j=1}^{r} C_{\theta_{i-1},\theta_i}.$$

Note that only the sign of $C_\theta(K)$ is affected by K. The statistical models that we will introduce later, as well as those of Röck and Nyberg [RN13], are actually benefiting from this property. In fact, this property still holds for the variant of the iterated Even-Mansour structure where the round key is only XORed with part of the state. A simple explanation of this property is: if one takes the part that does not XOR any key in each round as XORing all 0 key bits, such ciphers can be regarded as key-alternating ciphers. Another formal proof of this property for such ciphers is provided in Appendix A. For simplicity, we will still refer to them as key-alternating ciphers when introducing the statistical models.

A linear hull (u, w), as proposed by Nyberg [Nyb94], consists of all linear trails satisfying $u = \theta_0$ and $w = \theta_r$. Its correlation can be computed if we can know all correlations of the linear trails comprising this linear hull with the condition that they are estimated under the same key K. In other words, the correlation of the linear hull (u, w) under K is

$$C(K) = \sum_{\theta:\theta_0=u,\theta_r=w} C_\theta(K) = \sum_{\theta:\theta_0=u,\theta_r=w} (-1)^{\theta_0 \cdot k_0 \oplus \cdots \oplus \theta_r \cdot k_r} \prod_{j=1}^{r} C_{\theta_{i-1},\theta_i}.$$

In the original version of Matsui's Algorithm 1 [Mat93], one bit of information of the key can be obtained using the sign of the observed correlation $C(K)$. However, this will succeed only when there is a single dominant trail in this linear hull. In other words, other trails in this hull are assumed to have negligible contributions. Known examples of ciphers admitting this condition are DES and Serpent [BAK98], in contrast to for example PRESENT [BKL⁺07] and TinyJAMBU [WH19, WH21] which split all correlations into a large number of trail correlations without a single dominant trail. For ciphers that have multiple dominant trails, Matsui's Algorithm 1 is not suitable anymore. In order to recover key bits for these ciphers, Röck and Nyberg [RN13] generalized Matsui's Algorithm 1 to consider the linear hull effect.

2.2 Extension of Matsui's Algorithm 1 to Linear Hull

Recalling the definition of $C(K)$, we note that its value is fully determined by the master key K since $C_{\theta_{i-1},\theta_i}$ are independent of the key bits. Hence, different values of $C(K)$ will separate the whole key space into disjoint classes. With these key classes, we can recover some information of the key by identifying in which class the key lies according to the observed $C(K)$, given a sufficiently large amount of data. The extension of Matsui's Algorithm 1 proposed by Röck and Nyberg [RN13] is based on this observation.

However, there may be a lot of trails comprising a linear hull, so that considering all of them to compute $C(K)$ may not be practical. Therefore, Röck and Nyberg [RN13] only consider trails whose absolute correlations are above a certain threshold τ when computing $C(K)$ and rely on Assumption 1. This assumption may not hold for all ciphers. However, for the results in this paper on TinyJAMBU, we find all trails in the linear hull, therefore this assumption is not necessary.

Assumption 1. *(Röck and Nyberg [RN13].) The number of linear trails with correlation of magnitude at least τ is not too large and the total influence of trails with trail correlation of magnitude essentially smaller than τ is negligible.*

Röck and Nyberg [RN13] introduced the alternative key $\widehat{K} \in \mathbb{F}_2^l$ so that all the r round keys k_i can be derived linearly from $\widehat{K}$. If the key schedule is linear, $\widehat{K} = K$. In the case of a non-linear key schedule, the bits of the round key generated non-linearly from the master key bits can be seen as new bits. Hence, $\widehat{K}$ contains not only bits of K used to generate the round keys, but also bits associated to the non-linear relations on the bits of K.

Consequently, there is a linear relation between $\widehat{K} \in \mathbb{F}_2^l$ and the round keys, so that there must exist a linear function $f : (\mathbb{F}_2^n)^{r+1} \to \mathbb{F}_2^l$ that maps the round masks $\theta_0, \theta_1, \dots, \theta_r$ to a single mask $v = f(\theta_0, \theta_1, \dots, \theta_r)$ such that $v \cdot \widehat{K} = \theta_0 \cdot k_0 \oplus \cdots \oplus \theta_r \cdot k_r$ for all K. Under Assumption 1, let

$$\rho(v) = \sum_{\substack{f(\theta_0,\theta_1,\dots,\theta_r)=v \\ |C_\theta(K)|\geq\tau}} \prod_{j=1}^{r} C_{\theta_{i-1},\theta_i}.$$

Then the correlation of a linear hull (u, w) under K can be represented as

$$C(K) = \sum_{v \in V = \{v \in \mathbb{F}_2^l,\ \rho(v)\neq 0\}} (-1)^{v\cdot\widehat{K}} \rho(v).$$

To compute all possible values of $C(K)$ and their corresponding key classes, the trivial method is to traverse all 2^κ values of K. Röck and Nyberg [RN13] introduced a faster method by choosing a basis $\mathcal{B} = (b_0, b_1, \dots, b_{t-1})$ of $\mathrm{span}(V) \subset \mathbb{F}_2^l$, where t is the dimension of the vector space $\mathrm{span}(V)$. Denote B as the $t \times l$ matrix containing all these basis vectors and B^T as its transpose. Then each value $v \in V$ can be represented as $\boldsymbol{v} = (v_0, v_1, \dots, v_{t-1}) \in \mathbb{F}_2^t$ with $v = \sum_{j=0}^{t-1} v_j b_j = \boldsymbol{v}B$. Therefore, we can see that

$v \cdot \widehat{K} = \boldsymbol{v}B \cdot \widehat{K} = \boldsymbol{v} \cdot (\widehat{K}B^T) \overset{\text{def}}{=} \boldsymbol{v} \cdot \widehat{\boldsymbol{K}}$, where $\widehat{\boldsymbol{K}}$ is only t bits. Hence, by traversing all $\widehat{\boldsymbol{K}} \in \mathbb{F}_2^t$ and checking whether they can lead to a possible K, we can obtain all these key classes with their correlations $C(K)$.

Let $\mathcal{C} = \{c_0, c_1, \ldots, c_{|\mathcal{C}|-1}\}$ be the set of all possible values of $C(K)$. Then the whole key space is divided into $|\mathcal{C}|$ disjoint classes

$$\mathcal{K}(c_i) = \{K \in \mathbb{F}_2^\kappa \mid C(K) = c_i \in \mathcal{C}\}$$

for all $0 \le i \le |\mathcal{C}| - 1$. Note that it is equivalent to store $\widehat{\boldsymbol{K}} \in \mathbb{F}_2^t$ rather than K in $\mathcal{K}(c_i)$ when dealing with linear key schedules, as $\widehat{\boldsymbol{K}} = \widehat{K}B^T = KB^T$. With these key classes, Röck and Nyberg [RN13] constructed a statistical model by adopting $T_1 = N_0^K$ as a statistic, where N_0^K is the number of x satisfying the linear hull $u \cdot x \oplus w \cdot \mathcal{E}_K(x) = 0$ given N known data x. Then, $\mathcal{K}(c_i)$ is taken as the key class the secret key lies in if the index i maximizes

$$\log_2 \pi_i + N_0^K \log_2 p_i + (N - N_0^K) \log_2 (1 - p_i),$$

where $\pi_i = 2^{-\kappa}|\mathcal{K}(c_i)|$ and $p_i = (1 + c_i)/2$.

Given the value of the statistic T_1 evaluated under N known data, the decision function will output the key class in which the right one lies. However, it can make mistakes. Denote P_{ij} as the error probability that the decision function chooses i as its output while the right key lies in the j-th class. Thus, the error probability when the j-th class contains the right key is $\sum_{i \neq j} P_{ij}$. As shown in [RN13], this error probability is not only determined by N but also j. To theoretically evaluate N, Röck and Nyberg [RN13] regarded the average error probabilities over all keys as the total error probability $P_e = \sum_j \pi_j \sum_{i \neq j} P_{ij}$, where π_j is the proportion of each key class. When constructing the relation between P_e and N, they set each P_{ij} as $P_e/(|\mathcal{C}| - 1)$ with $|\mathcal{C}|$ representing the number of key classes. Then, to obtain P_{ij} when making the decision between the i-th and j-th classes, they can compute the data complexity N_{ij} using their statistical model. The data complexity N of the attack is then evaluated as the upper bound of all these N_{ij} in order to obtain P_e. For more details of their model, we refer to Röck and Nyberg [RN13].

In addition to this key recovery attack under the *direct attack* setting, Röck and Nyberg [RN13] also proposed attacks under the *basic* and *multiple related-key* settings. These two stronger settings are used with the aim of gaining more key information. For related keys (K, K') satisfying $\widehat{K} \oplus \widehat{K'} = \alpha \in \mathbb{F}_2^l$, they evaluated

$$\Delta_{K,\alpha} = C(K) - C(K') = \sum_{v \in V} (-1)^{v \cdot \widehat{K}} \rho(v) - \sum_{v \in V} (-1)^{v \cdot (\widehat{K} \oplus \alpha)} \rho(v).$$

Note that many terms in the sum cancel out. Hence, using related keys, we can reduce the number of v in the summation, and in most cases, reduce the number of possible values of the correlations. The difference $\Delta_{K,\alpha}$ can be calculated efficiently by using the basis $\mathcal{B}$. As in the case of direct attack setting, for ciphers with linear key schedules, we can directly choose the difference on $\widehat{\boldsymbol{K}}$.

Let $\mathcal{C}_\alpha = \{c_0, c_1, \ldots, c_{|\mathcal{C}_\alpha|-1}\}$ contain all possible values of $\Delta_{K,\alpha}$. Then the whole key space is divided into $|\mathcal{C}_\alpha|$ disjoint classes

$$\mathcal{K}_\alpha(c_i) = \{K \in \mathbb{F}_2^\kappa \mid \Delta_{K,\alpha} = c_i \in \mathcal{C}_\alpha\}$$

for all $0 \le i \le |\mathcal{C}_\alpha| - 1$. For the related-key setting, Röck and Nyberg [RN13] used the statistic $T_2 = N_0^K - N_0^{K'}$. The right key belongs to $\mathcal{K}_\alpha(c_i)$ if the index i maximizes

$$\ln \pi_i^\alpha - (T_2 - Nc_i/2)^2 / N,$$

where $\pi_i^\alpha = 2^{-\kappa}|\mathcal{K}_\alpha(c_i)|$. With this decision function and the statistic T_2, the model for the basic related-key setting was proposed under the assumption that N_0^K and $N_0^{K'}$ are independent.

In the multiple related-key setting, t differences $\alpha_0, \ldots, \alpha_{t-1} \in \mathbb{F}_2^l$ are chosen in such a way that they form a dual basis for $\mathcal{B}$, *i.e.*, $\alpha_j \cdot b_i = 1$ holds for $j = i$ and $\alpha_j \cdot b_i = 0$ otherwise. Since all basis vectors are independent, we can always solve this system if we can, for all j, find (K, K') such that $\widehat{K} \oplus \widehat{K'} = \alpha_j$ holds.

For ciphers with a linear key schedule, it is equivalent if we choose t differences α_j on $\widehat{\boldsymbol{K}}$ satisfying $\alpha_j = (0, \ldots, 0, 1, 0, \ldots, 0)$ with only one active bit at the j-th bit position. Then the statistic T_2 can be used under each difference α_j to decide in which key class the secret key lies.

Assume that the key class decided under difference α_j is $\mathcal{K}_{\alpha_j}(\eta_j)$. Then by combining the results for all $0 \leq j \leq t-1$, we can see that the right key must belong to

$$\mathcal{K}_{\mathcal{B}}(\boldsymbol{\eta}) = \bigcap_{0 \leq j \leq t-1} \mathcal{K}_{\alpha_j}(\eta_j).$$

Denote N_α as the data complexity of the basic related-key attack using the key difference α. Then the total data complexity is $\max_{0 \leq j \leq t-1} N_{\alpha_j} + \sum_{j=0}^{t-1} N_{\alpha_j}$. To obtain the total error probability P_e, the error probability of the j-th basic related-key attack is set as $P_e^{\alpha_j} = 1 - (1 - P_e)^{1/t}$.

Röck and Nyberg [RN13] pointed out that a shortcoming of their models is that the relation between N and P_e is not accurately described.

3 New Methodology and Accurate Statistical Models

In this section, we will introduce two kinds of new statistical models with accurate success probabilities under the direct attack, basic related-key, and multiple related-key settings.

These two kinds of statistical models follow the same methodology that the error probability is deduced using the cumulative distribution function (CDF) of the accurately approximated distribution of the statistic related to the data complexities. However, they exploit different decision functions. The threshold-based statistical models constructed in Sect. 3.1 are much easier to handle, but slightly less precise, compared with the MLE-based statistical models described in Sect. 3.2. In Sect. 4, we will perform experiments to verify the accuracy of these models. A detailed comparison of these two kinds of statistical models is shown in Appendix F.

3.1 Threshold-Based Statistical Models

3.1.1 Statistical Model in the Direct Attack Setting

Let $\mathcal{C} = \{c_0, c_1, \ldots, c_{|\mathcal{C}|-1}\}$ be the set of all possible values of $C(K)$ with the condition that $c_i < c_{i+1}$ for all $0 \leq i \leq |\mathcal{C}| - 2$, and let $\mathcal{K}(c_i)$ contain all keys K satisfying $C(K) = c_i$. With these key classes, we use the statistic

$$\mathcal{T}_1 = 2\frac{N_0^K}{N} - 1$$

to mount attacks in the direct attack setting, where N_0^K records how many x fulfill $u \cdot x \oplus w \cdot \mathcal{E}_K(x) = 0$ after collecting N known data x. Note that the data collected can be distinct or not; we will deal with both cases in our models.

Denote K^* as the right key. Then the hypothesis H_i is defined as $K^* \in \mathcal{K}(c_i)$, $0 \leq i \leq |\mathcal{C}| - 1$. Now we have to decide which one of these $|\mathcal{C}|$ different hypotheses is true. To deal with this multiple hypothesis testing problem, a decision function δ is adopted.

It takes the obtained $\mathcal{T}_1$ as input and outputs the index of the decided right key class by comparing $\mathcal{T}_1$ with one or two threshold values. The function δ is defined as[1]

$$\delta(\mathcal{T}_1) = \begin{cases} 0, & \text{if } \mathcal{T}_1 \leq (c_0 + c_1)/2, \\ |\mathcal{C}| - 1, & \text{if } \mathcal{T}_1 > (c_{|\mathcal{C}|-2} + c_{|\mathcal{C}|-1})/2, \\ i, & \text{if } (c_{i-1} + c_i)/2 < \mathcal{T}_1 \leq (c_i + c_{i+1})/2,\ 1 \leq i \leq |\mathcal{C}| - 2. \end{cases}$$

The error probability of δ choosing H_i when H_j is true is $P_{ij} = Pr[\delta(\mathcal{T}_1) = i|H_j]$.

Let $\mathcal{D}(\mathcal{T}_1|H_i)$ represent the distribution of the statistic $\mathcal{T}_1$ under the hypothesis H_i, and let $\mathcal{N}(\mu, \sigma^2)$ denote the normal distribution with expectation μ and variance σ^2. Then we have the following lemma:

Lemma 1. *For sufficiently large N, $\mathcal{D}(\mathcal{T}_1|H_i) = \mathcal{N}\left(c_i, \frac{1-c_i^2}{N} B\right)$ where*

$$B = \begin{cases} 1, & \text{for KP sampling,} \\ \frac{2^n - N}{2^n - 1}, & \text{for DKP sampling,} \end{cases}$$

and n is the length of plaintexts, assuming all events are i.i.d.[2] *random variables. KP sampling considers that the known plaintexts gathered may be repeated, whereas distinct known plaintexts are collected in the case of DKP sampling.*

Proof. Under the hypothesis H_i, the correlation of the linear hull $u \cdot x \oplus w \cdot \mathcal{E}_K(x) = 0$ is c_i and holds with probability $p_i = (1 + c_i)/2$. As we can see from [BN17, DR07, Mat93], in the case of KP sampling, N_0^K follows a binomial distribution with expectation Np_i and variance $Np_i(1 - p_i)$. In the case of DKP sampling, N_0^K follows a hypergeometric distribution with expectation Np_i and variance

$$Np_i(1 - p_i)\frac{2^n - N}{2^n - 1}.$$

The binomial distribution and hypergeometric distribution can be tightly approximated by the normal distribution when N is sufficiently large. Hence, in the case of KP sampling, $N_0^K \sim \mathcal{N}(Np_i, Np_i(1-p_i))$. When dealing with distinct plaintexts, $N_0^K \sim \mathcal{N}(Np_i, Np_i(1-p_i)\frac{2^n-N}{2^n-1})$. Therefore, we can obtain the distributions of $\mathcal{T}_1$ in both cases. □

With Lemma 1 and the decision function δ, we introduce Theorem 1 which states the relation between the data complexity and the total error probability of the attack in the direct attack setting.

Theorem 1. *Given N data, $\mathcal{K}(c_i)$ is taken as the right key class if $\delta(\mathcal{T}_1) = i$. For sufficiently large N, the total error probability is*

$$P_e = \sum_{j=0}^{|\mathcal{C}|-1} \pi_j \sum_{i,\ i \neq j} P_{ij}$$

[1] A similar decision function was adopted in [AR16, APSD20]. They also noticed that the correlation is key-dependent but did not describe the relation between error probabilities and data complexities. Hence, they did not construct a general attack model.

[2] independent and identically distributed

with $\pi_i = 2^{-\kappa}|\mathcal{K}(c_i)|$ *and*

$$P_{ij} = \begin{cases} \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_0+c_1}{2} - c_j\right)\right), & \text{if } i = 0, \\ 1 - \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_{|\mathcal{C}|-2}+c_{|\mathcal{C}|-1}}{2} - c_j\right)\right), & \text{if } i = |\mathcal{C}|-1, \\ \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_i+c_{i+1}}{2} - c_j\right)\right) & \\ \qquad -\Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_{i-1}+c_i}{2} - c_j\right)\right), & \text{if } 1 \le i \le |\mathcal{C}|-2, \end{cases}$$

where $\Phi(\cdot)$ *denotes the cumulative distribution function of* $\mathcal{N}(0,1)$ *and* B *is the constant defined in Lemma 1.*

Proof. Denote $U = \sqrt{\frac{N}{B(1-c_j^2)}}(\mathcal{T}_1 - c_j)$. Recall that $P_{ij} = \Pr[\delta(\mathcal{T}_1) = i|H_j]$ and $\mathcal{D}(\mathcal{T}_1|H_j) = \mathcal{N}(c_j, B(1-c_j^2)/N)$. Then $U \sim \mathcal{N}(0,1)$ if $\mathcal{T}_1 \sim \mathcal{D}(\mathcal{T}_1|H_j)$.

(1) When $i = 0$,

$$\begin{aligned} P_{ij} &= \Pr[\delta(\mathcal{T}_1) = 0 \mid H_j] \\ &= \Pr[\mathcal{T}_1 \le (c_0+c_1)/2 \mid \mathcal{T}_1 \sim \mathcal{N}(c_j, B(1-c_j^2)/N)] \\ &= \Pr\left[U \le \sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_0+c_1}{2} - c_j\right)\right] \\ &= \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_0+c_1}{2} - c_j\right)\right). \end{aligned}$$

(2) When $i = |\mathcal{C}| - 1$,

$$\begin{aligned} P_{ij} &= \Pr[\delta(\mathcal{T}_1) = |\mathcal{C}|-1 \mid H_j] \\ &= \Pr[\mathcal{T}_1 > (c_{|\mathcal{C}|-2}+c_{|\mathcal{C}|-1})/2 \mid \mathcal{T}_1 \sim \mathcal{N}(c_j, B(1-c_j^2)/N)] \\ &= \Pr\left[U > \sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_{|\mathcal{C}|-2}+c_{|\mathcal{C}|-1}}{2} - c_j\right)\right] \\ &= 1 - \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_{|\mathcal{C}|-2}+c_{|\mathcal{C}|-1}}{2} - c_j\right)\right). \end{aligned}$$

(3) When $1 \le i \le |\mathcal{C}| - 2$,

$$\begin{aligned} P_{ij} &= \Pr[\delta(\mathcal{T}_1) = i \mid H_j] \\ &= \Pr[(c_{i-1}+c_i)/2 < \mathcal{T}_1 \le (c_i+c_{i+1})/2 \mid \mathcal{T}_1 \sim \mathcal{N}(c_j, B(1-c_j^2)/N)] \\ &= \Pr\left[\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_{i-1}+c_i}{2} - c_j\right)\right. \\ &\qquad\qquad \left. < U \le \sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_i+c_{i+1}}{2} - c_j\right)\right] \\ &= \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_i+c_{i+1}}{2} - c_j\right)\right) \\ &\qquad - \Phi\left(\sqrt{\frac{N}{B(1-c_j^2)}}\left(\frac{c_{i-1}+c_i}{2} - c_j\right)\right). \end{aligned}$$

By the definition of P_{ij}, P_e follows from the Total Probability Theorem. □

According to Röck and Nyberg [RN13], the average key information learned from this attack is given by its Shannon entropy [Sha48]. Thus, on average $h = -\sum_i \pi_i \log_2(\pi_i)$ bits of key information can be recovered in the direct attack setting.

3.1.2 Statistical Model in the Basic and Multiple Related-Key Settings

In related-key settings, one can exploit different relations between correlations evaluated under related keys, such as their differences or sum. Following Röck and Nyberg [RN13], we consider their differences, namely $\Delta_{K,\alpha} = C(K) - C(K')$. In this way, one may obtain more key bits than the statistical model in the direct attack setting, as explained in Sect. 2.2. Let $\mathcal{C}_\alpha = \{c_0, c_1, \dots, c_{|\mathcal{C}_\alpha|-1}\}$ be the set containing all possible values of $\Delta_{K,\alpha}$ with the condition that $c_i < c_{i+1}$ for all $0 \le i \le |\mathcal{C}_\alpha| - 2$, and let $\mathcal{K}_\alpha(c_i)$ be the key class composed of all K satisfying $\Delta_{K,\alpha} = c_i$.

In the related-key setting, we use the statistic

$$\mathcal{T}_2 = 2\frac{N_0^K}{N} - 2\frac{N_0^{K'}}{N}$$

to recover the key bits, where N_0^K and $N_0^{K'}$ are evaluated under two independent data sets[3]. Denote the hypothesis H_i as $K^* \in \mathcal{K}_\alpha(c_i)$, $0 \le i \le |\mathcal{C}_\alpha| - 1$, where K^* denotes the right key. Then we are dealing with a similar hypothesis testing problem as in the direct attack setting. Hence, the function δ introduced in Sect. 3.1.1 can still be used here as a decision function, except that its input is now the value obtained by $\mathcal{T}_2$.

As before, in order to construct the relation between the data complexities and the total error probabilities, we have to determine $\mathcal{D}(\mathcal{T}_2|H_i)$.

Lemma 2. *For sufficiently large N, if N_0^K and $N_0^{K'}$ are calculated under two independent data sets,*

$$\mathcal{D}(\mathcal{T}_2|H_i) = \mathcal{N}(c_i, 2B/N),\ 0 \le i \le |\mathcal{C}_\alpha| - 1.$$

The constant B indicates whether or not distinct data is used, and its definition was given in Lemma 1.

Proof. Under the hypothesis H_i, we denote c_i^K as the correlation of $u \cdot x \oplus w \cdot \mathcal{E}_K(x) = 0$ and $c_i^{K'}$ as the correlation of $u \cdot x' \oplus w \cdot \mathcal{E}_{K'}(x') = 0$. Then $c_i = c_i^K - c_i^{K'}$. For sufficiently large N, according to Lemma 1, we have

$$N_0^K \sim \mathcal{N}\left(\frac{N(1+c_i^K)}{2}, \frac{N(1-(c_i^K)^2)}{4}B\right),$$

$$N_0^{K'} \sim \mathcal{N}\left(\frac{N(1+c_i^{K'})}{2}, \frac{N(1-(c_i^{K'})^2)}{4}B\right).$$

Since x and x' are chosen from two independent data sets, N_0^K and $N_0^{K'}$ are independently distributed. Therefore,

$$N_0^K - N_0^{K'} \sim \mathcal{N}\left(\frac{N(c_i^K - c_i^{K'})}{2}, \frac{N}{4}(2-(c_i^K)^2-(c_i^{K'})^2)B\right).$$

[3] A general form of $\mathcal{T}_2$ is $\mathcal{T}_2' = 2\frac{N_0^K}{N_1} - 2\frac{N_0^{K'}}{N_2}$, which means that N_0^K and $N_0^{K'}$ are obtained from two data sets with different size N_1 and N_2. Similar to Lemma 2, one can obtain its distribution $\mathcal{D}(\mathcal{T}_2'|H_i) = \mathcal{N}(c_i, (\frac{1}{N_1} + \frac{1}{N_2})B)$ by assuming that $(c_i^K)^2 \ll 1$ and $(c_i^{K'})^2 \ll 1$. Note that when $N_1 = N_2$, $\mathcal{T}_2'$ will have the smallest variance $2B/N_1$, which is exactly the same as that of $\mathcal{T}_2$. Considering that smaller variance will lead to higher success probability under the same data complexity, we directly adopted $\mathcal{T}_2$ as the statistic here, rather than $\mathcal{T}_2'$.

Since $c_i = c_i^K - c_i^{K'}$ and $(c_i^K)^2 + (c_i^{K'})^2 \ll 2$, we can see that

$$N_0^K - N_0^{K'} \sim \mathcal{N}\left(\frac{Nc_i}{2}, \frac{N}{2}B\right).$$

Thus, the distribution of $\mathcal{T}_2$ under H_i can be obtained. □

Using Lemma 2, the relation between the data complexity and the total error probability of the attack under the basic related-key setting is shown in Theorem 2. The proof is similar to the proof of Theorem 1, and is therefore omitted.

Theorem 2. *Given two independent data sets with size N, $\mathcal{K}_\alpha(c_i)$ is taken as the right key class if $\delta(\mathcal{T}_2) = i$. For sufficiently large N, the total error probability is*

$$P_e^\alpha = \sum_{j=0}^{|\mathcal{C}_\alpha|-1} \pi_j^\alpha \sum_{i,\ i\neq j} P_{ij}$$

with $\pi_i^\alpha = 2^{-\kappa}|\mathcal{K}_\alpha(c_i)|$ and P_{ij} can be computed by

$$P_{ij} = \begin{cases} \Phi\left(\sqrt{\frac{N}{2B}}\left(\frac{c_0+c_1}{2} - c_j\right)\right), & \text{if } i = 0, \\ 1 - \Phi\left(\sqrt{\frac{N}{2B}}\left(\frac{c_{|\mathcal{C}_\alpha|-2}+c_{|\mathcal{C}_\alpha|-1}}{2} - c_j\right)\right), & \text{if } i = |\mathcal{C}_\alpha| - 1. \\ \Phi\left(\sqrt{\frac{N}{2B}}\left(\frac{c_i+c_{i+1}}{2} - c_j\right)\right) & \\ \qquad -\Phi\left(\sqrt{\frac{N}{2B}}\left(\frac{c_{i-1}+c_i}{2} - c_j\right)\right), & \text{if } 1 \le i \le |\mathcal{C}_\alpha| - 2, \end{cases}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of $\mathcal{N}(0,1)$ and B is the constant defined in Lemma 1.

Using the attack in the basic related-key setting, $h_\alpha = -\sum_i \pi_i^\alpha \log_2(\pi_i^\alpha)$ bits of key information can be obtained on average.

To obtain more information about the key K, multiple key differences are used. Following Röck and Nyberg [RN13], t differences $\alpha_0, \ldots, \alpha_{t-1} \in \mathbb{F}_2^l$ are chosen that form a dual basis for $\mathcal{B}$. For each key difference α_j, we will proceed with an attack in the basic related-key setting using statistic $\mathcal{T}_2$ and obtain the right key class $\mathcal{K}_{\alpha_j}(\eta_j)$. If all these attacks succeed, the right key must belong to

$$\mathcal{K}_{\mathcal{B}}(\boldsymbol{\eta}) = \bigcap_{0\le j\le t-1} \mathcal{K}_{\alpha_j}(\eta_j),$$

so that we learn on average $h_{\boldsymbol{\eta}} = -\sum_{\boldsymbol{\eta}} \pi_{\boldsymbol{\eta}} \log_2 \pi_{\boldsymbol{\eta}}$ bits of key information, where $\pi_{\boldsymbol{\eta}} = 2^{-\kappa}|\mathcal{K}_{\mathcal{B}}(\boldsymbol{\eta})|$.

Note that the data encrypted by K in each basic related-key attack should be chosen independently rather than using the same data set. Otherwise, we cannot directly combine all these right key classes due to their dependency. Denote N_α as the size of data set in the basic related-key attack using the key difference α and P_e^α as the error probability. Then we need $2\sum_{j=0}^{t-1} N_{\alpha_j}$ data in total and the error probability is $P_e = 1 - \prod_{j=0}^{t-1}(1 - P_e^{\alpha_j})$.

3.2 MLE-Based Statistical Models

Here, we will introduce another kind of statistical models with the decision function used by Röck and Nyberg [RN13], however we will adopt the same methodology of deducing the relation between N and P_e.

As before, we will introduce statistical models under three settings: direct attack, basic related-key, and multiple related-key. For each setting, we will consider whether the data is drawn under KP sampling or DKP sampling.

3.2.1 Statistical Model in the Direct Attack Setting

Let $\mathcal{C} = \{c_0, c_1, \ldots, c_{|\mathcal{C}|-1}\}$ be the set of all possible values of $C(K)$ for all $0 \leq i \leq |\mathcal{C}| - 1$, and let $\mathcal{K}(c_i)$ contain all keys K satisfying $C(K) = c_i$. With these key classes, we use the statistic N_0^K to mount attacks in the direct attack setting. Note that N_0^K records how many x fulfill $u \cdot x \oplus w \cdot \mathcal{E}_K(x) = 0$ after collecting N known data x. It follows a binomial distribution with expectation Np_i and variance $Np_i(1-p_i)$ when data is drawn under KP sampling, where $p_i = (1 + c_i)/2$. Under DKP setting, it follows a hypergeometric distribution with same expectation but with variance $Np_i(1-p_i)\frac{2^n-N}{2^n-1}$.

Denote K^* as the right key. Then the hypothesis H_i is defined as $K^* \in \mathcal{K}(c_i)$, $0 \leq i \leq |\mathcal{C}| - 1$. Now we have to decide which one of these $|\mathcal{C}|$ different hypotheses is true after observing the value of N_0^K.

Results under KP Sampling. Röck and Nyberg [RN13] proposed a decision function based on the maximum likelihood estimate. By taking the prior information π_i of each hypothesis H_i into consideration, they gave the decision function $\delta_1^*(N_0^K)$ defined as the index i that maximizes

$$ML(i) = \log_2 \pi_i + N_0^K \log_2 p_i + (N - N_0^K) \log_2 (1 - p_i).$$

With this decision function, we can deduce the relation between N and P_e by adopting the same methodology as in Sect. 3.1. As before, we first obtain the error probability P_{ij} of accepting H_i when H_j is true, and then P_e can be obtained as $P_e = \sum_{j=0}^{|\mathcal{C}|-1} \pi_j \sum_{i,\ i \neq j} P_{ij}$ following the Total Probability Theorem.

To determine the error probability P_{ij}, the value range of N_0^K has to be obtained. Since $\delta_1^*(N_0^K) = i$, $ML(i) > ML(t)$ holds for any $t \neq i$. Therefore, for each $t \neq i$, we can obtain an interval of N_0^K, which is

$$\frac{\log_2 \pi_t - \log_2 \pi_i + N \log_2(1-p_t) - N \log_2(1-p_i)}{\log_2 p_i - \log_2(1-p_i) - \log_2 p_t + \log_2(1-p_t)} < N_0^K < N \text{ and } N_0^K \in \mathbb{Z}$$

when $p_i > p_t$ and

$$0 < N_0^K < \frac{\log_2 \pi_t - \log_2 \pi_i + N \log_2(1-p_t) - N \log_2(1-p_i)}{\log_2 p_i - \log_2(1-p_i) - \log_2 p_t + \log_2(1-p_t)} \text{ and } N_0^K \in \mathbb{Z}$$

when $p_t > p_i$.

By taking the intersection between the above $|C| - 1$ intervals, the value range of N_0^K can be obtained. Denote the minimum of its range as $N_{\min}^i$ and the maximum as $N_{\max}^i$. Then

$$P_{ij} = \Phi_{N,p_j}^b(N_{\max}^i) - \Phi_{N,p_j}^b(N_{\min}^i)$$

where Φ_{N,p_j}^b is the Cumulative Distribution Function (CDF) of binomial distribution with expectation Np_j and variance $Np_j(1-p_j)$. When implementing this model in practice, we found that the evaluation procedure is rather slow. Thus, we used the CDF of normal distribution with same expectation and variance instead to compute P_{ij}. Note that binomial distribution can be approximated by the normal distribution when N is extremely large. The validity of this statistical model as well as this approximation is confirmed in Sect. 4.4.

Results under DKP Sampling. N_0^K follows a hypergeometric distribution in this case. We tried to deduce the relation between N and P_e by using this accurate distribution based on the maximum likelihood estimate directly. However, it is difficult to determine the interval of N_0^K since there is no analytical solution of $ML(i) > ML(t)$ for any $t \neq i$. Thus, we use the normal distribution to approximate this distribution.

The decision function $\delta_2^*(N_0^K)$ is chosen as the index i that maximizes

$$ML(i) = \ln \pi_i - \frac{1}{2}\ln p_i - \frac{1}{2}\ln(1-p_i) - \frac{(N_0^K - Np_i)^2(2^n-1)}{2Np_i(1-p_i)(2^n-N)}.$$

To obtain P_{ij}, we have to deduce the interval of N_0^K. For each $t \neq i$, we can find that N_0^K fulfills $A_1(N_0^K)^2 + A_2 N_0^K + A_3 > 0$, where

$$\begin{aligned}
A_1 &= p_i(1-p_i) - p_t(1-p_t),\\
A_2 &= 2Np_t(1-p_t)p_i - 2Np_i(1-p_i)p_t,\\
A_3 &= N^2p_i(1-p_i)p_t^2 - N^2p_t(1-p_t)p_i^2\\
&\qquad - 2Np_i(1-p_i)p_t(1-p_t)\frac{2^n-N}{2^n-1}\left(\ln\frac{\pi_t}{\pi_i} - \frac{1}{2}\ln\frac{p_t}{p_i} - \frac{1}{2}\ln\frac{1-p_t}{1-p_i}\right).
\end{aligned}$$

By solving these $|C|-1$ inequalities and noticing that $0 < N_0^K < N$, we can obtain the interval of N_0^K. Then P_{ij} can be obtained using the CDF of the normal distribution $\mathcal{N}\left(Np_j, Np_j(1-p_j)\frac{2^n-N}{2^n-1}\right)$.

3.2.2 Statistical Models in Basic/Multiple Related-Key Settings

Here we consider the differences between the correlations under related key pairs as before. Let $\mathcal{C}_\alpha = \{c_0, c_1, \ldots, c_{|\mathcal{C}_\alpha|-1}\}$ be the set containing all possible values of $\Delta_{K,\alpha} = C(K) - C(K')$, and let $\mathcal{K}_\alpha(c_i)$ be the key class containing all K that satisfies $\Delta_{K,\alpha} = c_i$.

We adopt the statistic $N_\alpha = N_0^K - N_0^{K'}$ to mount key recovery attacks, where N_0^K and $N_0^{K'}$ are evaluated under two independent data sets. Hence, according to Lemma 2, it follows a normal distribution $\mathcal{N}\left(\frac{Nc_i}{2}, \frac{N}{2}B\right)$ where B is the constant defined in Lemma 1.

The decision function $\delta_3^*(N_\alpha)$ based on the maximum likelihood estimate is already given in [RN13], which outputs the index i maximizing

$$\ln \pi_i^\alpha - \frac{(N_\alpha - Nc_i/2)^2}{NB}.$$

When $\delta_3^*(N_\alpha) = i$ and j is the right index, the value range of N_α can be obtained by intersecting all intervals of N_α under each $t \neq i$. The interval of N_α when $c_i > c_t$ is

$$\frac{B\ln\pi_t^\alpha - B\ln\pi_i^\alpha - Nc_t^2/4 + Nc_i^2/4}{c_i - c_t} < N_\alpha < N,$$

and the interval when $c_i < c_t$ is

$$0 < N_\alpha < \frac{B\ln\pi_t^\alpha - B\ln\pi_i^\alpha - Nc_t^2/4 + Nc_i^2/4}{c_i - c_t}.$$

Using the CDF of normal distribution $\mathcal{N}\left(\frac{Nc_j}{2}, \frac{N}{2}B\right)$, we can obtain P_{ij} under both KP and DKP sampling. Then the total error probability P_e in the basic related-key setting can be obtained. By evaluating $P_e^{\alpha_j}$ under each key difference α_j, one can obtain the error probability P_e in the multiple related-key setting with $P_e = 1 - \prod_{j=0}^{t-1}(1 - P_e^{\alpha_j})$.

4 Application to TinyJAMBU

4.1 Brief Introduction to TinyJAMBU

TinyJAMBU [WH19] is a family of AEAD algorithms submitted to the NIST LWC Standardization project and has been chosen as one of the 32 second-round candidates.

The design is inspired by JAMBU [WH15], a third-round candidate of the CAESAR competition.

One year after the second-round candidates were selected, only one cryptanalysis result on TinyJAMBU was announced: Saha *et al.* [SSS+20] showed weaknesses in the original TinyJAMBU v1. To address these weaknesses, the designers proposed the updated version TinyJAMBU v2 [WH21] in September 2020, which extends all 384-round permutations used in the original version into 640 rounds. In March 2021, TinyJAMBU v2 was selected as one of the ten finalists of NIST LWC Standardization project [Nat21].

The TinyJAMBU AEAD mode can be divided into four phases: initialization (key setup and nonce setup), AD processing, encryption, and finalization. Both TinyJAMBU v1 and v2 use the same keyed permutation $\mathcal{P}_b : \mathbb{F}_2^{128} \rightarrow \mathbb{F}_2^{128}$ mapping a 128-bit state $(s_{127}, s_{126}, \ldots, s_0)$ into $(z, s_{127}, \ldots, s_1)$ with $z = s_0 \oplus s_{47} \oplus (\sim (s_{70} \& s_{85})) \oplus s_{91} \oplus b$, where b is one bit of the master key K.

Following Saha *et al.* [SSS+20], we use the term "tag generation" to refer to the part of the finalization phase after the keyed permutation $\mathcal{P}_b[l_2]$. Note that tag generation uses only the l_1-round keyed permutation, where $l_1 = 384$ for TinyJAMBU v1 and $l_1 = 640$ for TinyJAMBU v2.

Given a 96-bit Nonce, a key K of 128, 192, and 256 bits, TinyJAMBU processes the message M and the associated data A using a 128-bit internal state and keyed permutations $\mathcal{P}_b[l_1]$ and $\mathcal{P}_b[l_2]$. The TinyJAMBU authenticated encryption mode is illustrated in Fig. 1. Detailed parameters and security goals under unique nonces are listed in Table 3.

Denote κ as the length of master key $K = (k_{\kappa-1}, k_{\kappa-2}, \ldots, k_0)$. In each phase, an l-round $\mathcal{P}_b[l]$ is used and $b = k_{i \bmod \kappa}$ for the i-th round $(0 \leq i \leq l-1)$.

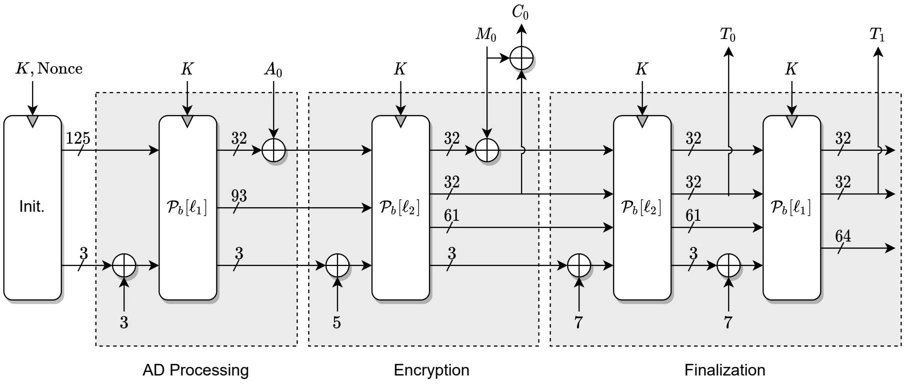


Figure 1: Structure of TinyJAMBU.

Table 3: TinyJAMBU and its security goal with a unique nonce [WH19, WH21].

Version	Key Size	State Size	Nonce Size	Tag Size	l_1	l_2	Security Goal Encr.	Security Goal Auth.
v1	128	128	96	64	384	1024	112-bit	64-bit
	192	128	96	64	384	1152	168-bit	64-bit
	256	128	96	64	384	1280	224-bit	64-bit
v2	128	128	96	64	640	1024	112-bit	64-bit
	192	128	96	64	640	1152	168-bit	64-bit
	256	128	96	64	640	1280	224-bit	64-bit

The designers restrict the amount of messages (associated data, plaintext, or ciphertext) per key to 2^{50} bytes, and each message is at least 8 bytes. In other words, TinyJAMBU can be invoked at most 2^{47} times under the same key if each message consists of 8 bytes (*i.e.,* two message blocks).

4.2 Searching All Linear Trails in a Given Hull for TinyJAMBU

In this paper, we focus on the linear cryptanalysis of the round-reduced permutation $\mathcal{P}_b[l_1]$ used in the tag generation phase. We aim to find a linear hull that only has non-zero masks on the tags $T_0||T_1$. The masks on the other state bits are all set to zero, as illustrated in Fig. 2. Linear correlations in the tag were already explored by Saha *et al.* [SSS+20], however they were unable to turn their observations into an attack on TinyJAMBU v1. We show how using this kind of linear hull, key recovery attacks can be mounted with the statistical models[4] introduced in Sect. 3 after gathering enough information on the tags $T_0||T_1$.

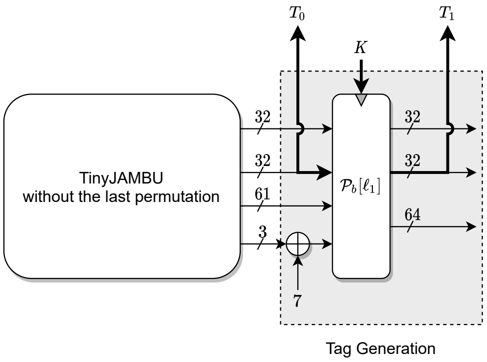


Figure 2: Linear hull for TinyJAMBU. Masks on normal lines are all-zero, while those on thick lines are non-zero.

To find such a linear hull, the trails comprising this hull should be obtained first. Since the work by Mouha *et al.* [MWGP11], many automatic search algorithms based on Mixed-Integer Linear Programming (MILP) have been proposed, such as [FWG+16, SSS+20, SSS+19, SHS+13, SHW+14a, SHW+14b, WWH+13]. In this paper, we also use MILP to help us to search for linear trails of TinyJAMBU. In order to focus on the main part of our model, we refer to Mouha *et al.* [MWGP11] for the detailed models of the basic operations.

Denote the 32-bit non-zero input (resp. output) mask illustrated in Fig. 2 as λ_0 (resp. λ_1). Note that bits of T_1 can be represented by non-linear Boolean functions in the bits of the input to $\mathcal{P}_b[l_1]$ and K. A direct way to evaluate the correlation of the linear hull (λ_0, λ_1) is to represent the approximation $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1$ as a non-linear Boolean function in the bits of the input to $\mathcal{P}_b[l_1]$ and K. If we can compute the correlation of this Boolean function, the correlation of (λ_0, λ_1) under K can then be obtained.

Unfortunately, there is no polynomial-time algorithm for computing the correlation of a Boolean function with a degree higher than two, as pointed out in [SSS+20, SSS+19]. Similar to the search algorithm proposed by Saha *et al.* [SSS+20], we make the assumption in our search algorithm that each output bit of the AND gate affected by a key bit can be regarded as a fresh new bit. This assumption is reasonable since key bits are randomly chosen. We also experimentally verified this assumption in Appendix C, which confirms its validity. Then in our search algorithm, each application of $\mathcal{P}_b$ introduces a new variable. Therefore, $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1$ can be expressed as a quadratic Boolean function in the bits of T_0 and these new variables.

[4] These models are applicable here since $\mathcal{P}_b[l_1]$ can be seen as an SPN cipher with a one-bit round key that is XORed to part of the 128-bit internal state, as we mentioned in Sect. 2.1.

In our MILP model, we declare $\mathcal{X}^r = \{\mathcal{X}^r_{127}, \mathcal{X}^r_{126}, \dots, \mathcal{X}^r_0\} \in \mathbb{F}_2^{128}$ with $0 \leq r \leq R$ to record input masks on r-th round and $\mathcal{Y}^r_0, \mathcal{Y}^r_1 \in \mathbb{F}_2$ with $0 \leq r \leq R-1$ to record the input masks of AND gate in r-th round when searching for R-round trails. According to the mask propagation rules, the following property holds:

Property 1. The output mask of the r-th round $\mathcal{X}^{r+1}$ can be computed as

$$\begin{aligned}
&\mathcal{X}^{r+1}_{127} = \mathcal{X}^r_0,\ \mathcal{X}^{r+1}_{46} = \mathcal{X}^r_{47} \oplus \mathcal{X}^r_0,\ \mathcal{X}^{r+1}_{90} = \mathcal{X}^r_{91} \oplus \mathcal{X}^r_0\\
&\mathcal{X}^{r+1}_{j-1} = \mathcal{X}^r_j,\ \forall\ j \in \{0,1,\dots,127\} \backslash \{0,47,70,85,91\}\\
&\mathcal{X}^{r+1}_{69} = \mathcal{X}^r_{70} \oplus \mathcal{Y}^r_0,\ \mathcal{X}^{r+1}_{84} = \mathcal{X}^r_{85} \oplus \mathcal{Y}^r_1
\end{aligned}$$

where $\mathcal{Y}^r_0$ and $\mathcal{Y}^r_1$ can take any value in $\mathbb{F}_2$ if $\mathcal{X}^r_0 = 1$, but $\mathcal{Y}^r_0 = 0 = \mathcal{Y}^r_1$ if $\mathcal{X}^r_0 = 0$. In other words, $\mathcal{Y}^r_0 \leq \mathcal{X}^r_0$ and $\mathcal{Y}^r_1 \leq \mathcal{X}^r_0$ should be satisfied.

Our search algorithm for a single linear trail has the objective to minimize the number of active AND gates in the hope of discovering a trail with a high correlation, such as the trail proposed in submission document of TinyJAMBU [WH19]. This is due to the fact that the Boolean function containing an active AND gate

$$f(x_0, x_1) = x_0 \& x_1 \oplus a_0 x_0 \oplus a_1 x_1$$

has correlation $(-1)^{a_0 a_1} 2^{-1}$, where $a_i \in \mathbb{F}_2$ are known coefficients and $x_i \in \mathbb{F}_2$. However, when multiple active AND gates are contained in the Boolean function, its absolute correlation is not always related with the number of AND gates due to the dependency between them. This has already been pointed out in several recent papers [SSS+20, SSS+19, STSH20]. For an illustration, see Example 1.

Example 1. Let $f(x_0, x_1, x_2) = x_0 \& x_1 \oplus x_1 \& x_2 \oplus a_0 x_0 \oplus a_1 x_1 \oplus a_2 x_2$ be a Boolean function where all a_i are known coefficients. One may say that its absolute correlation $|\mathrm{Cor}(f)|$ is 2^{-2} since there are two AND gates. However, due to the dependency between $x_0 \& x_1$ and $x_1 \& x_2$, $|\mathrm{Cor}(f)| = 2^{-1}$ if $a_0 = a_2$; otherwise, $|\mathrm{Cor}(f)| = 0$.

To deal with such a dependency, Shi *et al.* [SSS+19] proposed an algorithm that can transform a quadratic Boolean function into disjoint form. In other words, the input bits for each AND gate are independent in the transformed function. Denote $f(x_0, x_1, \dots, x_n)$ as the quadratic Boolean function

$$x_0 \& x_1 \oplus x_1 \& x_2 \oplus \cdots \oplus x_{n-1} \& x_n \oplus a_0 x_0 \oplus \cdots \oplus a_n x_n,$$

where all a_j are known coefficients. For such a quadratic Boolean function, each AND gate is chained with another one, and thus we call it a *Boolean function with chained AND gates.*

Proposition 1. *For the R-round keyed permutation $\mathcal{P}_b[R]$ in the tag generation phase of TinyJAMBU under key $K = (k_{\kappa-1}, k_{\kappa-2}, \dots, k_0)$ with key length $\kappa \in \{128, 192, 256\}$, $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1$ can be divided into 15 disjoint Boolean functions f_s. Moreover, f_s contains several Boolean functions with chained AND gates.*

Proof. Let $x^i = (x^i_{127}, x^i_{126}, \dots, x^i_0) \in \mathbb{F}_2^{128}$ represent the input value of $\mathcal{P}_b$ in the i-th round and let $x^{i+1} = \mathcal{P}_b(x^i)$ with $b = k_{i \bmod \kappa}$. Equivalently,

$$x^{i+1}_j = x^i_{j+1},\ \forall\ 0 \leq j \leq 126;\ x^{i+1}_{127} = x^i_0 \oplus x^i_{47} \oplus x^i_{70} \& x^i_{85} \oplus 1 \oplus x^i_{91} \oplus k_{i \bmod \kappa}.$$

Using Property 1, for the r-th round ($0 \le r \le R-1$), we have

$$\begin{aligned}
\mathcal{X}^r \cdot x^r \oplus \mathcal{X}^{r+1} \cdot x^{r+1} &= \bigoplus_{j=0}^{127} \mathcal{X}_j^r x_j^r \oplus \bigoplus_{j=0}^{127} \mathcal{X}_j^{r+1} x_j^{r+1} \\
&= \bigoplus_{j=1}^{127} (\mathcal{X}_j^r \oplus \mathcal{X}_{j-1}^{r+1}) x_j^r \oplus \mathcal{X}_{127}^{r+1} x_{127}^{r+1} \oplus \mathcal{X}_0^r x_0^r \\
&= \mathcal{X}_0^r (x_{47}^r \oplus x_{91}^r \oplus \mathcal{Y}_0^r x_{70}^r \oplus \mathcal{Y}_1^r x_{85}^r \oplus x_{127}^{r+1} \oplus x_0^r) \\
&= \mathcal{X}_0^r (x_{70}^r \& x_{85}^r \oplus \mathcal{Y}_0^r x_{70}^r \oplus \mathcal{Y}_1^r x_{85}^r \oplus 1 \oplus k_{r \bmod \kappa}).
\end{aligned}$$

Combining all these R equations, we get

$$\begin{aligned}
\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1 &= \mathcal{X}^0 \cdot x^0 \oplus \mathcal{X}^R \cdot x^R \\
&= \bigoplus_{r=0}^{R-1} (\mathcal{X}^r \cdot x^r \oplus \mathcal{X}^{r+1} \cdot x^{r+1}) \\
&= \bigoplus_{r=0}^{R-1} \mathcal{X}_0^r (x_{70}^r \& x_{85}^r \oplus \mathcal{Y}_0^r x_{70}^r \oplus \mathcal{Y}_1^r x_{85}^r \oplus 1 \oplus k_{r \bmod \kappa}).
\end{aligned}$$

Furthermore, we can see that $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1$ consists of 15 disjoint Boolean functions, *i.e.*, $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1 = \bigoplus_{s=0}^{14} f_s$, where

$$\begin{aligned}
f_s =& \bigoplus_{j=0}^{t_s} \mathcal{X}_0^{s+15j} (x_{70}^{s+15j} \& x_{85}^{s+15j}) \\
& \oplus \bigoplus_{j=0}^{t_s} \mathcal{X}_0^{s+15j} (\mathcal{Y}_0^{s+15j} x_{70}^{s+15j} \oplus \mathcal{Y}_1^{s+15j} x_{85}^{s+15j} \oplus 1 \oplus k_{s+15j \bmod \kappa})
\end{aligned}$$

and $t_s = \lfloor \frac{R-1}{15} \rfloor$ for all $0 \le s \le ((R-1) \bmod 15)$; otherwise, $t_s = \lfloor \frac{R-1}{15} \rfloor - 1$, where $\lfloor y \rfloor$ denotes the greatest integer less than or equal to y.

Recall that $x_{70}^r = x_{85}^{r-15}$ for $r \ge 15$, so that $x_{70}^{s+15j} = x_{85}^{s+15(j-1)}$ for $j \ge 1$. Hence, by replacing all x_{70}^{s+15j} with $x_{85}^{s+15(j-1)}$ when $j \ge 1$ in f_s, we obtain

$$\begin{aligned}
f_s =& \mathcal{X}_0^s (x_{70}^s \& x_{85}^s) \oplus \bigoplus_{j=1}^{t_s} \mathcal{X}_0^{s+15j} (x_{85}^{s+15(j-1)} \& x_{85}^{s+15j}) \\
& \oplus \mathcal{X}_0^s \mathcal{Y}_0^s x_{70}^s \oplus \bigoplus_{j=1}^{t_s} \left[\mathcal{X}_0^{s+15(j-1)} \mathcal{Y}_1^{s+15(j-1)} \oplus \mathcal{X}_0^{s+15j} \mathcal{Y}_0^{s+15j} \right] x_{85}^{s+15(j-1)} \\
& \oplus \mathcal{X}_0^{s+15t_s} \mathcal{Y}_1^{s+15t_s} x_{85}^{s+15t_s} \oplus \bigoplus_{j=0}^{t_s} \mathcal{X}_0^{s+15j} (1 \oplus k_{s+15j \bmod \kappa}).
\end{aligned}$$

From the above equation, we can see that f_s is composed of several Boolean functions with chained AND gates, since the bit x_{85}^{s+15j} may be chained with $x_{85}^{s+15(j-1)}$ and $x_{85}^{s+15(j+1)}$ if $\mathcal{X}_0^{s+15j} = 1 = \mathcal{X}_0^{s+15(j+1)}$. □

Song *et al.* [STSH20] introduced the following lemma that directly reveals the relation between the number of chained AND gates and the absolute correlations of Boolean functions based on the same idea as Shi *et al.* [SSS+19].

Lemma 3. *(Lemma 1 and 2 in [STSH20].) Denote the Boolean function with chained AND gates as*

$$f(x_0, \ldots, x_n) = x_0 \& x_1 \oplus x_1 \& x_2 \oplus \cdots \oplus x_{n-1} \& x_n \oplus a_0 x_0 \oplus \cdots \oplus a_n x_n.$$

(1) When n is odd, $|\mathrm{Cor}(f)| = 2^{-(n+1)/2}$.

(2) When n is even, $|\mathrm{Cor}(f)| = 2^{-n/2}$ *if* $\bigoplus_{j=0}^{n/2} a_{2j} = 0$*; otherwise,* $|\mathrm{Cor}(f)| = 0$.

The proof of Lemma 3 was given by Song *et al.* [STSH20]. Since we have to compute the correlation of the linear hull based on these trails, the sign of the correlation is also important here, but not given in [STSH20]. The proof of Corollary 1 is similar to the proof of Lemma 3, and is given in Appendix B.

Corollary 1. *Denote the sign of the correlation* $\mathrm{Cor}(f)$ *as* $\mathrm{Sign}(f)$*. Then*

$$\mathrm{Sign}(f) = \prod_{i=0}^{t-1} (-1)^{\left(\bigoplus_{j=0}^{i} a_{2j}\right) a_{2i+1}}$$

where $t = (n+1)/2$ *if* n *is odd,* $t = n/2$ *if* n *is even, and* $\bigoplus_{j=0}^{t} a_{2j} = 0$.

Since we aim to find all trails in a given linear hull, the actual correlation can be computed using Lemma 3 and Corollary 1 after we find a trail. Hence, in our MILP-based search algorithm, we still count all active AND gates with the aim of accelerating the search process.

According to Lemma 3 and Corollary 1, the correlation of such a Boolean function is only related to the number of AND gates and the coefficients a_i. Recall the definition of f_s described in the proof of Proposition 1. There can be trails having different $\mathcal{X}_0^r$, $\mathcal{Y}_0^r$ or $\mathcal{Y}_1^r$ leading to same coefficients of f_s benefiting from the XORed form of coefficients of $x_{85}^{s+15(j-1)}$ for $1 \le j \le t_s$, and their total contributions to the correlation of f_s is equivalent to that of one of them due to Lemma 3 and Corollary 1. To clarify this, consider the following example.

Example 2. Let $f(x_0, x_1, x_2) = x_0 \& x_1 \oplus x_1 \& x_2 \oplus x_0 \oplus (a_0 \oplus a_1) x_1 \oplus x_2$ denote a Boolean function having a similar form as f_s, where a_i are known coefficients. Due to Lemma 3 and Corollary 1, the correlation of f is $\mathrm{Cor}(f) = (-1)^{a_0 \oplus a_1} 2^{-1}$. Hence, $\mathrm{Cor}(f)$ under $(a_0, a_1) = (0, 0)$ is the same as under $(a_0, a_1) = (1, 1)$. These two trails are considered to be different since they have different a_i in our model. Nevertheless, their contribution to the hull is only $\mathrm{Cor}(f)$, rather than $2\mathrm{Cor}(f)$.

If all coefficients of f_s comprising $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1$ are the same among the trails, their total contribution to the correlation of this hull will be equivalent to contribution of one of the trails. This can be deduced from Lemma 3 and Corollary 1 due to the linear hull effect of such Boolean functions. In this paper, all these trails are said to be in the same *equivalence class.* To identify which trails belong to the same equivalence class, we introduce the following proposition.

Proposition 2. *Suppose that we have obtained several trails with masks* $\mathcal{X}^r$ *for all* $0 \le r \le R$ *and* $\mathcal{Y}_0^r, \mathcal{Y}_1^r$ *for all* $0 \le r \le R-1$*. They are in the same equivalence class if for any* $1 \le j \le t_s$ *and for any* $0 \le s \le 14$*,*

$$\mathcal{X}_0^s,\ \mathcal{X}_0^s \mathcal{Y}_0^s,\ \mathcal{X}_0^{s+15t_s} \mathcal{Y}_1^{s+15t_s},\ \mathcal{X}_0^{s+15j},\ \mathcal{X}_0^{s+15(j-1)} \mathcal{Y}_1^{s+15(j-1)} \oplus \mathcal{X}_0^{s+15j} \mathcal{Y}_0^{s+15j}$$

are the same among these trails.

In order to find all trails comprising the hull, after obtaining a trail, we have to add extra constraints in the MILP model to remove this trail as well as the trails belonging to the same equivalence class. This can be done with Property 2. The idea of this property was first proposed by Balas *et al.* [BJ72] and also used in [SHW+14a]. Before using this property, additional variables should be declared in order to restrict the value of

$\mathcal{X}_0^{s+15j}\mathcal{Y}_0^{s+15j}$ and $\mathcal{X}_0^{s+15j}\mathcal{Y}_1^{s+15j}$ for all $0 \leq j \leq t_s$ and for all $0 \leq s \leq 14$. Take $\mathcal{X}_0^s\mathcal{Y}_0^s$ as an example. Denote $\mathcal{A}_0^s \in \mathbb{F}_2$ as the extra variable, then the constraints

$$\begin{cases} \mathcal{X}_0^s \geq \mathcal{A}_0^s \\ \mathcal{Y}_0^s \geq \mathcal{A}_0^s \\ \mathcal{X}_0^s + \mathcal{Y}_0^s \leq \mathcal{A}_0^s + 1 \end{cases}$$

should be added to the model. In other words, $\mathcal{A}_0^s = \mathcal{X}_0^s\mathcal{Y}_0^s$. Hence, we can restrict the value of $\mathcal{X}_0^s\mathcal{Y}_0^s$ by restricting the value of $\mathcal{A}_0^s$ using Property 2.

Property 2. Let $\mathcal{Z} = (\mathcal{Z}_0, \mathcal{Z}_1, \ldots, \mathcal{Z}_{n-1}) \in \mathbb{F}_2^n$ represent all the coefficients shown in Proposition 2. Denote $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{n-1}) \in \mathbb{F}_2^n$ as a solution. Then the constraint

$$\sum_{i=0}^{n-1}[\sigma_i + (-1)^{\sigma_i}\mathcal{Z}_i] \geq 1$$

can be used to remove the solution $\mathcal{Z} = \sigma$.

The search algorithm for all trails in a given linear hull for the round-reduced $\mathcal{P}_b[l_1]$ of TinyJAMBU is illustrated in Algorithm 1. The masks of the linear hull should be determined in advance, and a different choice of (λ_0, λ_1) may influence the processing time of our algorithm. Our choice of (λ_0, λ_1) follows a two-step heuristic strategy. Firstly, we use the model proposed by Saha *et al.* [SSS+20] to search as many trails with high correlation as possible, and record corresponding linear hulls (λ_0, λ_1). Then for each linear hull, we try to find all trails comprising it using our Algorithm 1.

Algorithm 1: Searching All Trails in a Given Linear Hull for $\mathcal{P}_b[l_1]$

```
input  : R: number of rounds;
         $(\lambda_0, \lambda_1)$: 32-bit input and output masks of linear hull;
output : $\mathcal{L}$: a list containing all trails in this hull;
         $C(\mathcal{L})$: list with the signed correlations of the corresponding trails;
Declare an empty MILP model $\mathcal{M}$;
Declare variables $\mathcal{X}_i^r \in \mathbb{F}_2$ with $0 \leq i \leq 127$ for each $0 \leq r \leq R$;
Declare variables $\mathcal{Y}_j^r \in \mathbb{F}_2$ with $0 \leq j \leq 1$ for each $0 \leq r \leq R-1$;
for $r \leftarrow 0$ to $R-1$ do
    Add constraints for $r$-th round based on Property 1;
$\mathcal{M}.addCon(\ \mathcal{X}^0 = \lambda_0 \lll 64,\ \mathcal{X}^R = \lambda_1 \lll 64\ )$; // $\mathcal{X}^r = (\mathcal{X}_{127}^r, \mathcal{X}_{126}^r, \ldots, \mathcal{X}_0^r)$
$\mathcal{M}.addObj(\ \sum_{r=0}^{R-1} \mathcal{X}_0^r\ )$; // MILP finds minimum value of objective.
$\mathcal{L} \leftarrow [\ ],\ C(\mathcal{L}) \leftarrow [\ ]$;
Solve $\mathcal{M}$ to obtain a solution;
while $\mathcal{M}$ has a solution $\sigma$ do
    Compute the actual correlation $C(\sigma)$ using Lemma 3 and Corollary 1;
    if $|C(\sigma)| \neq 0$ then
        Add $\sigma$ into $\mathcal{L}$ and $C(\sigma)$ into $C(\mathcal{L})$;
    Update $\mathcal{M}$ by adding the extra condition given in Property 2;
    Solve $\mathcal{M}$ to obtain another solution;
```

4.3 Collecting Key Bits Involved in a Linear Trail

After obtaining all trails consisting of the given linear hull using Algorithm 1, the key bits involved in each linear trail should be collected in order to acquire key information with the statistical models introduced in Sect. 3. Saha *et al.* [SSS+20] proposed a similar open question about the existence of weak keys with a strong correlation. Our analysis here can help to find the answer to this question.

Using Lemma 3, each f_s comprising $\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1$ is approximated to

$$\bigoplus_{j=0}^{t_s} \mathcal{X}_0^{s+15j}(1 \oplus k_{s+15j \bmod \kappa}).$$

Consequently, according to Proposition 1, we can obtain

$$\lambda_0 \cdot T_0 \oplus \lambda_1 \cdot T_1 \approx \bigoplus_{s=0}^{14} \bigoplus_{j=0}^{t_s} \mathcal{X}_0^{s+15j}(1 \oplus k_{s+15j \bmod \kappa}) = \bigoplus_{r=0}^{R-1} \mathcal{X}_0^{r}(1 \oplus k_{r \bmod \kappa}).$$

Therefore, the master key bit k_j is involved in the linear trail $(\mathcal{X}^0, \mathcal{X}^1, \dots, \mathcal{X}^R)$ only if $\bigoplus_{r \in \mathcal{J}} \mathcal{X}_0^r = 1$, $\mathcal{J} = \{r \mid j = r \bmod k,\ 0 \le r \le R-1\}$.

4.4 Experimental Verification of our Statistical Models Using TinyJAMBU

To verify the statistical models introduced in Sect. 3, we now implement key recovery attacks in the direct attack, basic related-key, and multiple related-key settings on the 256-round keyed permutation of TinyJAMBU [WH19], respectively.

Using the search algorithm introduced in Sect. 4.2, we found a linear hull containing four trails for the 256-round permutation using the 128-bit $K = (k_{127}, \dots, k_0)$, as shown in Table 4. The key bits involved are derived using the method introduced in Sect. 4.3.

Table 4: Linear trails for 256-Round $\mathcal{P}_b$ of TinyJAMBU, where the input mask involves bits 7, 30, 37, 44, 54, 64, 77, 81, 84, 91, 98, 118, and 121, and the output mask involves bit 64.

Key Bits Involved	Correlation
7, 27, 30, 37, 44, 81, 111, 118	$+2^{-10}$
6, 7, 27, 30, 37, 44, 81, 111, 118	-2^{-11}
7, 21, 27, 30, 37, 44, 81, 111, 118	-2^{-11}
6, 7, 21, 27, 30, 37, 44, 81, 111, 118	$+2^{-11}$

Since the key schedule of TinyJAMBU is linear, the alternative key $\widehat{K}$ is equal to K. As pointed out in Sect. 2.2, to accelerate the computation of all possible values of $C(K)$, we can directly traverse all $\widehat{\boldsymbol{K}} = \widehat{K}B^T = (ek_2, ek_1, ek_0)$ where $ek_2 = k_6$, $ek_1 = k_7 \oplus k_{27} \oplus k_{30} \oplus k_{37} \oplus k_{44} \oplus k_{81} \oplus k_{111} \oplus k_{118}$, $ek_0 = k_{21}$. Then the key classes containing corresponding $\widehat{\boldsymbol{K}}$ can be obtained, which are

$$\mathcal{K}(c_0 = -2.5 \cdot 2^{-10}) = \{7\}, \mathcal{K}(c_1 = -0.5 \cdot 2^{-10}) = \{2, 3, 6\},$$
$$\mathcal{K}(c_2 = +0.5 \cdot 2^{-10}) = \{0, 1, 4\}, \mathcal{K}(c_3 = +2.5 \cdot 2^{-10}) = \{5\}.$$

Note that we only need to store the values of $\widehat{\boldsymbol{K}}$ rather than K in these key classes for ciphers with a linear key schedule, as explained in Sect. 2.2.

4.4.1 Experimental Verification of Threshold-Based Models

With the key classes $\mathcal{K}(c_i)$, key recovery attacks under the direct attack setting can be mounted using the threshold-based statistical model introduced in Sect. 3.1.1. In each experiment, we choose different values for N, compute $\mathcal{T}_1$, and then check whether the right key lies in $\mathcal{K}(c_i)$ with $i = \delta(\mathcal{T}_1)$. The attack succeeds if δ returns the right key class. After repeating this experiment 2000 times, we can obtain the experimental error probabilities $\widehat{P_e}$ while the theoretical probabilities P_e can be obtained by Theorem 1.

Regarding the statistical models in the related-key setting, since the key schedule is linear, we can directly choose the key difference α for $\widehat{\boldsymbol{K}}$. Here, we set $\alpha = 5$ and obtain the key classes

$$\begin{aligned}
\mathcal{K}_\alpha(c_0 = -2 \cdot 2^{-10}) &= \{0, 7\}, \\
\mathcal{K}_\alpha(c_1 = 0) &= \{1, 3, 4, 6\}, \\
\mathcal{K}_\alpha(c_2 = +2 \cdot 2^{-10}) &= \{2, 5\}.
\end{aligned}$$

Similar to the attack in the direct attack setting, we choose different values of N, compute $\mathcal{T}_2$, and then check whether the right key lies in $\mathcal{K}_\alpha(c_i)$ with $i = \delta(\mathcal{T}_2)$ in each experiment. Repeating this experiment 2000 times, the experimental error probabilities can be obtained. The theoretical values of error probabilities can be computed using Theorem 2.

In the multiple related-key setting, we choose $t = 3$ differences for $\widehat{K}$ that form a dual basis for $\mathcal{B}$. Since the key schedule is linear, it is equivalent to choose $\alpha_0 = 1$, $\alpha_1 = 2$, and $\alpha_2 = 4$ for $\widehat{\boldsymbol{K}}$. We mounted the three basic related-key attacks, and we obtained the total error probability $P_e = 1 - \prod_{j=0}^{2}(1 - P_e^{\alpha_j})$ both experimentally and theoretically.

Comparisons of $\widehat{P_e}$ and P_e in the above three attack settings under KP sampling are illustrated in Fig. 3, where N is the size of data set encrypted under one key. From this figure, we can see that the experimental error probabilities confirm the theoretical models.

For the models under DKP sampling, we can conclude that the error probabilities obtained experimentally also correspond to those obtained theoretically, since the probability that plaintexts gathered under KP sampling have repeated values is

$$1 - \prod_{s=1}^{N-1}\left(1 - \frac{s}{2^{128}}\right) \leq 1 - \left(1 - \frac{N-1}{2^{128}}\right)^{N-1} \approx 1 - \exp\left\{\frac{N-1}{2^{128}}(N-1)\right\} \approx 0$$

for all N. Moreover, since B under DKP sampling is $(2^{128} - N)/(2^{128} - 1) \approx 1$ for $n = 128$, the theoretical values under KP and DKP sampling are almost the same, and the difference is at most 2^{-15} according to our experiments.

As a comparison, we also mounted attacks on the same 256-round keyed permutation in the above three settings using the statistical models proposed by Röck and Nyberg [RN13]. The results are illustrated in Fig. 4. One can see that the maximum absolute value of the theoretical error probability minus the experimental one of our models is 2.19 %, compared to 93.45 % for their models.

4.4.2 Experimental Verification of MLE-Based Models

To show the validity of MLE-based statistical models, we perform a comparison of the theoretical and empirical error probabilities of the key recovery attacks. For each attack setting, we implemented 2000 experiments on the 256-round keyed permutation of TinyJAMBU with the same aforementioned linear hull. A comparison of the results under KP sampling is given in Fig. 5. For the direct attack statistical models constructed under DKP sampling, we provide a comparison in Fig. 6. As for the related-key statistical models under DKP sampling, the comparison results are omitted since both the empirical and theoretical error probabilities are very close to those under KP sampling. The maximum absolute distance between theoretical error probabilities predicted by the MLE-based models and experimental ones is 1.9%. This implies that our new proposed methodology is an improvement over the models of Röck and Nyberg [RN13].

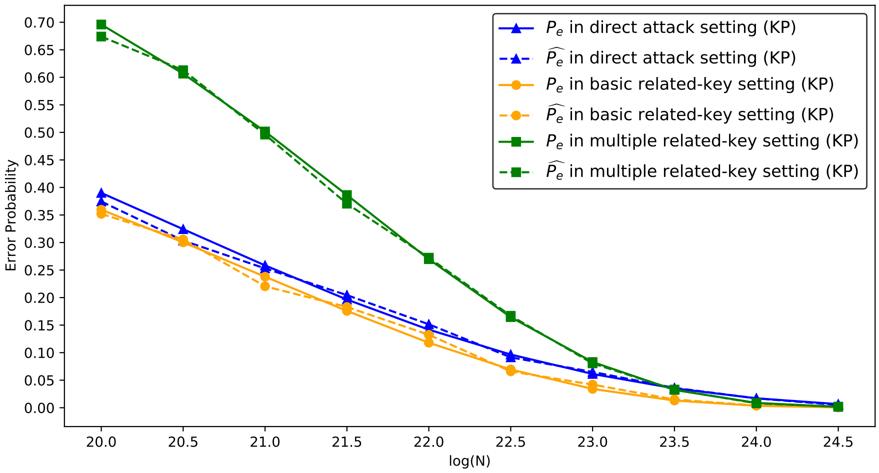


Figure 3: Comparison of error probabilities using the threshold-based models introduced in Sect. 3.1.

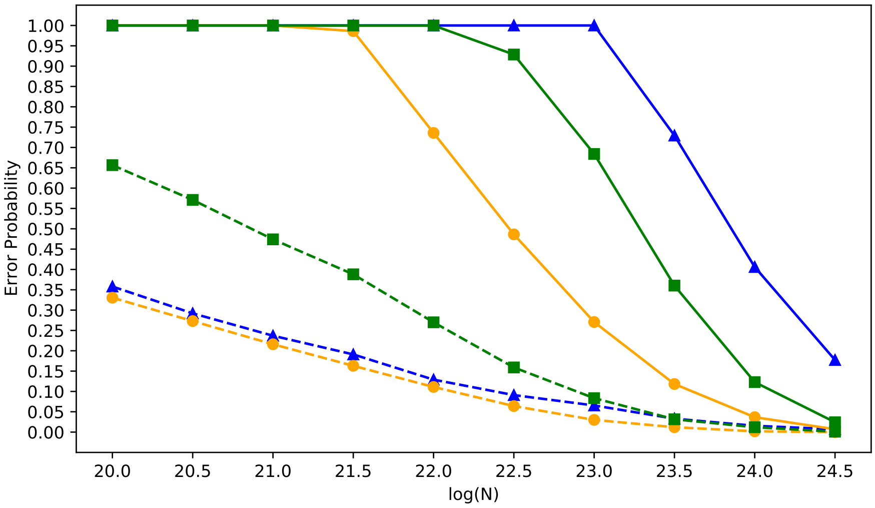


Figure 4: Comparison of error probabilities using models of Röck and Nyberg [RN13]. The legend of this graph is the same as Fig. 3.

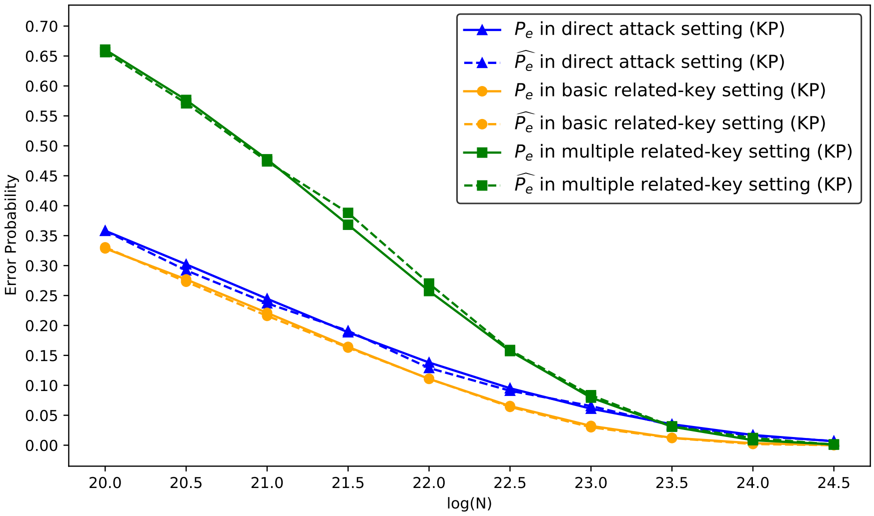


Figure 5: Comparison of error probabilities using the MLE-based models described in Sect. 3.2.

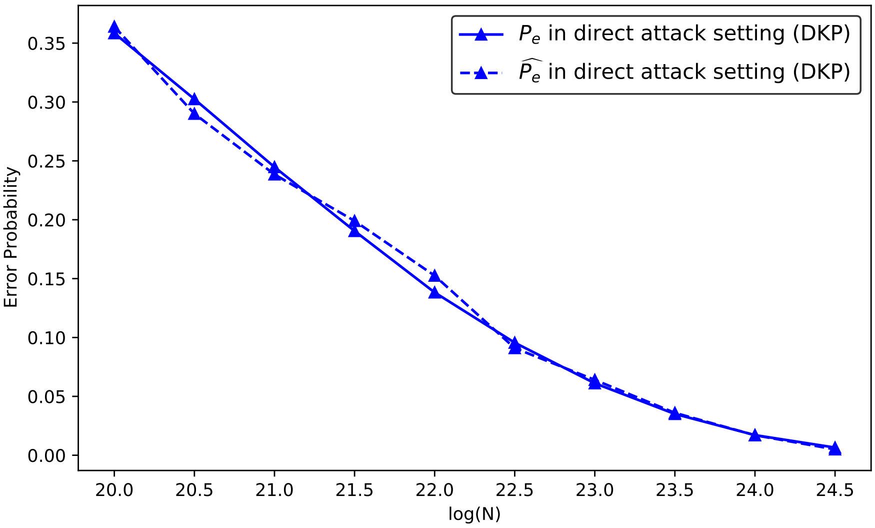


Figure 6: Comparison of error probabilities using the direct attack models under DKP sampling using the MLE-based models described in Sect. 3.2.

4.5 Partial Key Recovery Attacks on TinyJAMBU

In this subsection, we mount the first key recovery attacks on all key sizes using the statistical models introduced in Sect. 3, which recover partial key bits by using a linear hull on 384 (out of 384) and 387 (out of 640) rounds of the permutation $\mathcal{P}_b[l_1]$ used in the tag generation phase, for TinyJAMBU v1 and round-reduced TinyJAMBU v2 respectively. For simplicity, we only introduce the application of the threshold-based models in this subsection. Cryptanalysis results using the MLE-based models are summarized in Appendix F, where we also made some comparisons between these two kinds of statistical models.

First, we introduce attack results on TinyJAMBU v1 in the direct attack, basic related-key, and multiple related-key settings, and then we explain how a similar result can be applied to TinyJAMBU v2. To fully understand the security of TinyJAMBU, under each attack setting, we present two different types of key recovery results according to whether they are implemented in the single-user or the weak-key multi-user setting. The weak-key multi-user setting was used here because the number of tags collected per key is limited to 2^{47}. Hence, to obtain valid attacks with higher success probabilities, one has to consider multiple users and collect fewer than 2^{47} tags from each user. The keys owned by these users are different but must fulfill the same XOR relation in order to use our statistical models, as shown in Fig. 7. As a supplement of our security evaluation, we also show how these results behave when there is a single user available.

In Table 5, we present one result under each setting as an example. As shown in Table 5, given N tags in total, h key bits can be recovered with success probability $\Pr_{\text{success}}$ using our statistical models. All these tags are collected under 2^m, $2 \cdot 2^m$ and $15 \cdot 2^m$ distinct keys for the direct attack, basic related-key, and multiple related-key settings, respectively.

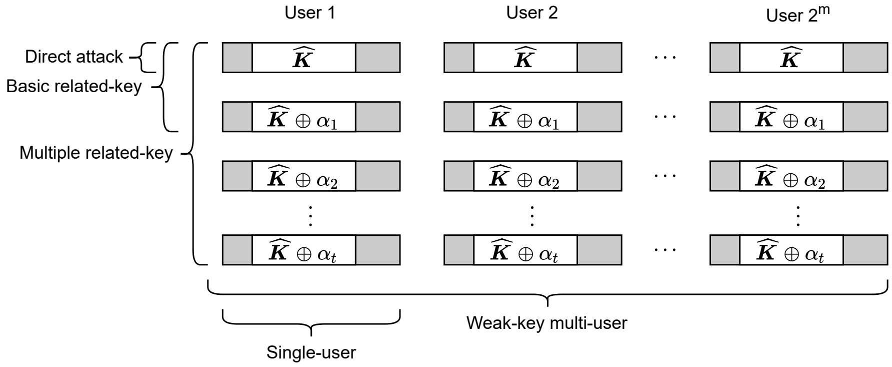


Figure 7: Examples of keys in the direct attack, basic related-key and multiple related-key settings, where each setting can be either single-user or weak-key multi-user. Note the additional assumption that the attacks are not done in the uniform standard model, as some bits of these keys must fulfill the same XOR relation. The white parts represent the XOR relation that each key owned by these users must fulfill; the grey parts are the other secret key bits which can be drawn uniformly at random but must be different within the same row.

We now give a detailed description of the results. First, we start by finding suitable linear hulls. Using Algorithm 1 with ($\lambda_0 = \texttt{0x8024C000}, \lambda_1 = \texttt{0x00220808}$) for 384 rounds, we obtain all trails comprising this hull. For each $\kappa \in \{128, 192, 256\}$, there are 850 trails in total. Their absolute correlations are computed with Lemma 3, while their signs are first determined by Corollary 1 and then multiplied by $(-1)^{\bigoplus_{r=0}^{R-1} \mathcal{X}_0^r}$ in order to eliminate $\mathcal{X}_0^r \cdot 1$ in the approximation. Details of the correlations are shown in Table 6.

Following Sect. 4.3, we can extract the key bits involved in each of the linear trails for every key length. Since every bit of the round key is equal to one bit of the master key,

Table 5: Cryptanalysis results on the full TinyJAMBU v1.

Setting	Type	N	2^m	h	$\Pr_{\text{success}}$	$\kappa = 128$	$\kappa \in \{192, 256\}$
Direct attack	weak-key multi-user	$2^{96.8}$	2^{50}	7.639	82.35 %		✓
	single-user	$2^{96.8}$	1	7.639	82.35 %	✓	✓
Basic related-key	weak-key multi-user	$2^{97.1}$	2^{50}	8.033	86.16 %		✓
	single-user	$2^{97.1}$	1	8.033	86.16 %	✓	✓
Multiple related-key	weak-key multi-user	$2^{102.31}$	2^{56}	14.063	84.85 %		✓
	single-user	$2^{102.31}$	1	14.063	84.85 %	✓	✓

Table 6: Correlations of 850 linear trails for 384-round $\mathcal{P}_b[l_1]$.

Correlation	$+2^{-42}$	$+2^{-43}$	$+2^{-44}$	$+2^{-45}$	$+2^{-46}$	$+2^{-47}$	$+2^{-48}$
Number of Trails	1	10	39	92	120	81	82
Correlation	-2^{-42}	-2^{-43}	-2^{-44}	-2^{-45}	-2^{-46}	-2^{-47}	-2^{-48}
Number of Trails	2	11	36	93	117	82	84

the alternative key $\widehat{K}$ is equal to $K \in \mathbb{F}_2^{\kappa}$. To accelerate the computation of $C(K)$ and find key classes $\mathcal{K}(c_i)$, as explained in Sect. 2.2, we directly traverse all $\widehat{\boldsymbol{K}} = \widehat{K}B^T$. For each key length, $\widehat{\boldsymbol{K}}$ is a 15-bit value $(ek_{14}, ek_{13}, \ldots, ek_0)$ that contains different bits of K, as shown in Appendix D.

By traversing all 2^{15} $\widehat{\boldsymbol{K}}$, we can get all possible values of the correlations $C(K)$ and their corresponding key classes $\mathcal{K}(c_i)$ in the direct attack setting. We find that there are 727 different values of $C(K)$ ranging from -77.875×2^{-42} to $+77.875 \times 2^{-42}$ and therefore 727 key classes. Moreover, for every key length, they share the same values of c_i and $\pi_i = 2^{-15}|\mathcal{K}(c_i)|$, however the key classes $\mathcal{K}(c_i)$ are different.

4.5.1 Key Recovery Attacks on TinyJAMBU v1 in the Direct Attack Setting

Using the statistical model introduced in Sect. 3.1.1, we can mount attacks in the direct attack setting using N known tags with correlations c_i.

The time complexities of these attacks correspond to the N encryption units to gather N tags. Using Theorem 1, the success probabilities can be determined after N is chosen. Different choices of N will lead to different success probabilities. The relation between the data complexities and their corresponding success probabilities is illustrated in Fig. 8. If the N tags need to be distinct (*i.e.,* DKP sampling), we can obtain a relation between the data complexities and their success probabilities that is almost the same as the relation in Fig. 8 since $B \approx 1$ in this case. Hence, all these attacks have approximately the same data and time complexities if N is fixed, no matter what data sampling is used. The key information obtained from these attacks is always 7.639 bits.

To ensure a high success probability, the attacks here require more than 2^{96} tags. However, the TinyJAMBU design document [WH19, WH21] assumes that at most 2^{47} tags can be obtained per key. To satisfy this restriction, we provide variants for all the attacks in the multi-user setting [Bir05] where tags are gathered under distinct keys. In this case, fewer than 2^{47} tags are required per key.

The related-key and multi-user settings are well-established in cryptographic literature,

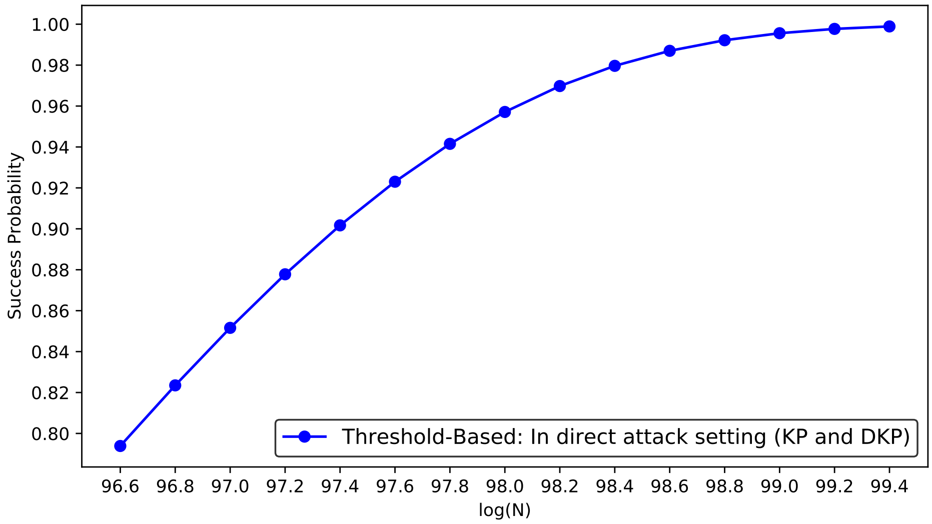


Figure 8: Success probabilities of direct attacks on TinyJAMBU using the threshold-based model.

and recent works such as Bose *et al.* [BHT18] also consider keys generated according to arbitrary, possibly correlated distributions. In lightweight applications, the practical relevance of the related-key and (weak-key) multi-user settings cannot be understated. A very large number of devices is often deployed (each with its own key). The keys may be weak keys; they may have known XOR relations [ABM$^+$12, § 5.2], and some bits of the keys may be set to specific values [VGE13, Sect. 7]. Such attacks should also be relevant to NIST: its recommendations for key generation allow keys with known XOR relations [BRD20, § 6.3] and it may occur that keys deployed in validated modules have reduced randomness [Yub19].

Note the weak-key multi-user setting gives the adversary more power than strictly required. We actually allow less powerful adversaries: the first key can be without any restrictions (it can be drawn uniformly at random from the entire key space), but it does impose restrictions on any subsequent keys that are derived from it (they are weak keys because the same XOR relation needs to hold on some bits of the key). This distinction can be relevant when the adversary does not own or control one device (to fully compromise it by extracting its key, and thereby determining the class of weak keys), but instead between the some bits of every key there is some fixed XOR relation that is unknown to the adversary.

Several reviewers correctly pointed out that this can be considered as some kind of related-key setting, but we wanted to maintain a distinction because we consider that the "direct attack" setting (including multi-user weak-key) is much more practically relevant than the "basic related-key" and "multiple related-key" settings (where we recovery more key bits).

Now we give detailed descriptions on our attacks. Note that the correlation of a linear hull is only determined by the key bits involved in this hull rather than the whole key. Denote key bits involved as K_I, and the other bits as K_O. Furthermore, the correlation is equal for all K_I with same $\widehat{\boldsymbol{K}}$. Here, $\widehat{\boldsymbol{K}}$ is the aforementioned 15-bit equivalent key.

The value of the statistic used in the statistical model introduced in Sect. 3.1.1 will not be affected by the value of K_O, and it also remains unchanged for different K_I if $\widehat{\boldsymbol{K}}$ is the same. The computation is done under tags generated with a fixed unknown value of K_I and K_O (*i.e.,* the whole key) when the attacks are implemented in the single-user setting. However, in the weak-key multi-user setting, it is evaluated through tags collected under several unknown values of K_I and K_O with same $\widehat{\boldsymbol{K}}$. The statistic computed with multiple keys has the same distribution as the statistic in the single-user setting if both statistics are computed under the same $\widehat{\boldsymbol{K}}$. At the same time, tags gathered under these keys are assumed to be independent with each other. Since we obtain them by encrypting randomly chosen messages under distinct nonces, this independence assumption can be assured by the randomness of the cipher itself. Hence, we can still use the statistical model introduced in Sect. 3.1.1 in the weak-key multi-user setting if the keys share the same $\widehat{\boldsymbol{K}}$.

Note that the security margin of TinyJAMBU in the multi-user setting will drop from 2^d into 2^{d-m} when 2^m different values of keys are used, and d is the number of bits of encryption security illustrated in Table 3. Under each key, $N/2^m$ tags are collected. Due to the designers' restriction of the data per key, $N/2^m \leq 2^{47}$. Meanwhile, since there are at most $2^{\kappa-|\widehat{\boldsymbol{K}}|}$ different keys sharing the same $\widehat{\boldsymbol{K}}$, m should satisfy that $2^m \leq 2^{\kappa-|\widehat{\boldsymbol{K}}|}$, where $|\widehat{\boldsymbol{K}}|$ denotes the size of $\widehat{\boldsymbol{K}}$. Thus, $N \leq 2^{47+m} \leq 2^{47+\kappa-|\widehat{\boldsymbol{K}}|}$. Combining all of the above, only when $N \leq \min\{2^{d-m}, 2^{47+\kappa-|\widehat{\boldsymbol{K}}|}\}$, attacks in the weak-key multi-user setting can be considered to be valid ones.

In the attacks on TinyJAMBU, $|\widehat{\boldsymbol{K}}| = 15$ for all $\kappa \in \{128, 192, 256\}$. Given $N = 2^{96.8}$ tags, key information can be derived with $\mathrm{Pr}_{\mathrm{success}} = 82.35\,\%$ when 2^{50} distinct keys are used. These kinds of attacks are only applicable to key lengths $\kappa \in \{192, 256\}$ since $N > \min\{2^{112-50}, 2^{128-15+47}\} = 2^{62}$ when $\kappa = 128$.

The same results can be applied to all key lengths in the direct attack setting if tags are collected from a single user. Note that TinyJAMBU adopts an unusual definition of the nonce-respecting setting. As stated in its design documents [WH19, WH21]: "the associated data is part of the nonce in TinyJAMBU, *i.e.*,, the combination (nonce $\|$ associated data) is the effective nonce of the cipher." Therefore, although the nonce in TinyJAMBU is only 96 bits, the effective nonce can be longer, so that it is technically possible to specify more than 2^{96} values without repetition. However, such results in the single-user setting do not threaten the security of TinyJAMBU since they need more tags per key than allowed by the designers.

4.5.2 Key Recovery Attacks on TinyJAMBU v1 in the Basic and Multiple Related-Key Settings

In the basic related-key setting, the key difference α must be chosen carefully since it influences $\mathrm{Pr}_{\mathrm{success}}$ even when N is fixed. To evaluate effects under different α, the notation *achieved* key information $\widehat{h}$ of [RN13] is adopted here, which is the information of the recovered key bits h multiplied by the success probability $\mathrm{Pr}_{\mathrm{success}}$.

After independently collecting two sets of tags under $\widehat{\boldsymbol{K}}$ and $\widehat{\boldsymbol{K}} \oplus \alpha$ with each set containing N_α tags, we can get $\mathrm{Pr}_{\mathrm{success}}$ using Theorem 2 for both KP and DKP sampling. The relation between $\widehat{h}$ and N_α is described in Fig. 9, which holds for all κ, but with different α, and does not depend on whether KP or DKP sampling is used. The data and time complexities of these related-key attacks are $N = 2N_\alpha$ tags and N encryption units, respectively. The details of the key differences chosen for each N_α can be found in Appendix E.

In the weak-key multi-user setting, we gather these two sets of N_α tags generated under related keys from 2^m users. For each user, the values of K_O used in related keys are chosen independently. Such attacks are valid only when $N \leq \min\{2^{d-m}, 2^{\kappa-|\widehat{\boldsymbol{K}}|+48}\}$. As such, they are applicable for all key lengths except $\kappa = 128$. For instance, if $N_\alpha = 2^{96.1}$ (*i.e.,* $N = 2^{97.1}$), 8.033 bits of key information can be recovered with $\mathrm{Pr}_{\mathrm{success}} = 86.16\,\%$ for both samplings where $m = 50$.

The same amount of key information can be obtained for all key lengths with same complexities, as well as the same $\mathrm{Pr}_{\mathrm{success}}$, in the single-user setting.

The key information recovered can be improved to 14.063 bits when multiple key

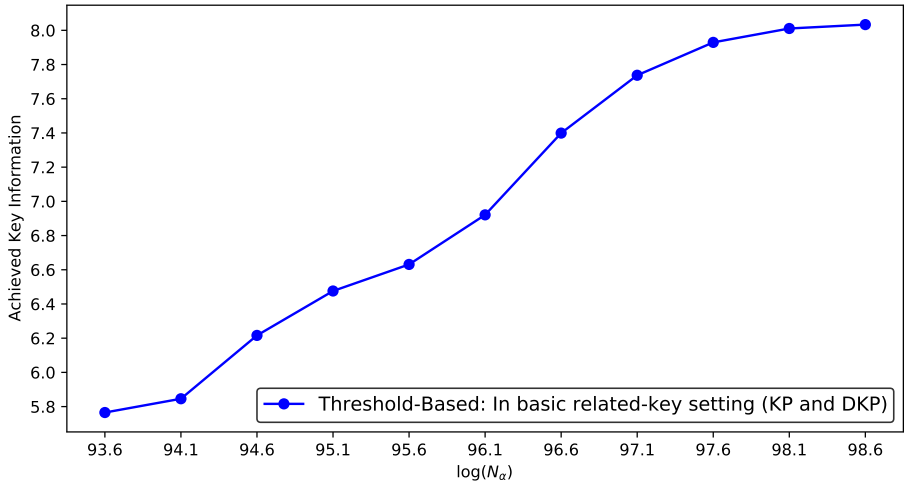


Figure 9: The achieved key information of the basic related-key attacks on TinyJAMBU using the threshold-based model.

differences are adopted simultaneously. In such attacks, we choose 15 differences $\alpha_i = (0,\ldots,0,1,0\ldots,0) \in \mathbb{F}_2^{15}$ for $\widehat{\boldsymbol{K}}$, where only the i-th bit is active in α_i. Under each α_i, we performed a basic related-key attack after independently obtaining two sets of N_α tags. Then, the total data complexities of these attacks are $N = 30N_\alpha$ and time complexities are N encryption units. The relation between $\Pr_{\text{success}}$ and N_α is illustrated in Fig. 10, which is approximately the same for any key length and for both KP and DKP sampling.

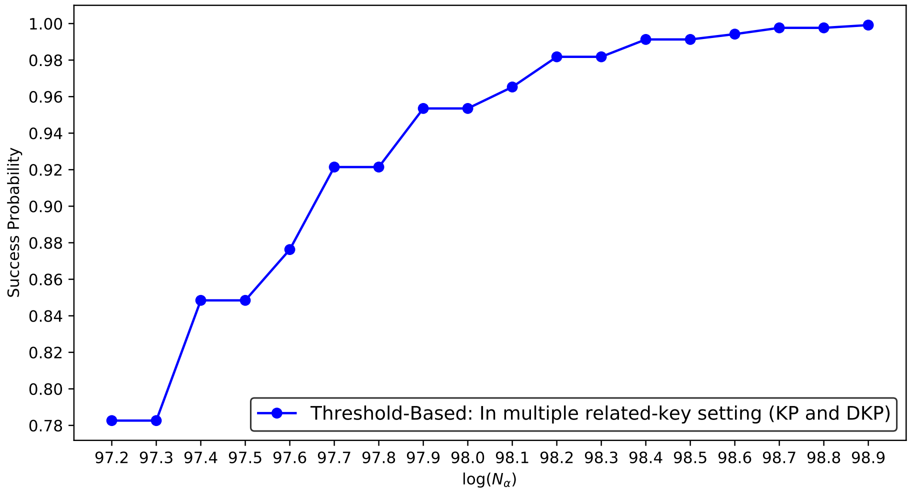


Figure 10: Success probabilities of multiple related-key attacks on TinyJAMBU using threshold-based model.

In this multiple related-key setting, $N \le \min\{2^{d-m}, 2^{\kappa-|\widehat{\boldsymbol{K}}|+48}\}$, leading to valid attacks in the weak-key multi-user setting for key lengths $\kappa \in \{192, 256\}$. If we choose $N_\alpha = 2^{97.4}$ and $N \approx 2^{102.31}$, 14.063 bits of key information can be recovered with $\Pr_{\text{success}} = 84.85\,\%$ for $\kappa \in \{192, 256\}$ for both KP and DKP sampling. In the single-user setting, we can recover 14.063 bits of key information on all three variants of TinyJAMBU v1 with the same complexities and success probabilities.

4.5.3 Key Recovery Attacks on Round-Reduced TinyJAMBU v2

Benefiting from the keyed permutation $\mathcal{P}_b$, we can extend our 384-round linear hull into a 387-round one with $\lambda'_0 = \texttt{0x8024C000} = \lambda_0$ and $\lambda'_1 = \texttt{0x00044101} = \lambda_1 \ggg 3$ without any extra cost. The reason is that there are no active AND gates among these three rounds, therefore they do not contain extra key bits in the linear trails. Hence, the linear hull on 387 rounds is actually the same as the 384-round one except that they contain different bits of the tags that do not influence the above results. Therefore, all cryptanalysis results proposed for TinyJAMBU v1 can also be applied to the 387-round TinyJAMBU v2.

5 Conclusion and Future Work

We revisited Röck and Nyberg's extension of Matsui's Algorithm 1 to linear hulls, introducing new statistical models under all settings considered by Röck and Nyberg. Our models are highly accurate: the absolute error between the theoretical and experimental probabilities is 2.19 % (threshold-based) or 1.9 % (MLE-based), compared to 93.45 % for Röck and Nyberg. Improvements on the accuracy between our model and theirs are mainly due to the new methodology of deducing the relation between success probabilities and data complexities. We obtain cryptanalysis results on TinyJAMBU, which is one of the ten finalists of the currently ongoing NIST LWC Standardization project. Our results are under the nonce-respecting setting on the full TinyJAMBU v1 and round-reduced TinyJAMBU v2. To maintain high success probabilities, the number of tags gathered per key is higher than 2^{96}. However, the designers restrict the data per key to at most 2^{47} tags. We overcame this by implementing the attacks under the weak-key multi-user setting where tags are collected under several distinct keys with a fixed XOR relation instead of being uniformly chosen. This weak-key multi-user setting is highly relevant for lightweight applications where often many small devices are deployed, each with their own key and with some bits of the key set to specific values. We also provided results in the single-user setting under the assumption that there is no restriction on the number of tags per key. Nevertheless, cryptanalysis results in this case do not violate the designers' security claims. Benefiting from the unusual nonce-respecting setting adopted by TinyJAMBU, more than 2^{96} tags can be specified by putting part of the nonce inside the associated data. Unfortunately our attacks are only partial key recovery: an exhaustive search on the remaining key bits would exceed the claimed 112, 168, and 224 bits of security. Although our results do not threaten the practical security of TinyJAMBU, they are the first cryptanalysis results in the nonce-respecting setting covering all key lengths of full TinyJAMBU v1 and the first one on TinyJAMBU v2. For future work, we suggest to further apply our statistical models on other cryptographic algorithms, and to investigate whether repeating the attack with different masks or different key relations may recover more bits of information about the key. In particular, we leave the application of our new model to PRESENT to future work, and note that it may be necessary to rely on Assumption 1 for PRESENT as there is a very significant amount of trail clustering, rather than considering all trails as we do in this paper for TinyJAMBU. Lastly, we recall that our results do not violate the security claims made by the designers.

Acknowledgments

The authors would like to thank the anonymous reviewers whose comments, together with shepherding by Tomer Ashur, greatly improved this paper. This work was supported in part by the National Natural Science Foundation of China (Grant No. 62032014, Grant No. 62002201, Grant No. 62002204), the National Key Research and Development Program of China (Grant No. 2018YFA0704702, Grant No. 2021YFA1000600), the Major Basic

Research Project of Natural Science Foundation of Shandong Province, China (Grant No. ZR202010220025), and the Program of Qilu Young Scholars (Grant No. 61580082063088) of Shandong University.

References

[ABM+12] Wim Aerts, Eli Biham, Dieter De Moitié, Elke De Mulder, Orr Dunkelman, Sebastiaan Indesteege, Nathan Keller, Bart Preneel, Guy A. E. Vandenbosch, and Ingrid Verbauwhede. A practical attack on KeeLoq. *J. Cryptol.*, 25(1):136–157, 2012.

[APSD20] Tomer Ashur, Raluca Posteuca, Danilo Sijacic, and Stef D'haeseleer. Generalized Matsui algorithm 1 with application for the full DES. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 2020*, volume 12238 of *LNCS*, pages 448–467. Springer, 2020.

[AR16] Tomer Ashur and Vincent Rijmen. On linear hulls and trails. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *INDOCRYPT 2016*, volume 10095 of *LNCS*, pages 269–286, 2016.

[BAK98] Eli Biham, Ross J. Anderson, and Lars R. Knudsen. Serpent: A new block cipher proposal. In Serge Vaudenay, editor, *FSE 1998*, volume 1372 of *LNCS*, pages 222–238. Springer, 1998.

[BBR+13] Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen, and Jingyuan Zhao. Key difference invariant bias in block ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *LNCS*, pages 357–376. Springer, 2013.

[BHT18] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: multi-user security, faster key derivation, and better bounds. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 468–499. Springer, 2018.

[Bir05] Alex Biryukov. Some thoughts on time-memory-data tradeoffs. *IACR Cryptol. ePrint Arch.*, 2005:207, 2005. http://eprint.iacr.org/2005/207.

[BJ72] Egon Balas and Robert Jeroslow. Canonical cuts on the unit hypercube. *SIAM J. Appl. Math.*, 23(1):61–69, 1972.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, 2007.

[BLNW12] Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. Integral and multidimensional linear distinguishers with correlation zero. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *LNCS*, pages 244–261. Springer, 2012.

[BN17] Céline Blondeau and Kaisa Nyberg. Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. *Des. Codes Cryptogr.*, 82(1-2):319–349, 2017.

[BRD20] Elaine Barker, Allen Roginsky, and Richard Davis. Recommendation for cryptographic key generation. NIST SP 800-133 Revision 2, June 2020.

[BW12] Andrey Bogdanov and Meiqin Wang. Zero correlation linear cryptanalysis with reduced data complexity. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *LNCS*, pages 29–48. Springer, 2012.

[Cho10] Joo Yeon Cho. Linear cryptanalysis of reduced-round PRESENT. In Josef Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 302–317. Springer, 2010.

[CS11] Baudoin Collard and François-Xavier Standaert. Experimenting linear cryptanalysis. In Pascal Junod and Anne Canteaut, editors, *Advanced Linear Cryptanalysis of Block and Stream Ciphers*, chapter 7, pages 1–28. IOS Press, 2011.

[DGV94] Joan Daemen, René Govaerts, and Joos Vandewalle. Correlation matrices. In Bart Preneel, editor, *FSE 1994*, volume 1008 of *LNCS*, pages 275–285. Springer, 1994.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[DR07] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *J. Math. Cryptol.*, 1(3):221–242, 2007.

[FWG$^+$16] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. MILP-based automatic search algorithms for differential and linear trails for Speck. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 268–288. Springer, 2016.

[HN11] Miia Hermelin and Kaisa Nyberg. Linear cryptanalysis using multiple linear approximations. In Pascal Junod and Anne Canteaut, editors, *Advanced Linear Cryptanalysis of Block and Stream Ciphers*, pages 29–53. IOS Press, Amsterdam, The Netherlands, 2011.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 386–397. Springer, 1993.

[Mur12] Sean Murphy. The effectiveness of the linear hull effect. *J. Math. Cryptol.*, 6(2):137–147, 2012.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Inscrypt 2011*, volume 7537 of *LNCS*, pages 57–76. Springer, 2011.

[Nat21] National Institute of Standards and Technology. Lightweight cryptography: Finalists, 2021. https://csrc.nist.gov/Projects/lightweight-cryptography/finalists.

[NH07] Kaisa Nyberg and Risto M. Hakala. A key-recovery attack on SOBER-128. In Eli Biham, Helena Handschuh, Stefan Lucks, and Vincent Rijmen, editors, *Symmetric Cryptography, 07.01. - 12.01.2007*, volume 07021 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. http://drops.dagstuhl.de/opus/volltexte/2007/1018.

[Nyb94] Kaisa Nyberg. Linear approximation of block ciphers. In Alfredo De Santis, editor, *EUROCRYPT 1994*, volume 950 of *LNCS*, pages 439–444. Springer, 1994.

[RN13] Andrea Röck and Kaisa Nyberg. Generalization of Matsui's Algorithm 1 to linear hull for key-alternating block ciphers. *Des. Codes Cryptogr.*, 66(1-3):175–193, 2013.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[SHS+13] Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang. Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks. In Dongdai Lin, Shouhuai Xu, and Moti Yung, editors, *Inscrypt 2013*, volume 8567 of *LNCS*, pages 39–51. Springer, 2013.

[SHW+14a] 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 propertie. *IACR Cryptol. ePrint Arch.*, 2014:747, 2014. https://eprint.iacr.org/2014/747.

[SHW+14b] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 158–178. Springer, 2014.

[SSS+19] Danping Shi, Siwei Sun, Yu Sasaki, Chaoyun Li, and Lei Hu. Correlation of quadratic boolean functions: Cryptanalysis of all versions of full MORUS. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019*, volume 11693 of *LNCS*, pages 180–209. Springer, 2019.

[SSS+20] Dhiman Saha, Yu Sasaki, Danping Shi, Ferdinand Sibleyras, Siwei Sun, and Yingjie Zhang. On the security margin of TinyJAMBU with refined differential and linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2020(3):152–174, 2020.

[STSH20] Ling Song, Yi Tu, Danping Shi, and Lei Hu. Security analysis of Subterranean 2.0. *IACR Cryptol. ePrint Arch.*, 2020:1133, 2020. https://eprint.iacr.org/2020/1133.

[TSY+21] Wil Liam Teng, Md. Iftekhar Salam, Wei-Chuen Yau, Josef Pieprzyk, and Raphaël C.-W. Phan. Cube attacks on round-reduced TinyJAMBU. *IACR Cryptol. ePrint Arch.*, page 1164, 2021. https://eprint.iacr.org/2021/1164.pdf.

[VGE13] Roel Verdult, Flavio D. Garcia, and Barış Ege. Dismantling Megamos crypto: Wirelessly lockpicking a vehicle immobilizer. In Samuel T. King, editor, *USENIX 2013*, pages 703–718. USENIX Association, 2013. https://www.usenix.org/sites/default/files/sec15_supplement.pdf.

[WCC+16] Meiqin Wang, Tingting Cui, Huaifeng Chen, Ling Sun, Long Wen, and Andrey Bogdanov. Integrals go statistical: Cryptanalysis of full skipjack variants. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 399–415. Springer, 2016.

[WH15] Hongjun Wu and Tao Huang. JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU. Submission to CAESAR: Competition for Authenticated Encryption. Security, Applicability, and Robustness, 2015. http://competitions.cr.yp.to/round2/aesjambuv2.pdf.

[WH19] Hongjun Wu and Tao Huang. TinyJAMBU: a family of lightweight authenticated encryption algorithms. Submitted to the NIST Lightweight Cryptography (LWC) Standardization project, 2019. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf.

[WH21] Hongjun Wu and Tao Huang. TinyJAMBU: a family of lightweight authenticated encryption algorithms (version 2). Submitted to the NIST Lightweight Cryptography (LWC) Standardization project, 2021. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/TinyJAMBU-spec-round2.pdf.

[WWH+13] Shengbao Wu, Hongjun Wu, Tao Huang, Mingsheng Wang, and Wenling Wu. Leaked-state-forgery attack against the authenticated encryption algorithm ALE. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013*, volume 8269 of *LNCS*, pages 377–404. Springer, 2013.

[Yub19] Yubico. Security advisory YSA-2019-02 – reduced initial randomness on FIPS keys, June 2019. https://www.yubico.com/support/security-advisories/ysa-2019-02/.

A Explanation of the Property Described in Sect. 2.1

Theorem 3. *For an SPN cipher $\mathcal{E}$, let $G(x,k) : \mathbb{F}_2^n \times \mathbb{F}_2^s \to \mathbb{F}_2^n$ be the round function with $s < n$, and let k_0 be the initial round key that is XORed with part of the plaintext before the first round. Given an input $x \in \mathbb{F}_2^n$ and a round key $k \in \mathbb{F}_2^s$, G first consists of the unkeyed round function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, followed by the XOR of only part of the output with the round key k. The correlation of a linear trail for $\mathcal{E}$ satisfies the property that only the sign of the correlation is influenced by the key.*

Proof. Without loss of generality, assume that the highest s bits of $F(x)$ are XORed with the round key k, and we denote them as $F_1(x)$. The other $(n-s)$ bits are denoted as $F_2(x)$. In other words, $G(x,k) = (F_1(x) \oplus k)||F_2(x)$. Given a linear trail $(\Gamma, \Lambda = \Lambda_1||\Lambda_2)$ of round function G, the correlation of this trail can be represented as

$$\begin{aligned} C = \sum_x (-1)^{\Gamma \cdot x \oplus \Lambda \cdot G(x,k)} &= \sum_x (-1)^{\Gamma \cdot x \oplus \Lambda_1 \cdot (F_1(x) \oplus k) \oplus \Lambda_2 \cdot F_2(x)} \\ &= \sum_x (-1)^{\Gamma \cdot x \oplus \Lambda_1 \cdot F_1(x) \oplus \Lambda_1 \cdot k \oplus \Lambda_2 \cdot F_2(x)} \\ &= \sum_x (-1)^{\Gamma \cdot x \oplus \Lambda \cdot F(x) \oplus \Lambda_1 \cdot k} \\ &= (-1)^{\Lambda_1 \cdot k} \sum_x (-1)^{\Gamma \cdot x \oplus \Lambda \cdot F(x)}. \end{aligned}$$

Hence, only the sign of the correlation is influenced by key. □

B Proof of Corollary 1

Proof. We use the same transformations provided by Song *et al.* [STSH20] when proving Lemma 3.

(1) When n is odd, the following transformation is used:

$$\begin{aligned} y_{2j-1} &= x_{2j-1}, && 1 \leq j \leq t, \\ y_{2j} &= x_{2j} \oplus x_{2(j+1)}, && 0 \leq j \leq t-2, \\ y_{2t-2} &= x_{2t-2}. \end{aligned}$$

Then we can transform f into

$$g(y_0, \ldots, y_{2t-1}) = y_0 \& y_1 \oplus y_2 \& y_3 \oplus \cdots \oplus y_{2(t-1)} \& y_{2t-1} \oplus a'_0 y_0 \oplus \cdots \oplus a'_{2t-1} y_{2t-1}$$

satisfying $\mathrm{Cor}(f) = \mathrm{Cor}(g)$. The coefficients of y_i in g are

$$a'_{2i} = \oplus_{j=0}^{i} a_{2j},\ a'_{2i+1} = a_{2i+1},\ 0 \leq i \leq t-1.$$

Since the sign of $\mathrm{Cor}(x_0 \& x_1 \oplus a_0 x_0 \oplus a_1 x_1)$ is $(-1)^{a_0 a_1}$, the sign of $\mathrm{Cor}(g)$ is

$$\mathrm{Sign}(g) = \prod_{i=0}^{t-1} (-1)^{a'_{2i} a'_{2i+1}} = \prod_{i=0}^{t-1} (-1)^{\left(\bigoplus_{j=0}^{i} a_{2j}\right) a_{2i+1}}.$$

Therefore, we have $\mathrm{Sign}(f) = \mathrm{Sign}(g)$.

(2) When n is even, another transformation can be used:

$$\begin{aligned} y_{2j-1} &= x_{2j-1}, && 1 \leq j \leq t, \\ y_{2j} &= x_{2j} \oplus x_{2(j+1)}, && 0 \leq j \leq t-1, \\ y_{2t} &= x_{2t}. \end{aligned}$$

Then f can be transformed into

$$g(y_0, \ldots, y_{2t}) = y_0 \& y_1 \oplus y_2 \& y_3 \oplus \cdots \oplus y_{2(t-1)} \& y_{2t-1} \oplus a'_0 y_0 \oplus \cdots \oplus a'_{2t} y_{2t}$$

with coefficients

$$a'_{2i} = \bigoplus_{j=0}^{i} a_{2j},\ 0 \leq i \leq t;\ a'_{2i+1} = a_{2i+1},\ 0 \leq i \leq t-1.$$

Since y_{2t} does not appear in any of these AND gates in g, we have $\mathrm{Cor}(g) = 0$ if $a'_{2t} = \bigoplus_{j=0}^{t} a_{2j} = 1$. If $a'_{2t} = 0$, we have

$$\mathrm{Sign}(f) = \mathrm{Sign}(g) = \prod_{i=0}^{t-1} (-1)^{a'_{2i} a'_{2i+1}} = \prod_{i=0}^{t-1} (-1)^{\left(\bigoplus_{j=0}^{i} a_{2j}\right) a_{2i+1}}. \qquad \square$$

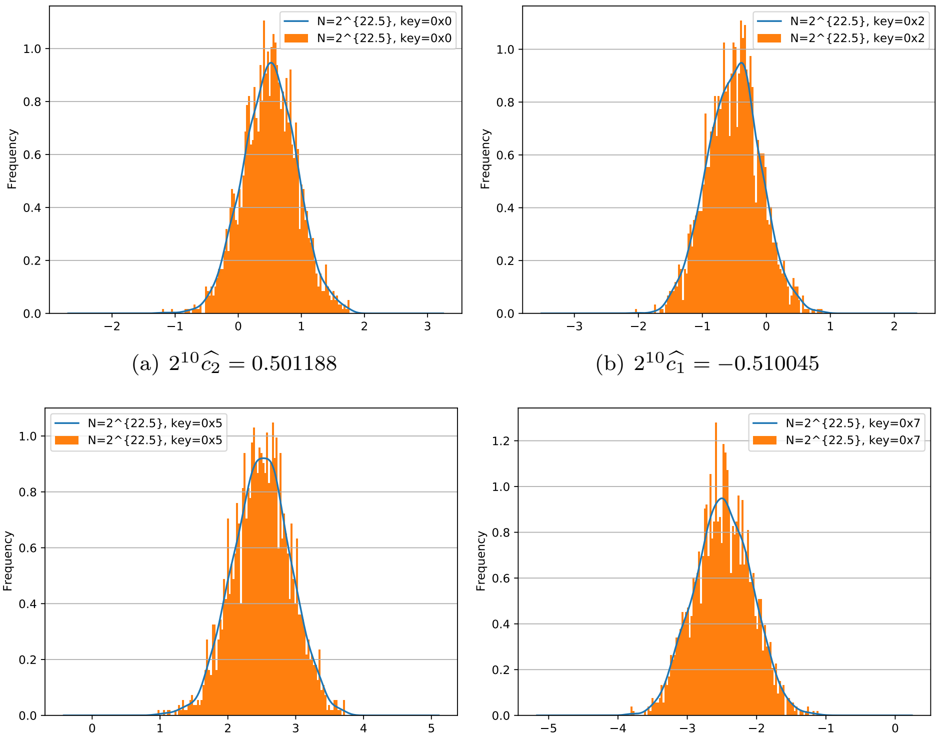
C Experimental Verification of the Assumption Adopted in our Search Algorithm

In our searching algorithm introduced in Sect. 4.2, we made the assumption that "each output of the AND gate affected by a key bit can be regarded as a fresh new bit". Here, we use the same round-reduced cipher adopted in Sect. 4.4, the 256-round keyed permutation of TinyJAMBU, to verify this assumption. Specifically, we check whether the theoretical correlation of the 256-round linear hull evaluated under different keys using our search algorithm fulfills corresponding empirical correlations.

Theoretical correlations of the linear hull have been introduced in Sect. 4.4, which are:

$$\mathcal{K}(c_0 = -2.5 \cdot 2^{-10}) = \{7\}, \mathcal{K}(c_1 = -0.5 \cdot 2^{-10}) = \{2, 3, 6\},$$
$$\mathcal{K}(c_2 = +0.5 \cdot 2^{-10}) = \{0, 1, 4\}, \mathcal{K}(c_3 = +2.5 \cdot 2^{-10}) = \{5\}.$$

In each experiment, we randomly choose N inputs and obtain its outputs under a fixed key. After repeating this experiment 2000 times, we can obtain the empirical correlations $\widehat{c_i}$. To make it clear, we only take the case when $N = 2^{22.5}$ for an illustration in Fig. 11. Since all $\widehat{c_i}$ is approximately equal to c_i for $i \in \{0, 1, 2, 3\}$, we can conclude that the assumption used in the search algorithm is reasonable.



(a) $2^{10}\widehat{c_2} = 0.501188$

(b) $2^{10}\widehat{c_1} = -0.510045$

(c) $2^{10}\widehat{c_3} = 2.496263$

(d) $2^{10}\widehat{c_0} = -2.487718$

Figure 11: Experimental verification of the assumption adopted in our search algorithm. Here, we only present the case when $N = 2^{22.5}$ for an illustration.

D Equivalent Key $\widehat{K}$ for 384-Round Hull

An equivalent key for the 384-round hull is given in Table 7.

E Key Differences Used in Basic Related-Key Attacks on TinyJAMBU

In Table 8, we provide the key differences adopted in the basic related-key attacks for data complexities N_α.

Table 7: Equivalent key $\widehat{\boldsymbol{K}} = (ek_{14}, ek_{13}, \ldots, ek_0)$ for 384-round hull.

	index j of k_j composing ek_i		
	$\kappa = 128$	$\kappa = 192$	$\kappa = 256$
ek_{14}	0, 1, 7, 11, 14, 17, 21, 24, 30, 41, 42, 48, 51, 52, 58, 61, 75, 81, 82, 84, 85, 89, 92, 95, 98, 108, 112, 121, 122, 125, 126	3, 14, 15, 20, 21, 25, 28, 34, 44 48, 57, 58, 64, 68, 71, 75, 78, 79, 81, 82, 85, 88, 94, 95, 106, 108, 112, 125, 126, 129, 131, 132, 139, 142, 145, 169, 172, 179, 180, 186, 189	0, 4, 7, 11, 17, 24, 30, 42, 44, 48, 67, 75, 78, 79, 81, 82, 95, 125, 126, 129, 132, 142, 149, 169, 172, 179, 180, 186, 189, 195, 206, 207, 212, 213, 217, 220, 226, 236, 240, 249, 250
ek_{13}	5, 89	7, 88, 155	5, 170, 217
ek_{12}	20	10, 25, 165, 180	24, 155, 199
ek_{11}	24, 71	25, 180	27, 42
ek_{10}	27, 42	33, 48	42
ek_9	35	48	148
ek_8	36	51, 152	152, 243
ek_7	37, 52, 74, 89	54, 155, 165	155, 202, 246
ek_6	42	63	163
ek_5	52, 89	69, 170, 180	164
ek_4	71, 115	91, 106	165, 180, 202, 217
ek_3	74, 118	106	180, 217
ek_2	97, 112	148	225, 240
ek_1	112	163	240
ek_0	127	164	255

Table 8: Key differences used in basic related-key attacks on TinyJAMBU.

$\log_2(N_\alpha)$	$\kappa = 128$	$\kappa = 192$	$\kappa = 256$
93.6	0x4010	0x4100	0x4100
94.1, 94.6, 95.1	0x4012,0x4016	0x4300,0x4700	0x4102,0x4106
95.6	0x609A,0x609B, 0x609E,0x609F	0x5320,0x5360, 0x5720,0x5760	0x6192,0x6193, 0x6196,0x6197
96.1	0x7A0A,0x7A0B, 0x7A0E,0x7A0F	0x63A6,0x63E6, 0x67A6,0x67E6	0x73C2,0x73C3, 0x73C6,0x73C7
96.6, 97.1, 97.6 98.1, 98.6	0x4B0A,0x4B0B, 0x4B0E,0x4B0F, 0x590A,0x590B, 0x590F,0x590E	0x6383,0x6385, 0x63C3,0x63C5, 0x6783,0x6785, 0x67C3,0x67C5	0x51E2,0x51E3, 0x51E6,0x51E7, 0x53A2,0x53A3, 0x53A6,0x53A7

F Comparison between Threshold-Based and MLE-Based Statistical Models

To better understand the difference between the threshold-based and MLE-based models, we compare the theoretical error probabilities predicted by these two models using the same linear hull on the round-reduced cipher. Since the theoretical values under DKP sampling are almost the same as those under KP sampling in both models, we only show the comparison under KP sampling in Fig. 12. As the figure shows, the MLE-based models have slightly higher success probabilities compared to the threshold-based models when error probabilities are higher than 20%. A possible reason is that the prior probability of each key class is considered in the decision making process of MLE-based models, but not for the threshold-based models. However, taking prior probabilities into account seems to be less important when the data complexity increases. Moreover, attacks usually involve error probabilities less than 20%. Considering this, the threshold-based and MLE-based models will have similar results, as shown in Fig. 12. The absolute distance between the theoretical values of the two models in this case is at most 1.15%.

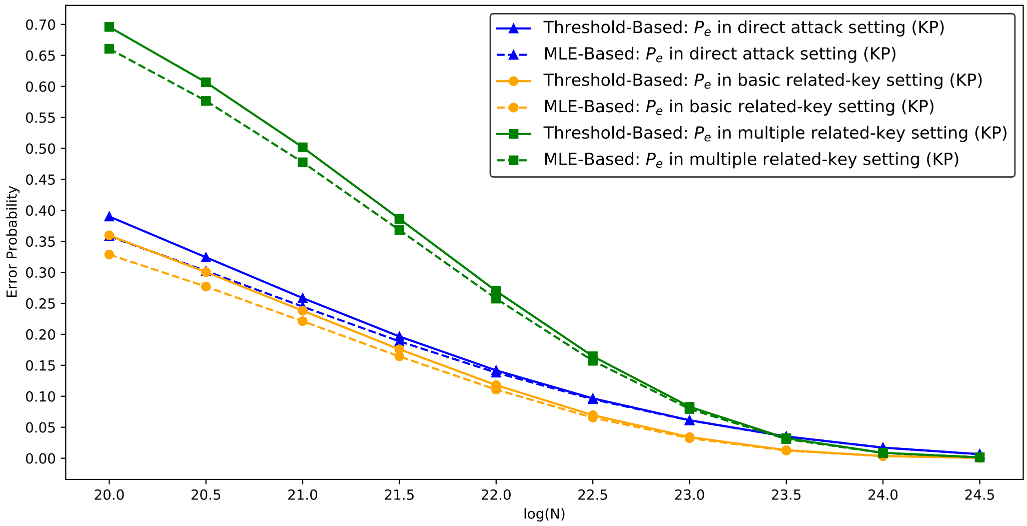


Figure 12: Comparison of error probabilities predicted by the threshold-based model and MLE-based model.

From Sect. 3.2, one can see that the evaluation of error probabilities P_{ij} requires computing and then intersecting all intervals obtained from inequalities $ML(i) > ML(t)$ for any $t \neq i$. Compared to the evaluation procedure in the threshold-based models, this requires more computations and therefore a longer running time. We have applied these MLE-based models on TinyJAMBU under the three attack settings, and confirmed that the MLE-based models indeed have longer running times to evaluate the error probabilities.

A detailed comparison of the running times for the threshold-based models and MLE-based models is shown in Table 9. The running time to evaluate the success probability depends on the data complexity N; in the table we show the longest running time. A total of 2^{15} key differences are traversed in the basic related-key setting, since we want to find those leading to the highest achieved key information. All experiments were performed on a server with an AMD EPYC 7302 16-core processor.

The success probabilities under the MLE-based models are illustrated in Figs. 13, 14, and 15, respectively. For comparison, we also include the success probabilities of the threshold-based models in these figures. In the direct attack setting (Fig. 13), the

Table 9: Comparison of the running times of the threshold-based and MLE-based models.

Setting	Threshold-Based Model	MLE-Based Model
Direct attack	$\leq$ 10 seconds (1 thread)	40 minutes (64 threads)
Basic related-key	one hour (64 threads)	two days (64 threads)
Multiple related-key	$\leq$ 1 minute (1 thread)	$\leq$ 4 minutes (1 thread)

MLE-based models provide slight improvements of the success probabilities. For example, the success probability increased by about 2% when $N = 2^{96.8}$. On the other hand, for the basic related-key setting (Fig. 14), the achieved key information of these two models is almost the same. However, under the multiple related-key setting (Fig. 15), these two models behave differently as expected. According to our theoretical comparisons on the reduced cipher in Fig. 12, the MLE-based model should provide a slightly higher success probability than the threshold-based model. Nevertheless, the results in Fig. 15 show some minor discrepancies that are within the range of rounding errors.

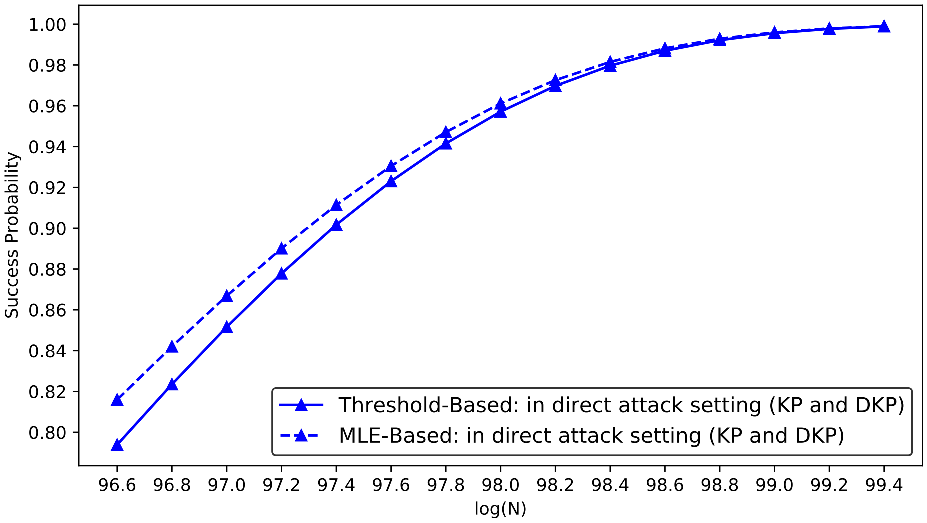


Figure 13: Comparison between the MLE-based model and the threshold-based model of the success probabilities of direct attacks on TinyJAMBU.

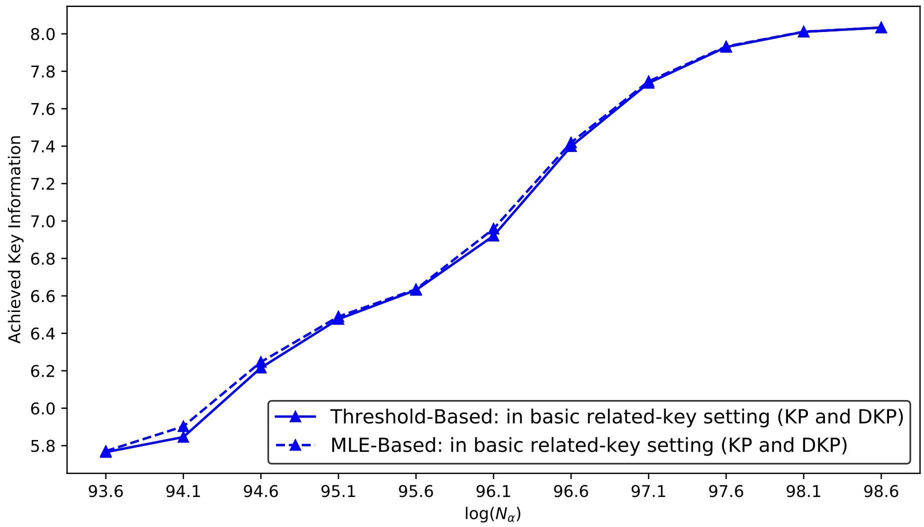


Figure 14: Comparison between the MLE-based model and the threshold-based model of the achieved key information of the basic related-key attacks on TinyJAMBU.

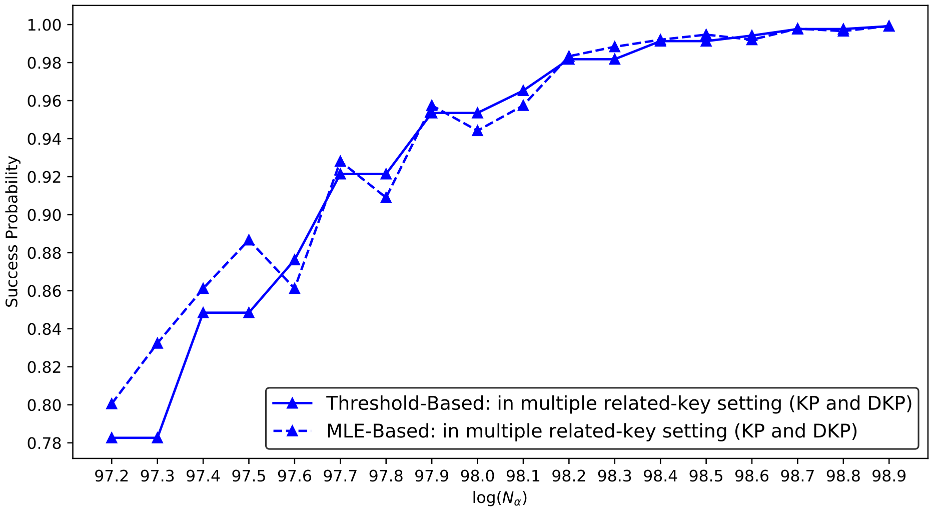


Figure 15: Comparison between the MLE-based model and the threshold-based model of the success probabilities of multiple related-key attacks on TinyJAMBU.