Probabilistic Related-Key Statistical Saturation Cryptanalysis

Muzhou Li[1,2], Nicky Mouha[3], Ling Sun[1,2,4,5], 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
[5] State Key Laboratory of Cryptology, P.O.Box 5159, Beijing, 100878, China
muzhouli@mail.sdu.edu.cn, nicky@mouha.be, lingsun@sdu.edu.cn, mqwang@sdu.edu.cn

Abstract. The related-key statistical saturation (RKSS) attack is a cryptanalysis method proposed by Li *et al.* at FSE 2019. It can be seen as the extension of previous statistical saturation attacks under the related-key setting. The attack takes advantage of a set of plaintexts with some bits fixed, while the other bits take all possible values, and considers the relation between the value distributions of a part of the ciphertext bits generated under related keys. Usually, RKSS distinguishers exploit the property that the value distribution stays invariant under the modification of the key. However, this property can only be deterministically verified if the plaintexts cover all possible values of a selection of bits. In this paper, we propose the *probabilistic RKSS cryptanalysis* which avoids iterating over all non-fixed plaintext bits by applying a statistical method on top of the original RKSS distinguisher. Compared to the RKSS attack, this newly proposed attack has a significantly lower data complexity and has the potential of attacking more rounds. As an illustration, for reduced-round Piccolo, we obtain the best key recovery attacks (considering both pre- and post-whitening keys) on both versions in terms of the number of rounds. Note that these attacks do not threaten the full-round security of Piccolo.

Keywords: Related-Key Statistical Saturation · Piccolo · Statistic

1 Introduction

Integral cryptanalysis is a cryptanalytic method for symmetric-key ciphers. First proposed by Daemen *et al.* as a dedicated attack on the Square cipher [10], the technique was later generalized by Knudsen and Wagner as the integral attack [21]. The integral distinguisher used in such an attack exploits the propagation of well-chosen sets of plaintexts through the cipher. In practice, a part of the plaintext bits is often fixed to some constant while all possible values are taken for the other bits, and the evolution of the variable bits in the cipher

state is tracked. To reduce its data complexity, the statistical integral attack [36] was proposed by Wang *et al.* at FSE 2016. It avoids iterating over all non-fixed plaintext bits by applying a statistical technique on top of the original integral attack. In [14], Dobraunig *et al.* introduced a related-tweak square attack on KIASU-BC that extends the single-key attack by one round.

The statistical saturation attack [8] was proposed by Collard and Standaert. It uses the same set of plaintexts as integral distinguishers, however, it tracks the evolution of a non-uniform value distribution of the ciphertext. At FSE 2019, Li *et al.* introduced the related-key statistical saturation (RKSS) attack [22] for key-alternating ciphers [11]. It also takes advantage of a set of plaintexts with some bits fixed while the others take all possible values, however, it considers the relation between the value distributions of a part of the ciphertext bits generated under related keys. RKSS distinguishers exploit the property that a part of the ciphertexts keeps their value distribution invariant under the modification of the key. However, this property can only be deterministically verified if the plaintexts cover all possible values of a selection of bits.

In this paper, we revisit the RKSS cryptanalysis and propose a new method that can address such limitations with the help of a statistical model. This new method is referred to as *probabilistic RKSS cryptanalysis*. Compared to the original method, the data complexity here can be much smaller with only a small decrease in success probability. An intuitive comparison of these two methods is shown by their applications on Piccolo [31].

We now provide a detailed overview of the contributions of this paper.

Probabilistic RKSS Cryptanalysis. In Sect. 3, we will introduce the probabilistic RKSS cryptanalysis method, which avoids iterating over all non-fixed plaintext bits. In this way, we require less data than the original RKSS method, but the same value distribution property of the original RKSS will not strictly hold.

However, we can still distinguish between a right key guess and a wrong key guess by choosing an appropriate statistic that considers the different distributions in these two cases. First, we recall the value distribution property that the original RKSS method relies on. Let s be the number of plaintext bits that take all possible values while the other bits are fixed. For all these 2^s plaintexts, we encrypt them under related-key pairs and obtain two sets of ciphertexts. Denote t as the number of ciphertext bits whose value distribution is considered here. For any t-bit value of this part, we have the same number of occurrences in these two sets of ciphertexts. When less than 2^s plaintexts are available, the occurrences of each t-bit value may not be the same anymore, but their differences may be small if enough plaintexts are given. Hence, the statistic is constructed by summing all 2^t squared differences of the number of occurrences counted under these two related keys. With the help of Stuart-Maxwell tests for marginal homogeneity [26,33], we can prove that such a statistic follows a ^{2}-distribution with different parameters for right and wrong key guesses. The validity of this statistical model is also confirmed experimentally on a toy cipher.

With this statistical model, the data complexity of the RKSS attack can be reduced from 2^s to

$$N = 2^s \quad (2^s \quad 1) \frac{q_{\alpha_1}^{(2^t \quad 1)}}{q_{1 \quad \alpha_0}^{(2^t \quad 1)}};$$

where $q_{\alpha_1}^{(2^t \quad 1)}$ and $q_{1 \quad \alpha_0}^{(2^t \quad 1)}$ represent the quantiles of the central ^{2}-distribution with each having a degree of freedom equal to $2^t \quad 1$. Meanwhile, $_0$ (resp. $_1$) is the probability of rejecting the right key (resp. of accepting a wrong key). This new attack has a success probability of $\Pr_s = 1 \quad _0$. Note that the trade-off between the success probability $\Pr_s$ and the data complexity N allows the attack to cover more rounds than the original RKSS method.

Improved Key Recovery Attacks on Round-Reduced Piccolo with both Whitenings. Piccolo [31] is a 64-bit ultra-lightweight key-alternating block cipher designed by Shibutani *et al.* at CHES 2011. It is suitable for constrained environments such as RFID tags and sensor nodes. The cipher supports 80-bit and 128-bit keys, denoted as Piccolo-80 and Piccolo-128, respectively.

Since its proposal, many key recovery attacks have been introduced such as (conditional) linear attacks [2], (multidimensional) zero-correlation linear attacks [1,17], meet-in-the-middle attacks [18,23,24,35], and (related-key) impossible differential attacks [4,27,34]. In addition, there are some other results such as biclique attacks [19,37]. However, there is a consensus in the literature that biclique attacks are not a threat to a cipher, as they require an exhaustive search over a reduced number of rounds of the cipher.

From all these attacks, we find that the security resistance of Piccolo is different depending on whether the pre/post-whitening key layers are included or not. Specifically, when both whitenings are considered, the best-known attack on Piccolo-80 is on 8 rounds [2], not including biclique attacks. Meanwhile, the best result on Piccolo-128 with both whitenings is a biclique attack [19]. When including none or only one of these two whitening key layers, the best key recovery attack can cover 14 rounds for Piccolo-80 [23,35] and 18 rounds for Piccolo-128 [23]. This confirms that key whitening may strengthen the security of Piccolo. Thus, we are motivated to investigate its real impact on security, and try to narrow the gap between the cryptanalytic results in the above two cases.

In Sect. 4, we mount several key recovery attacks on both variants of Piccolo using the probabilistic RKSS method. To show the effectiveness of this new method, we also propose attacks using the RKSS method in Sect. 4. All these results are presented in Table 1. Compared to previous results, they are the best key recovery attacks containing both pre- and post-whitening keys on Piccolo.

From Table 1, for 16-round Piccolo-128, we can see that the probabilistic RKSS method needs only 3.44% of the number of plaintexts required in the RKSS attack with only a little decrease in its success probability from 100% to 99%. Moreover, the probabilistic RKSS method can cover one more round than the RKSS method. As for Piccolo-80, the data complexity used in the new method is only 10% of that required in the RKSS method where its success probability is 99%.

Table 1. Comparison of attacks on Piccolo containing both pre- and post-whitening key layers. Time complexities are evaluated in encryption units, while memory costs are evaluated in bits, and $\#k$ denotes the number of different keys used.

Cipher	Attacks	Rounds	Data	Time	Memory	$\#k$	Ref.
Piccolo-128	RKSS	16	2^{49}	$2^{114.19}$	2^{38}	2	Sect. 4.2
	Prob. RKSS	**16**	$\mathbf{2^{44.14}}$	$\mathbf{2^{114.18}}$	$\mathbf{2^{38}}$	**2**	**Sect. 4.2**
	Prob. RKSS	**17**	$\mathbf{2^{60.14}}$	$\mathbf{2^{115.44}}$	$\mathbf{2^{67.14}}$	**2**	**Sect. 4.3**
Piccolo-80	Cond. Linear	8	2^{54}	2^{54}	N.A.	1	[2]
	RKSS	10	2^{41}	$2^{74.49}$	$2^{33.81}$	2	Sect. 4.1
	Prob. RKSS	**10**	$\mathbf{2^{37.68}}$	$\mathbf{2^{74.48}}$	$\mathbf{2^{33.81}}$	**2**	**Sect. 4.1**

2 Preliminaries

Key-alternating ciphers form a significant subset of modern block ciphers, which was introduced by Daemen and Rijmen in [11]. Many block ciphers, including almost all Substitution-Permutation Networks (SPNs) and some Feistel ciphers, belong to this subset [12].

De nition 1. ***(Key-Alternating Block Cipher [11])*** *Given an r-round iterative block cipher E, let k_i represent its i-th round key with $1 \quad i \quad r$. If k_i is XORed into the state at the end of the i-th round and there exists a subkey k_0 introduced by XORing with the plaintext before the first round, the block cipher E is a key-alternating block cipher.*

The related-key statistical saturation (RKSS) attack [22] is a new cryptanalytic method for key-alternating ciphers proposed by Li *et al.* at FSE 2019. This method can be regarded as an extension of statistical saturation attack [8] in the related-key setting. As pointed out in [22], this method is also applicable for tweak/tweakey-alternating ciphers, where related-tweak/tweakey are taken into consideration, since tweak/tweakey can be seen as a kind of key. For simplicity, all of these are referred to as RKSS attacks in this paper. The main idea of the RKSS attack is that we fix a part of the plaintext bits and take all possible values for the other bits, and then consider the relation between the value distributions of a part of the ciphertext bits under related-key pairs $(z; z^0 = z \quad z)$, where z is a fixed value for all possible values of the key z. To obtain such RKSS distinguishers, Li *et al.* [22] introduced a conditional equivalent property between the KDIB distinguisher [7] and the RKSS distinguisher.

The KDIB technique [7] is another method proposed for key-alternating ciphers, which can be seen as an extension of linear cryptanalysis [25]. Linear cryptanalysis typically uses a linear trail. Denote $= (\ _0; \ _1; \quad ; \ _r)$ as an r-round linear trail, where $_i \ _1$ is the input mask of round i $(1 \quad i \quad r)$ and $_i$ is the output mask. Its bias $"_\theta$ is related to the unknown key z. For key-alternating ciphers, only the sign of $"_\theta$ is affected by z. A linear hull $(u; w)$ consists of all trails satisfying $u = \ _0$ and $w = \ _r$ [29], whose bias is evaluated by summing

all biases of these trails under the same key z. Hence, the bias of a linear hull can be invariant if it is evaluated under related-key pairs (z, z') fulfilling some specific key difference Δz. This is the fact that the KDIB distinguisher exploits.

To explain the conditional equivalent property between KDIB and RKSS distinguishers, we adopt the same notation used in [22]. Denote $\mathbb{F}_2^n$ as the space of n-dimensional binary vectors over $\mathbb{F}_2 = \{0, 1\}$. Let $H : \mathbb{F}_2^n \times \mathbb{F}_2^k \to \mathbb{F}_2^n$ be the target block cipher with block size n and key size k. The n-bit input of H is split into two parts (x, y), where x is the part fixed and y is the part taking all possible values. Note that these two parts can be composed of arbitrary input bits. Similarly, the output of H is also divided into two parts $(H_1(x, y, z), H_2(x, y, z))$ and only the value distribution of $H_1(x, y, z)$ is considered. Thus, we have

$$H : \mathbb{F}_2^r \times \mathbb{F}_2^s \times \mathbb{F}_2^k \to \mathbb{F}_2^t \times \mathbb{F}_2^u;\ H(x, y, z) = (H_1(x, y, z), H_2(x, y, z)).$$

Fixing x to a constant value I and only focusing on the H_1 part of the output, we can obtain the function $T_I : \mathbb{F}_2^s \times \mathbb{F}_2^k \to \mathbb{F}_2^t;\ T_I(y, z) = H_1(I, y, z)$. In an RKSS distinguisher, we will consider the relation between the value distributions of $T_I(y, z)$ and $T_I(y, z')$ after encrypting all possible values of y.

Given the above notation, the conditional equivalent property between the KDIB and the RKSS distinguishers can be described in Theorem 1 and illustrated in Fig. 1. Once the KDIB distinguisher is found, an RKSS distinguisher covering the same rounds can also be obtained using Theorem 1.

Theorem 1. (Conditional Equivalent Property [22]) Let (α, β) be the linear hull of the target block cipher with $\alpha = (\alpha_{in}, 0)$ and $\beta = (\beta_{out}, 0)$, where $\alpha_{in} \in \mathbb{F}_2^r$ and $\beta_{out} \in \mathbb{F}_2^t \setminus \{0\}$. Given a fixed Δz, if for all possible mask pairs $(\alpha_{in}, \beta_{out})$, the bias is invariant under related-key pairs $(z, z' = z \oplus \Delta z)$, $T_I(y, z)$ will have the same value distribution as $T_I(y, z')$ when y takes all possible values and vice versa. In other words, for any $c \in \mathbb{F}_2^t$, we have $\#\{y \in \mathbb{F}_2^s \mid T_I(y, z) = c\} = \#\{y \in \mathbb{F}_2^s \mid T_I(y, z') = c\}$. Note that this holds for any $I \in \mathbb{F}_2^r$.

Note that in Theorem 1, the restriction to masks of the form $(\alpha_{in}, 0)$ and $(\beta_{out}, 0)$, where the last bits are fixed to zeros, is solely for the simplicity of notation. As pointed out in [22], the positions of the zero bits do not affect the applicability of this property.

Fig. 1. Equivalence between KDIB and RKSS distinguishers [22].

From Theorem 1, we can see that the RKSS distinguisher exploits the property that the value distribution of some ciphertext bits stays invariant under

the modification of the key. When mounting the RKSS key recovery attack, we have to traverse all possible values of y under a fixed value of x, and ask for ciphertexts under z and z'. Thus, we can observe whether $T_l(y;z)$ has the same value distribution with $T_l(y;z')$ after guessing the corresponding key bits. If so, the guessed key bits will be taken as the right key bits. Otherwise, they will be discarded. According to Theorem 1, for a right key guess, $T_l(y;z)$ always has the same value distribution with $T_l(y;z')$. Hence, the probability of rejecting the right key $_0$ is zero. As for the probability of accepting a wrong key $_1$, they proved that $\log_2(_1)$ is no more than $(2^t - 1 - t)2^{s+1} - 2^{s(2^t-1)=2}$, which is extremely small. For instance, when Liet al. [22] attacked 10-round QARMA-64 [3] with $s = 56$ and $t = 4$, it was found that $\log_2(_1) \approx 2.7 \cdot 10^{126}$, which implies that $_1 \approx 0$.

3 Probabilistic Related-Key Statistical Saturation Attack

3.1 Introducing a Statistical Model into RKSS Cryptanalysis

In this subsection, we adopt the notation introduced in Sect. 2. Let q_j (resp. q'_j) denote the probability that $T_l(y;z) = j$ (resp. $T_l(y;z') = j$) when iterating over all possible values of $y \in F_2^s$. Thus, $\sum_{j=0}^{2^t-1} q_j = 1$ and $\sum_{j=0}^{2^t-1} q'_j = 1$. Note that in the RKSS attack, q_j and q'_j can take various values for different wrong key candidates z and z', while $q_j = q'_j$ holds for any j for a right key guess. Let $^2(l;\ )$ represent the noncentral ^{2}-distribution with degree of freedom l and noncentrality parameter $\ $. For an RKSS distinguisher, we can obtain Lemma 1 for both wrong and right key guesses, according to Stuart-Maxwell [26, 33] tests for marginal homogeneity.

Lemma 1. When 2^s is sufficiently large, for a wrong key guess, the statistic

$$= \sum_{j=0}^{2^t-1} \frac{2^s q_j \quad 2^s q'_j{}^2}{2^s q_j + 2^s q'_j}$$

approximately follows $^2(2^t - 1; 0)$. For the right key guess, the statistic $= 0$.

Proof. For the right key guess, $= 0$ holds according to Theorem 1. While for a wrong key guess, we can prove it as follows.

Denote $_{j_1;j_2}$ as the probability that $T_l(y;z) = j_1$ and $T_l(y;z') = j_2$ simultaneously holds for all possible 2^s values of y. Thus, $q_j = \sum_{j_2=0}^{2^t-1} {}_{j;j_2}$ and $q'_j = \sum_{j_1=0}^{2^t-1} {}_{j_1;j}$. Given an RKSS distinguisher, we want to test whether $q_j = q'_j$ holds for any $0 \quad j \quad 2^t - 1$ after obtaining 2^s samples. Thus, it is equivalent to testing for marginal homogeneity of the frequency table described in Table 2.

To test for marginal homogeneity, we can use the Stuart-Maxwell statistic $W = d^T M^{-1} d$,[6] under the null hypothesis $H_0 : 2^s q_j = 2^s q'_j; 8\ 0 \quad j \quad 2^t - 1$.

[6] Note that the Stuart-Maxwell test relies on the assumption that all paired data $(T_l(y;z); T_l(y;z'))$ evaluated under the same sample y are pairwise in-

Table 2. Frequency table used to prove Lemma 1.

$T_l(y;z')$ \ $T_l(y;z)$	0	1	j_2	2^t-1	Total
0	$2^s{}_{0;0}$	$2^s{}_{0;1}$	$2^s{}_{0;j_2}$	$2^s{}_{0;2^t-1}$	$2^s q_0$
1	$2^s{}_{1;0}$	$2^s{}_{1;1}$	$2^s{}_{1;j_2}$	$2^s{}_{1;2^t-1}$	$2^s q_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
j_1	$2^s{}_{j_1;0}$	$2^s{}_{j_1;1}$	$2^s{}_{j_1;j_2}$	$2^s{}_{j_1;2^t-1}$	$2^s q_{j_1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2^t-1	$2^s{}_{2^t-1;0}$	$2^s{}_{2^t-1;1}$	$2^s{}_{2^t-1;j_2}$	$2^s{}_{2^t-1;2^t-1}$	$2^s q_{2^t-1}$
Total	$2^s q'_0$	$2^s q'_1$	$2^s q'_{j_2}$	$2^s q'_{2^t-1}$	2^s

In the statistic W, d is a (2^t-1)-dimensional vector $(2^s q_1 - 2^s q'_1, \ldots, 2^s q_{2^t-1} - 2^s q'_{2^t-1})^T$. M is a $(2^t-1)\times(2^t-1)$ matrix and its elements are

$$M_{i;i} = 2^s q_i + 2^s q'_i - 2 \cdot 2^s{}_{i;i}; \quad M_{i;j} = -2^s{}_{i;j} - 2^s{}_{j;i};$$

where $1 \le i \le 2^t-1$ and $1 \le j \le 2^t-1$. According to [26, 33], W approximately follows $\chi^2(2^t-1;0)$ when 2^s is sufficiently large.

Denote M^{-1} as the following $(2^t-1)\times(2^t-1)$ matrix

$$\begin{bmatrix} \frac{1}{2^s q_1 + 2^s q'_1} & & & \\ & \frac{1}{2^s q_2 + 2^s q'_2} & & \\ & & \ddots & \\ & & & \frac{1}{2^s q_{2^t-1} + 2^s q'_{2^t-1}} \end{bmatrix} + \frac{Y}{2^s q_0 + 2^s q'_0};$$

and Y is a $(2^t-1)\times(2^t-1)$ matrix where all entries are equal to one. Thus, $M M^{-1} = I + A$ where I is the identity matrix and A is a matrix where the element is

$$A_{i;j} = \frac{2^s{}_{i;0} + 2^s{}_{0;i}}{2^s q_0 + 2^s q'_0} - \frac{2^s{}_{i;j} + 2^s{}_{j;i}}{2^s q_i + 2^s q'_j};$$

where $1 \le i \le 2^t-1$ and $1 \le j \le 2^t-1$. For each i and j, $A_{i;j}$ can be approximated[7] by 0. Therefore, $\chi^2 = d^T M^{-1} d \approx d^T M^{-1} d$ approximately follows $\chi^2(2^t-1;0)$ since $M M^{-1} \approx I$. □

The only way to reduce the data complexity of an RKSS attack is to reduce the number of y that are chosen. However, the same value distribution property

dependent. That is, given any two different samples y_1 and y_2, the paired data $(T_l(y_1;z); T_l(y_1;z'))$ collected in the wrong-key case is independent of $(T_l(y_2;z); T_l(y_2;z'))$. This assumption has been verified experimentally. We refer to Appendix B for an illustration.

[7] We have experimentally verified this in Appendix B.

under a right key guess will not hold if we choose some random values for y. The advantage is that we can distinguish a right key guess from a wrong one by constructing a statistic with the information of similar frequencies of each possible output under related-key pairs (z, z'), if a considerable number of distinct values of plaintexts are reachable. This new kind of RKSS attack with reduced data complexity will be referred to as a probabilistic RKSS attack hereafter.

Assume that we have obtained two independent randomly chosen distinct plaintext sets S and S' with the same size N. All plaintexts share the same fixed I. For each $y \in S$ (resp. $y' \in S'$), we can get a t-bit value $T_I(y, z)$ (resp. $T_I(y', z')$) that is computed under z (resp. z'). Then we respectively add one to the counter $V[j_1]$ and $V'[j_2]$, where $j_1 = T_I(y, z)$ and $j_2 = T_I(y', z')$. After traversing all these N values of y and N values of y', we can construct an efficient distinguisher by investigating the distribution of the following statistic

$$C = \sum_{j=0}^{2^t-1} \frac{(V[j] - V'[j])^2}{2N - 2^t},$$

where $V[j] = \#\{y \in S \mid T_I(y, z) = j\}$ and $V'[j] = \#\{y' \in S' \mid T_I(y', z') = j\}$.

This statistic C considers different distributions determined by whether we are dealing with an actual cipher (right key guess) or a random permutation (wrong key guess). These two distributions of C are derived under Hypothesis 1. The validity of this hypothesis has been verified experimentally in Appendix B.

Hypothesis 1 For any $0 \le i \le 2^t - 1$, $0 \le j \le 2^t - 1$, we assume that $q_i q_j \approx (2^{-t})^2$, $q'_i q'_j \approx (2^{-t})^2$, and $q_i + q'_j \approx 2 \cdot 2^{-t}$ hold when 2^s is sufficiently large[8].

Proposition 1. Denote C_{random} as the statistic C for a wrong key guess and C_{cipher} as the statistic C for the right key guess. Under Hypothesis 1, for sufficiently large N, the statistic

$$\frac{2^s - 1}{2^s - N} C_{cipher} \approx \chi^2_{2^t - 1, 0};$$

while the statistic

$$C_{random} \approx \chi^2_{2^t - 1, 0}.$$

To prove this proposition, we have to recall the following lemma.

Lemma 2. (See [13]) Let $X = (X_1, X_2, \ldots, X_d)^T$ be a d-dimensional statistic vector that follows the multivariate normal distribution with expectation μ and covariance matrix Σ, where Σ is a symmetric matrix of rank $r \le d$. If $\Sigma^2 = \Sigma$ and $\mu = \Sigma$, we have $X^T X \approx \chi^2_{r, \mu^T \mu}$.

With Hypothesis 1 and Lemmas 1 and 2, we can prove Proposition 1 as follows.

[8] In our experimental verification, $s = 12$ and it is enough to ensure the validity of this hypothesis, as well as other assumptions used in this paper.

Proof. Recall that when mounting probabilistic RKSS attacks, the counters $V[T_l(y,z)]$ and $V'[T_l(y',z')]$ are generated by encrypting two independently chosen values y and y' under z and z'. Therefore, these two counters are independent of each other.

Since we choose distinct values of y (sampling without replacement), the statistic vector $(V[0], V[1], \ldots, V[2^t-1])$ follows a multivariate hypergeometric distribution with parameters $(K, 2^s, N)$ where $K = (Nq_0, Nq_1, \ldots, Nq_{2^t-1})$. Similarly, the vector $(V'[0], V'[1], \ldots, V'[2^t-1])$ also follows a multivariate hypergeometric distribution however the parameters are $(K', 2^s, N)$ where $K' = (Nq'_0, Nq'_1, \ldots, Nq'_{2^t-1})$. When N is sufficiently large, both hypergeometric distributions can be approximated into multivariate normal ones.

For any $0 \le j \le 2^t-1$, define $X_j = V[j] - V'[j]$. Then we have that $X = (X_0, X_1, \ldots, X_{2^t-1})$ also follows a multivariate normal distribution. Since expectation of X_j is $E(V[j] - V'[j]) = E(V[j]) - E(V'[j]) = Nq_j - Nq'_j$, the expectation of X can be obtained. The covariance between X_i and X_j can be computed by

$$\begin{aligned}
&\mathrm{Cov}(X_i, X_j) = E(X_i X_j) - E(X_i)E(X_j) = E((V[i] - V'[i])(V[j] - V'[j])) \\
&\quad (E(V[i]) - E(V'[i]))(E(V[j]) - E(V'[j])) \\
&= E(V[i]V[j]) + E(V'[i]V'[j]) - E(V[i]V'[j]) - E(V'[i]V[j]) \\
&E(V[i])E(V[j]) - E(V'[i])E(V'[j]) + E(V[i])E(V'[j]) + E(V'[i])E(V[j]) \\
&= \mathrm{Cov}(V[i], V[j]) + \mathrm{Cov}(V'[i], V'[j]) - \mathrm{Cov}(V[i], V'[j]) - \mathrm{Cov}(V'[i], V[j]) \\
&= \mathrm{Cov}(V[i], V[j]) + \mathrm{Cov}(V'[i], V'[j]),
\end{aligned}$$

where the last equality comes from the independence of the counters $V[T_l(y,z)]$ and $V'[T_l(y',z')]$.

Let $X_j = X_j / \sqrt{2N2^{-t}\frac{2^s-N}{2^s-1}}$. Then $X = X / \sqrt{2N2^{-t}\frac{2^s-N}{2^s-1}}$ also follows a multivariate normal distribution with expectation $= (\ _0, \ _1, \ldots, \ _{2^t-1})$ where

$$_j = E(X_j) = \sqrt{2N2^{-t}\frac{2^s-N}{2^s-1}} = (Nq_j - Nq'_j) = \sqrt{2N2^{-t}\frac{2^s-N}{2^s-1}};$$

and covariance matrix where

$$_{i,i} = \frac{q_i(1-q_i) + q'_i(1-q'_i)}{2^{-2t}}; \quad _{i,j} = \frac{q_iq_j - q'_iq'_j}{2^{-2t}}.$$

Due to Hypothesis 1, $_{i,i} \approx 1 \cdot 2^{-t}$ and $_{i,j} \approx 2^{-t}$. Notice that is symmetric and its rank is $2^t - 1$. It is easy to verify that $^2 =$ and $=$. According to Lemma 2, we can conclude that

$$\frac{2^s-1}{2^s-N}\sum_{j=0}^{2^t-1}\frac{(V[j]-V'[j])^2}{2N2^{-t}} \quad ^2(2^t-1; \ ) \text{ with } = \frac{2^s-1}{2^s-N}\sum_{j=0}^{2^t-1}\frac{Nq_j - Nq'^2_j}{2N2^{-t}}.$$

Under Hypothesis 1, in Lemma 1 can be approximated as

$$\sum_{j=0}^{2^t-1} \frac{2^s q_j \quad 2^s q_j'^{\,2}}{2 \; 2^s \; 2^t}$$

and then $\frac{2^s \; 1}{2^s \; N} \frac{N}{2^s}$. Thus, for a right key guess, = 0 since = 0. In other words,

$$\frac{2^s \quad 1}{2^s \quad N} C_{\text{cipher}} \quad {}^2 \; 2^t \quad 1; 0 \; :$$

While for a wrong key guess, $\frac{2^s}{N} \frac{2^s \; N}{2^s \; 1}$ ${}^2(2^t \quad 1; 0)$ according to Lemma 1. Thus, the distribution of C_{random} can be obtained with the characteristic functions of ^{2}-distributions.

For a noncentral ^{2}-distribution U ${}^2(l; \;)$, the characteristic function is

$$CF_U(it) = \frac{1}{(1 \quad 2it)^{l=2}} \exp \quad \frac{it}{1 \quad 2it}$$

with i being the imaginary unit. If is a random variable, we will denote the characteristic function as $CF_{U\,j}$ for clarity. Moreover, by the de nition of characteristic functions, for any a, CF_{aU} is the same as CF_U with ait substituted everywhere for it. Therefore,

$$CF_{C_{\text{random}} \quad j}(it) = \frac{\exp \quad \frac{\frac{2^s \quad N}{2^s \quad 1} it}{1 \quad 2\frac{2^s \quad N}{2^s \quad 1} it}}{(1 \quad 2\frac{2^s \quad N}{2^s \quad 1} it)^{(2^t \quad 1)=2}}; \quad CF \; (it) = \frac{1}{(1 \quad 2\frac{N}{2^s} \frac{2^s \quad 1}{2^s \quad N} it)^{(2^t \quad 1)=2}}:$$

Replacing it by $\frac{\frac{2^s \quad N}{2^s \quad 1} it}{1 \quad 2\frac{2^s \quad N}{2^s \quad 1} it}$ in CF , we can integrate out from $CF_{C_{\text{random}} \quad j}$. Thus,

$$\begin{aligned} CF_{C_{\text{random}} \quad}(it) &= \frac{1}{(1 \quad 2\frac{2^s \quad N}{2^s \quad 1} it)^{(2^t \quad 1)=2}} CF \; \left(\frac{\frac{2^s \quad N}{2^s \quad 1} it}{1 \quad 2\frac{2^s \quad N}{2^s \quad 1} it}\right) \\ &= \frac{1}{(1 \quad 2(\frac{2^s \quad N}{2^s \quad 1} + \frac{N}{2^s}) it)^{(2^t \quad 1)=2}} \quad \frac{1}{(1 \quad 2it)^{(2^t \quad 1)=2}}: \end{aligned}$$

In other words, C_{random} follows ${}^2(2^t \quad 1; 0)$. tu

To decide whether the obtained statistic C is computed from the cipher (a right key guess) or the random permutation (a wrong key guess), we have to perform a statistic test. In this test, we compare C to a threshold value . If C , we conclude that C is obtained from the cipher; otherwise, it is from a random permutation. The data complexity needed to perform the statistic test and the threshold value can be computed as follows, given error probabilities.

Corollary 1. Denote α_0 as the probability of rejecting the right key and α_1 as the probability of accepting a wrong key. Under the assumption of Proposition 1, the number of distinct plaintexts encrypted under a single key is

$$N = 2^s \quad (2^s \quad 1)\frac{q_1^{(2^t \quad 1)}}{q_1^{(2^t \quad 1)} \quad q_0^{(2^t \quad 1)}};$$

and the threshold value is $= q_1^{(2^t \quad 1)} = \frac{2^s \quad N}{2^s \quad 1} q_1^{(2^t \quad 1)} \quad q_0^{(2^t \quad 1)}$, where $q_1^{(2^t \quad 1)}$ and $q_1^{(2^t \quad 1)} \quad q_0^{(2^t \quad 1)}$ are the respective quantiles of $^2(2^t \quad 1; 0)$.

Proof. By the definition of α_0 and our statistic test, we have $\Pr\{C_{cipher} > g\} = \alpha_0$. Then

$$\Pr\left[\frac{2^s \quad 1}{2^s \quad N} C_{cipher} > \frac{2^s \quad 1}{2^s \quad N} \quad = \alpha_0\right].$$

By the definition of a quantile, we know that $\frac{2^s \quad 1}{2^s \quad N} = q_1^{(2^t \quad 1)} \quad q_0^{(2^t \quad 1)}$. Similarly, we can obtain $= q_1^{(2^t \quad 1)}$ due to $\Pr\{C_{random} < g\} = \alpha_1$. Hence, we see that

$$\frac{2^s \quad 1}{2^s \quad N} q_1^{(2^t \quad 1)} = q_1^{(2^t \quad 1)} \quad q_0^{(2^t \quad 1)}$$

holds by eliminating from the above two equations. In this case, N can be obtained. □

According to Corollary 1, we can see that the data encrypted under a single key in the probabilistic RKSS attack is less than 2^s, which is the data collected under a single key of the original RKSS attack. In other words, our newly proposed method needs less data than the original one. Meanwhile, the success probability of this attack is $\Pr_s = 1 \quad \alpha_0$. Note that such a trade-off between $\Pr_s$ and N can make it possible to mount attacks that cover more rounds than the original RKSS method. Further comparisons between these two methods are shown in Appendix C.

3.2 Experimental Verification of the Statistical Model

To verify the theoretical model, we implement a distinguishing attack on a mini version of an SPN cipher denoted as SmallSPN (a variant of Mini-AES [30]). [9]

SmallSPN is a 20-round key-alternating cipher with a block size of 16 bits. Its round function contains four operations, i.e., SB, SR, MC, and AK. Additionally, there is another AK operation before the first round. The 16-bit plaintext $P = P_0||P_1||P_2||P_3$ is arranged into a 2×2 matrix $\begin{matrix} P_0 & P_1 \\ P_2 & P_3 \end{matrix}$ and SB uses 4-bit S-box in QARMA-64 [3]. SR is the operation interchanging P_2 and P_3. The matrix

[9] SmallSPN has a structure that is similar to Mini-AES, but they have different number of rounds, S-box, linear matrix, and key schedule.

used in MC is $\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}$. Denote rk^i as the round key in the i-th round, $0 \quad i \quad 20$, and rk^i_j is the j-th nibble of rk^i where $0 \quad j \quad 3$. Each subkey rk^i will be XORed with the nibbles in AK operations, all of which are chosen uniformly at random.

Fig. 2. Experimental results related to the statistical model using SmallSPN

The 20-round RKSS distinguisher used here can be described as follows: when we x P_3 and iterate over all 2^{12} possible values of $P_0 jj P_1 jj P_2$, the value distributions of C_3 obtained under K and C^0_3 obtained under K^0 will be the same. K^0 and K only have non-zero di erences on rk^0_0, rk^0_1, rk^0_2, rk^1_1, rk^1_2, rk^1_3, rk^2_1, rk^2_3, rk^3_3, rk^{18}_1, rk^{19}_0, rk^{18}_3, rk^{20}_0, rk^{20}_1, and rk^{20}_2. Now we mount the probabilistic RKSS attack using the statistical model described in Proposition 1 where $s = 12$ and $t = 4$. Setting $_0 = 0:2$ and choosing di erent values for N, we can obtain $_1$ and according to Corollary 1. In each experiment, we independently and randomly collect two plaintext sets with size N, where all plaintexts share the same xed I, and query their ciphertexts generated with SmallSPN After computing the statistic C and comparing it with , we can decide whether we are facing the real cipher. By launching this experiment 1000 times, we can obtain the empirical error probability $\hat{}_0$. Similarly, if we generate these ciphertexts with random permutations, we can obtain the empirical error probability $\hat{}_1$ following the same procedure. Thereafter, we can compare these error probabilities with theoretical ones $_0$ and $_1$, which is illustrated in Fig. 2. From Fig. 2, we can see that the test results for error probabilities are in good accordance with those for the theoretical model. Thus, our statistical model is accurately constructed.

4 Improved Key Recovery Attacks on Piccolo Considering Pre- and Post-Whitening

At CHES 2011, Piccolo was proposed by Shibutani et al. [31] as a lightweight block cipher with a 64-bit block size. The key size can be either 80 or 128 bits, and we will denote these variants as Piccolo-80 and Piccolo-128, respectively. A brief introduction to Piccolo is presented in Appendix A.

In this section, we provide the best key recovery attacks on Piccolo (containing both pre- and post-whitening key layers) in terms of the number of rounds, compared to previous results. When no whitening keys or only either pre- or post-whitening is considered, the best attacks on Piccolo are meet-in-the-middle (MITM) attacks [23, 24]. However, according to [16, 31], whitening keys are essential to construct ciphers that are resistant to MITM attacks. Thus, to check the resistance of Piccolo against MITM attacks when both whitening keys are included, we had a private communication with the authors of [23, 24]. We both agree that MITM cannot attack 10-round Piccolo-80 and 16-round Piccolo-128 in this case since almost all key bits have to be guessed. Hence, to the best of our knowledge, our key recovery attacks are the best-known attacks on Piccolo.

4.1 Probabilistic RKSS Attack on 10-Round Piccolo-80

The first step to mount attacks is to find an RKSS distinguisher. As explained in Sect. 2, Li et al. [22] constructed a search algorithm for KDIB distinguishers, and then RKSS distinguishers covering the same rounds can be obtained using Theorem 1. To make our paper self-contained, we briefly recall the principle of this automatic search algorithm. For more details, we refer to [22].

Their search algorithm is based on STP,[10] which is a Boolean Satisfiability Problem (SAT) [9]/Satisfiability Modulo Theories (SMT) problem [5] solver. The application of STP as an automatic search tool for differential cryptanalysis was first suggested by Mouha and Preneel in [28]. It takes a set of equations as input and decides whether or not they have a valid solution. Therefore, when using STP to find KDIB distinguishers, we have to build some equations that describe the propagation properties of each operation. More specifically, for operations in the round function, the word-level mask propagation properties should be described; while for each operation in the key schedule, we have to describe its bit-level difference propagation property. Moreover, there are also some equations required to describe the relation between the masks and the key difference. Inserting all these equations into STP, we can obtain a KDIB distinguisher for a fixed number of rounds or conclude that no KDIB distinguishers exist.

Like other related-key attacks, the starting round of the distinguisher has an impact on the length of the distinguisher. Using this automatic search algorithm, we found an 8-round KDIB distinguisher with the pre-whitening key layer starting from the third round, which is illustrated in Fig. 11 of Appendix D. The key difference of this distinguisher is $k_4[1] =$ which can be any non-zero value

[10] http://stp.github.io/

in $\mathbb{F}_2^4$. Denote the 16-bit value X as $X = X[0]||X[1]||X[2]||X[3]$ with $X[i] \in \mathbb{F}_2^4$, and let $X[i,j]$ represents $X[i]||X[j]$. Combining the 8-round KDIB distinguisher with Theorem 1 leads to the following RKSS distinguisher.

Fig. 3. Probabilistic RKSS attack on 10-round Piccolo-80 with full whitening, where are active nibbles and are nibbles that we need to know in the key recovery procedure.

Corollary 2. With the notation of Fig. 3, for the 8-round Piccolo-80 including pre-whitening key layer, when we take all 2^{40} plaintexts with $P_0[0,2,3]||P_2[0,2,3]$ fixed, the value distribution of the 12-bit value $W_1[0,2,3] \oplus k_2[0,2,3]$ stays invariant under (K, K'), where K and K' only differ at $k_4[1]$.

Proof. By Theorem 1, $W_3[0,2,3]$ encrypted under K has the same value distribution as $W_3'[0,2,2]$ encrypted under K'. Since $k_2[0,2,3] = k_2'[0,2,3]$, we can conclude that $W_3[0,2,3] \oplus k_2[0,2,3]$ also has the same value distribution with $W_3'[0,2,3] \oplus k_2'[0,2,3]$. Therefore, we can avoid guessing $k_2[0,2,3]$ in key recovery attacks. □

Using this distinguisher, a probabilistic key recovery attack on 10-round Piccolo-80 can be carried out by adding two rounds and the post-whitening key layer at the end. Algorithm 1 and Fig. 3 show the details of this attack. As usual, we collect N plaintexts P with $P_0[0,2,3]$ and $P_2[0,2,3]$ fixed. For each plaintext, we can query its corresponding ciphertext. Since wk_2 and wk_3 have been guessed, we can compute x_1 and increase $V_1[x_1]$ by one. With a similar procedure, another counter V_1' can be obtained from another N plaintexts P' where $P_0'[0,2,3] = P_0[0,2,3]$ and $P_2'[0,2,3] = P_2[0,2,3]$. With another guess of k_0^R and k_1^L, we can obtain the counters V_2 and V_2' from V_1 and V_1', respectively. Using the statistical model proposed in Sect. 3, we can get the right key after checking its validity with two new plaintext-ciphertext pairs.

Suppose that one memory access to an array of size 2^{28} costs less than one encryption of 10-round Piccolo-80. Then, the time complexity of this key recovery

attack is at most $T = 2^{32}N \cdot 2 \cdot (1+1) + 2^{32} \cdot 2^{16} \cdot 2^{28} \cdot 2 \cdot (1/2) \cdot (1/10) + 2 \cdot 2^{80} \cdot {}_1$, where N can be computed using Corollary 1 after choosing the values of ${}_0$ and ${}_1$. Here, we set ${}_0 = 0.01$ and ${}_1 = 2^{-7.16}$. In this way, $N \approx 2^{36.68}$, $2^{11.92}$. Hence, the data complexity is $D = 2N \approx 2^{37.68}$ chosen plaintext-ciphertext pairs, while the time complexity is $T = 2^{74.48}$ 10-round encryptions. The memory requirements are $M = 2 \cdot 2^{28} \cdot 28 \approx 2^{33.81}$ bits needed for arrays.

To show the advantages of our newly proposed method, we also give the complexity of the original RKSS attack using the same distinguisher. Since we have to iterate over all possible values of $P_0[1] \| P_1 \| P_2[1] \| P_3$ in the original RKSS attack, the data complexity will be $D' = 2^{41}$ chosen plaintext-ciphertext pairs. The time complexity can be computed as before except that it is 2^{32} rather than $2^{80} \cdot {}_1$ and $N = 2^{40}$, which is $T' = 2^{74.49}$ times a 10-round encryption. The memory requirement is $M' = M$. As we can see, $D < D'$. More precisely, $D = 10\% \cdot D'$.

4.2 Probabilistic RKSS Attack on 16-Round Piccolo-128

In this subsection, we provide a probabilistic RKSS key recovery attack on 16-round Piccolo-128 containing both pre- and post-whitening layers. This attack is based on the 11-round RKSS distinguisher starting from the 14-th round described in Corollary 3.

Corollary 3. With the notation of Fig. 4, for the 11-round Piccolo-128, when we take all 2^{48} input values of 14-th round with $X_0[0,1] \| X_2[2,3]$ fixed, the value distribution of the 16-bit value $W_3 \oplus k_3$ stays invariant under (K, K'), where K and K' only differ at $k_0[2,3] = \{\alpha \cdot F_{2^8}^* \cup 0\}$.

The probabilistic RKSS attack on 16-round Piccolo-128 can be mounted by adding the pre-whitening key layer before the distinguisher and five rounds, as well as the post-whitening key layer at the end. The detailed key recovery procedure is illustrated in Fig. 4 and described in Algorithm 2. One thing we should mention here is that to get the same value distribution property, we have to encrypt two independent data sets with $X_0[0,1]$ and $X_2[2,3]$ fixed under related keys. Since $wk_1[2,3] = k_0[2,3]$ has a non-zero known difference α, we can obtain the same fixed $X_2[2,3]$ by setting $P'[2,3] = P[2,3] \oplus \alpha$.

Suppose that one memory access to an array of size 2^{32} costs less than one encryption of 16-round Piccolo-128. Then, the time complexity of this key recovery attack can be computed as $T = 2^{64}N \cdot 2 \cdot (1 + 4/16 + 1) + 2^{64} \cdot 2^{16} \cdot 2^{32} \cdot 2 \cdot (1/2) \cdot (1/16) + 2 \cdot 2^{128} \cdot {}_1$. By setting ${}_0 = 0.01$ and ${}_1 = 2^{-14.89}$, we can obtain $N \approx 2^{43.14}$ with $\approx 2^{15.97}$ according to Corollary 1. Thus, the data complexity is $D \approx 2^{44.14}$ chosen plaintext-ciphertext pairs, while the time complexity is $T \approx 2^{114.18}$ 16-round encryptions. The memory requirements are $M = 2 \cdot 2^{32} \cdot 32 = 2^{38}$ bits needed for these arrays.

Compared to the RKSS key recovery attack using the same distinguisher, which needs $D' = 2^{49}$ chosen plaintext-ciphertext pairs and $T' \approx 2^{114.19}$ 16-round encryptions, the probabilistic RKSS method performs much better than the original one. Specifically, $D = 3.44\% \cdot D'$.

Algorithm 1: Key recovery attack procedure of 10-round Piccolo-80 containing both pre- and post-whitening keys.

```
for 2^16 wk_2 and 2^16 wk_3 do
    Allocate and initialize two arrays V_1[x_1] and V_1^0[x_1^0] with |x_1| = 28 = |x_1^0|;
    wk_2^0 = wk_2 ⊕ 0x0 00 and wk_3^0 = wk_3;
    for N plaintexts P with P_0[0,2,3] and P_2[0,2,3] ⊕ed do
        Query the ciphertexts C under K;
        Decrypt C_0, C_2 to get Y_0[2,3], Z_0[0], Y_1[0,1] and Z_2[2,3];
        Let x_1 ← Z_0[0] || (Y_0[2,3] ⊕ C_1[2,3]) || Z_2[2,3] || (Y_1[0,1] ⊕ C_3[0,1]) and
          V_1[x_1] ← V_1[x_1] + 1;
    for N plaintexts P^0 with P_0^0[0,2,3] = P_0[0,2,3] and P_2^0[0,2,3] = P_2[0,2,3]
      do
        Query the ciphertexts C^0 under K^0;
        Decrypt C_0^0, C_2^0 to get Y_0^0[2,3], Z_0^0[0], Y_1^0[0,1] and Z_2^0[2,3];
        Let x_1^0 ← Z_0^0[0] || (Y_0^0[2,3] ⊕ C_1^0[2,3]) || Z_2^0[2,3] || (Y_1^0[0,1] ⊕ C_3^0[0,1]) and
          V_1^0[x_1^0] ← V_1^0[x_1^0] + 1;
    for 2^8 k_0^R and 2^8 k_1^L do
        Allocate V_2[x_2] and V_2^0[x_2^0] with |x_2| = 12 = |x_2^0|, and initialize them to
          zeros;
        (k_0^0)^R = k_0^R and (k_1^0)^L = k_1^L;
        for 2^28 x_1 and x_1^0 do
            Decrypt half-round for x_1 and x_1^0 to get W_1[0,2,3] ⊕ k_2[0,2,3] and
              W_1^0[0,2,3] ⊕ k_2^0[0,2,3];
            Let x_2 ← W_1[0,2,3] ⊕ k_2[0,2,3] and V_2[x_2] ← V_2[x_2] + V_1[x_1];
            Let x_2^0 ← W_1^0[0,2,3] ⊕ k_2^0[0,2,3] and V_2^0[x_2^0] ← V_2^0[x_2^0] + V_1^0[x_1^0];
        C = 0;
        for 2^12 x do
            C ← C + P_{x=0}^{2^12 - 1} (V_2[x] ⊕ V_2^0[x])^2 - (2N ⊕ 2^12);
        if C ≤ τ then
            The guessed key bits are possibly right;
            for 2^16 k_2, 2^8 k_0^L and 2^8 k_1^R do
                Use two plaintext-ciphertext pairs to check if they are right;
```

4.3 Probabilistic RKSS Attack on 17-Round Piccolo-128

Using the same distinguisher introduced in Corollary 3, we can mount a 17-round key recovery attack on Piccolo-128 by adding an extra round before it. This key recovery attack is the best one on Piccolo-128 considering both pre- and post-whitening keys in terms of the number of rounds, compared to previous known results.

Due to Corollary 3, to guarantee that $W_3 \oplus k_3$ has the same value distribution with $W_3^0 \oplus k_3^0$, we need to iterate over all possible values of the input of 14-th round $X = X_0 || X_1 || X_2 || X_3$ with $X_0[0,1] || X_2[2,3]$ ⊕ed, which is equivalent to all possible values of $U = U_0 || U_1 || U_2 || U_3$ with U_1 ⊕ed (See Fig. 5). In other words, $s = 48$ and $t = 16$ here. Under $\alpha_0 = 0.01$ and $\alpha_1 = 2^{-14.89}$, we need $N \approx 2^{43.14}$ U with the same U_1 and the threshold value $\tau \approx 2^{15.97}$. To generate

Algorithm 2: Key recovery attack procedure of 16-round Piccolo-128 with both pre- and post-whitening keys.

```
for 2^16 k_4, 2^16 k_7, 2^16 k_0 and 2^16 k_1 do
    wk_2 = k_4^L || k_7^R and wk_3 = k_7^L || k_4^R;
    wk_2^0 = wk_2, wk_3^0 = wk_3, k_4^0 = k_4, k_7^0 = k_7, k_0^0 = k_0  0x00  and k_1^0 = k_1;
    Allocate and initialize two arrays  V_1[x_1] and V_1^0[x_1^0] with |x_1| = 32 = |x_1^0|;
    for N plaintexts P with P_0[0,1] and P_2[2,3] xed do
        Query the ciphertext C for P under K;
        Decrypt C to get Z_0[2,3], Z_1[0,1], Z_2[0,1], Z_3[2,3], Y_0[0,1] and Y_1[2,3];
        Let x_1  Z_0[2,3] || (Z_1[0,1]  Y_0[0,1]) || Z_2[0,1] || (Z_3[2,3]  Y_1[2,3]) and
          V_1[x_1]  V_1[x_1] + 1;
    for N plaintexts P^0 with P_0^0[0,1] = P_0[0,1] and P_2^0[2,3] = P_2[2,3]  do
        Decrypt C^0 to get Z_0^0[2,3], Z_1^0[0,1], Z_2^0[0,1], Z_3^0[2,3], Y_0^0[0,1] and
          Y_1^0[2,3];
        Let x_1^0  Z_0^0[2,3] || (Z_1^0[0,1]  Y_0^0[0,1]) || Z_2^0[0,1] || (Z_3^0[2,3]  Y_1^0[2,3]) and
          V_1^0[x_1^0]  V_1^0[x_1^0] + 1;
    for 2^8 k_2^L and 2^8 k_5^R do
        (k_2^0)^L = k_2^L and (k_5^0)^R = k_5^R;
        Allocate and initialize two arrays  V_2[x_2] and V_2^0[x_2^0] with
          |x_2| = 16 = |x_2^0|;
        for 2^32 x_1 and x_1^0 do
            Decrypt half-round for  x_1 and x_1^0 to get W_3  k_3 and W_3^0  k_3^0;
            Let x_2  W_3  k_3, and V_2[x_2]  V_2[x_2] + V_1[x_1];
            Let x_2^0  W_3^0  k_3^0, and V_2^0[x_2^0]  V_2^0[x_2^0] + V_1^0[x_1^0];
        C = 0;
        for 2^16 x do
            C  C + ( P_{x=0}^{2^16 - 1} (V_2[x]  V_2^0[x])^2 = (2N  2^16) );
        if C  then
            The guessed key bits are possibly right;
            for 2^8 k_2^R, 2^16 k_3, 2^8 k_5^L and 2^16 k_6 do
                Use two plaintext-ciphertext pairs to check if they are right;
```

these N values of U, we traverse all possible values of P_0 and P_2, randomly choose $2^{11:14}$ values for P_3, and set $P_2 = F(P_0 \ \ wk_0)$ after guessing wk_0. U^0 can be obtained similarly. All key bits can then be recovered following Algorithm 3.

Suppose that one memory access to an array of size 2^{32} or of size $2^{59:14}$ costs less than one encryption of 17-round Piccolo-128. Then, the time complexity of this attack is $T = 2^{59:14} \ \ 2 + 2^{59:14} \ \ 4 + 2^{64} \ \ 2^{43:14} \ \ 4 + 2^{64} \ \ 2^{43:14} \ \ 2 \ \ (4=17 + 1) + 2^{64} \ \ 2^{16} \ \ 2^{32} \ \ 2 \ \ (1=2) \ \ (1=17) + 2 \ \ 2^{128} \ \ _1 \ \ 2^{115:44}$ 17-round encryptions. The data complexity is $D = 2 \ \ 2^{16}N \ \ 2^{60:14}$ chosen plaintext-ciphertext pairs. The dominant memory requirements are to store these plaintext-ciphertext pairs, about $M = 4 \ \ 2^{59:14} \ \ 64 = 2^{67:14}$ bits are needed for these arrays.

To show the advantage of this new method, we also try to mount an RKSS attack based on the same distinguisher. However, we have to use $\Omega = 2 \ \ 2^{16} \ \ 2^{48} = 2^{65}$ chosen plaintext-ciphertext pairs in such an attack. In other words, the full

Algorithm 3: Key recovery attack procedure of 17-round Piccolo-128 with both pre- and post-whitening keys.

Allocate and initialize four arrays $V_P[]$, $V_P^0[]$, $V_C[]$ and $V_C^0[]$ with size $2^{59:14}$;
Take $2^{11:14}$ distinct random values of P_3 and store them in a set S;
Choose another $2^{11:14}$ distinct random values of P_3^0 and store them in a set S^0;
a 0;
for 2^{16} P_0, 2^{16} P_1, and 2^{16} P_2 do
  for $2^{11:14}$ P_3 in set S do
    Query the ciphertexts C for P under K;
    $V_P[a] = P$, $V_C[a] = C$, and increase a by one;
a 0;
for 2^{16} P_0, 2^{16} P_1, and 2^{16} P_2 do
  for $2^{11:14}$ P_3^0 in set S^0 do
    Query the ciphertexts C^0 for P^0 under K^0;
    $V_P^0[a] = P^0$, $V_C^0[a] = C^0$, and increase a by one;
for 2^{16} k_4, 2^{16} k_7, 2^{16} k_0, and 2^{16} k_1 do
  $wk_0 = k_0^L \| k_1^R$, $wk_2 = k_4^L \| k_7^R$, and $wk_3 = k_7^L \| k_4^R$, $wk_0^0 = wk_0$, $wk_2^0 = wk_2$, $wk_3^0 = wk_3$, $k_4^0 = k_4$, $k_7^0 = k_7$, $k_0^0 = k_0$ 0x00 $_1$ $_2$, and $k_1^0 = k_1$;
  Allocate and initialize two arrays $V_1[x_1]$ and $V_1^0[x_1^0]$ with $|x_1| = 32 = |x_1^0|$;
  for 2^{16} P_0, 2^{16} P_2, and $2^{11:14}$ P_3 in set S do
    Compute $P_1 = F(P_0 \quad wk_0)$; // We have $2^{43:14}$ U with the same U_1
    Access $V_P[]$ with $P_0 \| P_1 \| P_2 \| P_3$ and get the index a, then access $V_C[a]$ to get the corresponding ciphertexts C;
    Decrypt C to get $Z_0[2;3]$, $Z_1[0;1]$, $Z_2[0;1]$, $Z_3[2;3]$, $Y_0[0;1]$ and $Y_1[2;3]$;
    Let x_1 $Z_0[2;3] \| (Z_1[0;1] \quad Y_0[0;1]) \| Z_2[0;1] \| (Z_3[2;3] \quad Y_1[2;3])$ and $V_1[x_1]$ $V_1[x_1] + 1$;
  for 2^{16} P_0, 2^{16} P_2, and $2^{11:14}$ P_3^0 in set S^0 do
    Compute $P_1 = F(P_0 \quad wk_0)$;// We have $2^{43:14}$ U^0 with $U_1^0 = U_1$
    Access $V_P^0[]$ with $P_0 \| P_1 \| P_2 \| P_3^0$ and get the index a, then access $V_C^0[a]$ to get the corresponding ciphertexts C^0;
    Decrypt C^0 to get $Z_0^0[2;3]$, $Z_1^0[0;1]$, $Z_2^0[0;1]$, $Z_3^0[2;3]$, $Y_0^0[0;1]$, $Y_1^0[2;3]$;
    Let x_1^0 $Z_0^0[2;3] \| (Z_1^0[0;1] \quad Y_0^0[0;1]) \| Z_2^0[0;1] \| (Z_3^0[2;3] \quad Y_1^0[2;3])$ and $V_1^0[x_1^0]$ $V_1^0[x_1^0] + 1$;
  for 2^8 k_2^L and 2^8 k_5^R do
    $(k_2^0)^L = k_2^L$ and $(k_5^0)^R = k_5^R$;
    Allocate and initialize two arrays $V_2[x_2]$ and $V_2^0[x_2^0]$ with $|x_2| = 16 = |x_2^0|$;
    for 2^{32} x_1 and x_1^0 do
      Decrypt half-round for x_1 and x_1^0 to get W_3 k_3 and W_3^0 k_3^0;
      Let x_2 W_3 k_3, and $V_2[x_2]$ $V_2[x_2] + V_1[x_1]$;
      Let x_2^0 W_3^0 k_3^0, and $V_2^0[x_2^0]$ $V_2^0[x_2^0] + V_1^0[x_1^0]$;
    $C = 0$;
    for 2^{16} x do
      C $C + \sum_{x=0}^{2^{16}-1} (V_2[x] \quad V_2^0[x])^2 = (2N \quad 2^{16})$;
    if C then
      The guessed key bits are possibly right;
      for 2^8 k_2^R, 2^{16} k_3, 2^8 k_5^L and 2^{16} k_6 do
        Use two plaintext-ciphertext pairs to check if they are right;

Fig. 4. Probabilistic RKSS attack on 16-round Piccolo-128 with full whitening, where are active nibbles and are nibbles that we need to know in the key recovery procedure.

codebook is used, and the attack would not be valid. Therefore, the probabilistic RKSS method can make it possible to cover one more round than the original RKSS method.

5 Conclusion and Future Work

In this paper, we revisited the RKSS cryptanalysis technique and proposed a new method called probabilistic RKSS cryptanalysis, which requires a lower

Fig. 5. One round added before the distinguisher when attacking 17-round Piccolo-128, where are active nibbles and are nibbles that we need to know in the key recovery procedure.

data complexity and has the potential of attacking more rounds than the original RKSS method. This new method was proposed by adopting an appropriate statistic that considers di erent ^{2}-distributions under right and wrong key guesses. The statistic is constructed as the squared Euclidean distance between the partial-value distributions of two ciphertext sets obtained from encrypting two independently chosen plaintext sets under related keys. The distributions of this statistic have been proved rigorously under several reasonable assumptions and con rmed experimentally using a toy cipher.

To show the e ectiveness of this new method, we have applied it to the reduced-round Piccolo. As a result, we obtained the best key recovery attacks containing both pre- and post-whitening keys on 10-round Piccolo-80 and 17-round Piccolo-128. Note that we only use 10% of the number of plaintexts required for RKSS attacks on the 10-round Piccolo-80 and the success probability only decreases by 1%. Meanwhile, the data complexity needed in the new method on 16-round Piccolo-128 is only 3 4% of that required in the RKSS method. Moreover, we can cover one additional round on Piccolo-128 using the new method.

To make a more clear comparison between the probabilistic RKSS method and the original RKSS method, some theoretical discussions, as well as key recovery attacks on reduced-round SKINNY-128-256 and full-round LiCi-2, are given in the Appendix due to space constraints.

The probabilistic RKSS method has shown its advantage compared to the original RKSS by new cryptanalysis results on Piccolo, SKINNY-128-256 and LiCi-2. The applications of this new method on other primitives are an interesting topic to explore in future work.

References

1. Ahangarkolaei, M.Z., Najarkolaei, S.R.H., Ahmadi, S., Aref, M.R.: Zero correlation linear attack on reduced round Piccolo-80. In: ISCISC 2016. pp. 66{71. IEEE (2016). https://doi.org/10.1109/ISCISC.2016.7736453
2. Ashur, T., Dunkelman, O., Masalha, N.: Linear cryptanalysis reduced round of Piccolo-80. In: Dolev, S., Hendler, D., Lodha, S., Yung, M. (eds.) CSCML 2019. LNCS, vol. 11527, pp. 16{32. Springer (2019). https://doi.org/10.1007/978-3-030-20951-3_2
3. Avanzi, R.: The QARMA block cipher family. Almost MDS matrices over rings with zero divisors, nearly symmetric Even-Mansour constructions with non-involutory central rounds, and search heuristics for low-latency S-boxes. IACR Trans. Symmetric Cryptol. 2017(1), 4{44 (2017). https://doi.org/10.13154/tosc.v2017.i1.4-44
4. Azimi, S.A., Ahmadian, Z., Mohajeri, J., Aref, M.R.: Impossible di erential cryptanalysis of Piccolo lightweight block cipher. In: ISCISC 2014. pp. 89{94. IEEE (2014). https://doi.org/10.1109/ISCISC.2014.6994028
5. Barrett, C.W., Sebastiani, R., Seshia, S.A., Tinelli, C.: Satis ability modulo theories. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satis ability, Frontiers in Arti cial Intelligence and Applications, vol. 185, pp. 825{885. IOS Press (2009). https://doi.org/10.3233/978-1-58603-929-5-825
6. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 123{153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5
7. Bogdanov, A., Boura, C., Rijmen, V., Wang, M., Wen, L., Zhao, J.: Key di erence invariant bias in block ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 357{376. Springer (2013). https://doi.org/10.1007/978-3-642-42033-7_19
8. Collard, B., Standaert, F.: A statistical saturation attack against the block cipher PRESENT. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 195{210. Springer (2009). https://doi.org/10.1007/978-3-642-00862-7_13
9. Cook, S.A.: The complexity of theorem-proving procedures. In: Harrison, M.A., Banerji, R.B., Ullman, J.D. (eds.) Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA. pp. 151{158. ACM (1971). https://doi.org/10.1145/800157.805047
10. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher Square. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149{165. Springer (1997). https://doi.org/10.1007/BFb0052343
11. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002). https://doi.org/10.1007/978-3-662-04722-4
12. Daemen, J., Rijmen, V.: Probability distributions of correlation and di erentials in block ciphers. J. Math. Cryptol. 1(3), 221{242 (2007). https://doi.org/10.1515/JMC.2007.011
13. DasGupta, A.: Asymptotic theory of statistics and probability. Springer-Verlag New York (2008). https://doi.org/10.1007/978-0-387-75971-5
14. Dobraunig, C., Eichlseder, M., Mendel, F.: Square attack on 7-round KIASU-BC. In: ACNS 2016. pp. 500{517 (2016). https://doi.org/10.1007/978-3-319-39555-5_27

15. Dong, X., Qin, L., Sun, S., Wang, X.: Key guessing strategies for linear key-schedule algorithms in rectangle attacks. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022. LNCS, vol. 13277, pp. 3{33. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_1
16. Fouque, P., Karpman, P.: Security ampli cation against meet-in-the-middle attacks using whitening. In: Stam, M. (ed.) Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8308, pp. 252{269. Springer (2013), https://doi.org/10.1007/978-3-642-45239-0_15
17. Fu, L., Jin, C., Li, X.: Multidimensional zero-correlation linear cryptanalysis of lightweight block cipher Piccolo-128. Secur. Commun. Networks 9(17), 4520{4535 (2016). https://doi.org/10.1002/sec.1644
18. Isobe, T., Shibutani, K.: Security analysis of the lightweight block ciphers XTEA, LED and Piccolo. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 71{86. Springer (2012). https://doi.org/10.1007/978-3-642-31448-3_6
19. Jeong, K., Kang, H., Lee, C., Sung, J., Hong, S.: Biclique cryptanalysis of lightweight block ciphers PRESENT, Piccolo and LED. Cryptology ePrint Archive, Paper 2012/621 (2012), https://eprint.iacr.org/2012/621
20. Khairnar, S., Bansod, G., Dahiphale, V.: A light weight cryptographic solution for 6LoWPAN protocol stack. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) Intelligent Computing. pp. 977{994. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-01177-2_71
21. Knudsen, L.R., Wagner, D.A.: Integral cryptanalysis. In: FSE 2002. pp. 112{127 (2002). https://doi.org/10.1007/3-540-45661-9_9
22. Li, M., Hu, K., Wang, M.: Related-tweak statistical saturation cryptanalysis and its application on QARMA. IACR Trans. Symmetric Cryptol. 2019(1), 236{263 (2019). https://doi.org/10.13154/tosc.v2019.i1.236-263
23. Liu, Y., Cheng, L., Liu, Z., Li, W., Wang, Q., Gu, D.: Improved meet-in-the-middle attacks on reduced-round Piccolo. Sci. China Inf. Sci. 61(3), 032108:1{032108:13 (2018). https://doi.org/10.1007/s11432-016-9157-y
24. Liu, Y., Cheng, L., Zhao, F., Su, C., Liu, Z., Li, W., Gu, D.: New analysis of reduced-version of Piccolo in the single-key scenario. KSII Trans. Internet Inf. Syst. 13(9), 4727{4741 (2019). https://doi.org/10.3837/tiis.2019.09.022
25. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386{397. Springer (1993). https://doi.org/10.1007/3-540-48285-7_33
26. Maxwell, A.E.: Comparing the classi cation of subjects by two independent judges. The British Journal of Psychiatry 116, 651{655 (1970). https://doi.org/10.1192/bjp.116.535.651
27. Minier, M.: On the security of Piccolo lightweight block cipher against related-key impossible di erentials. In: Paul, G., Vaudenay, S. (eds.) INDOCRYPT 2013. LNCS, vol. 8250, pp. 308{318. Springer (2013). https://doi.org/10.1007/978-3-319-03515-4_21
28. Mouha, N., Preneel, B.: Towards nding optimal di erential characteristics for ARX: Application to Salsa20. Cryptology ePrint Archive, Paper 2013/328 (2013), https://eprint.iacr.org/2013/328
29. Nyberg, K.: Linear approximation of block ciphers. In: Santis, A.D. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 439{444. Springer (1994). https://doi.org/10.1007/BFb0053460

30. Phan, R.C.: Mini advanced encryption standard (mini-aes): a testbed for cryptanalysis students. Cryptologia 26(4), 283{306 (2002). https://doi.org/10.1080/0161-110291890948
31. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342{357. Springer (2011). https://doi.org/10.1007/978-3-642-23951-9_23
32. Shumway, R.H., Sto er, D.S.: Time Series Analysis and Its Applications: With R Examples. Springer Texts in Statistics, Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-52452-8
33. Stuart, A.: A test for homogeneity of the marginal distribution of a two-way classi cation. Biometrika 42, 412{416 (1955), https://doi.org/10.1093/biomet/42.3-4.412
34. Todo, Y.: Impossible di erential attack against 14-round Piccolo-80 without relying on full code book. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 99-A (1), 154{157 (2016). https://doi.org/10.1587/transfun.E99.A.154
35. Tolba, M., Abdelkhalek, A., Youssef, A.M.: Meet-in-the-middle attacks on reduced round Piccolo. In: Güneysu, T., Leander, G., Moradi, A. (eds.) LightSec 2015. LNCS, vol. 9542, pp. 3{20. Springer (2015). https://doi.org/10.1007/978-3-319-29078-2_1
36. Wang, M., Cui, T., Chen, H., Sun, L., Wen, L., Bogdanov, A.: Integrals go statistical: Cryptanalysis of full Skipjack variants. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 399{415. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_20
37. Wang, Y., Wu, W., Yu, X.: Biclique cryptanalysis of reduced-round Piccolo block cipher. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 337{352. Springer (2012). https://doi.org/10.1007/978-3-642-29101-2_23
38. Zhang, K., Lai, X., Wang, L., Guan, J., Hu, B., Wang, S., Shi, T.: Related-key multiple impossible di erential cryptanalysis on full-round LiCi-2 designed for IoT. In: Security and Communication Networks (2022). https://doi.org/10.1155/2022/3611840

A Brief Introduction to Piccolo

Piccolo [31] is a 64-bit lightweight block cipher proposed at CHES 2011. The two variants Piccolo-80 and Piccolo-128 have key sizes of 80 and 128 bits, respectively.

These two variants have the same iterative structure which is a type of generalized Feistel network, but the number of rounds is di erent. The number of rounds for Piccolo-80 and Piccolo-128 is 25 and 31, respectively. Fig. 6 shows the detailed structure of Piccolo. A 64-bit plaintext P is rst divided into four 16-bit parts P_0, P_1, P_2 and P_3. Then P_0 and P_2 will be XORed with the pre-whitening keys wk_0 and wk_1, respectively. After that, 25 or 31 rounds will be evaluated to get the corresponding ciphertext C. At last, a part of the ciphertext C_0 and C_2 will be XORed with the post-whitening keys wk_2 and wk_3, respectively. The round function F consists of two S-box layers, which are composed of four parallel 4-bit S-boxes, separated by an MDS matrix M. M is a circulant matrix de ned as $\mathrm{circ}(2;3;1;1)$ where the multiplications are performed over the Galois Field $GF(2^4)$ de ned by an irreducible polynomial $x^4 + x + 1$. The round

permutation RP takes a 64-bit input value $X = (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and outputs a 64-bit value $Y = (x_2, x_7, x_4, x_1, x_6, x_3, x_0, x_5)$.

Fig. 6. The detailed structure of Piccolo.

The key schedule of Piccolo is linear. Denote $k_j = k_j^L \| k_j^R$ as a 16-bit key word, where $k_j \in \mathbb{F}_2^{16}$, $k_j^L \in \mathbb{F}_2^8$ and $k_j^R \in \mathbb{F}_2^8$. The round constants con_j^{80} and con_j^{128} are used in Piccolo-80 and Piccolo-128, respectively. For the 80-bit key $K = k_0 \| k_1 \| k_2 \| k_3 \| k_4$, the whitening keys are

$$wk_0 = k_0^L \| k_1^R, wk_1 = k_1^L \| k_0^R, wk_2 = k_4^L \| k_3^R, wk_3 = k_3^L \| k_4^R$$

and the round keys for the $(i+1)$-th round $(0 \le i \le 24)$ are

$$(rk_{2i}, rk_{2i+1}) = (con_{2i}^{80}, con_{2i+1}^{80}) \begin{cases} (k_2, k_3) & \text{if } i \bmod 5 = 0 \text{ or } 2 \\ (k_0, k_1) & \text{if } i \bmod 5 = 1 \text{ or } 4 \\ (k_4, k_4) & \text{if } i \bmod 5 = 3. \end{cases}$$

For a 128-bit key $K = k_0 \| k_1 \| k_2 \| k_3 \| k_4 \| k_5 \| k_6 \| k_7$, the whitening keys are

$$wk_0 = k_0^L \| k_1^R, wk_1 = k_1^L \| k_0^R, wk_2 = k_4^L \| k_7^R, wk_3 = k_7^L \| k_4^R.$$

Before extracting rk_j $(0 \le j \le 61)$, a word-wise permutation h will operate on K only when $(j+2) \bmod 8 = 0$, where $h(K) = k_2 \| k_1 \| k_6 \| k_7 \| k_0 \| k_3 \| k_4 \| k_5$. Hence, $rk_j = con_j^{128} \oplus k_{(j+2) \bmod 8}$.

B Experimental Verification of Assumptions Adopted

In this section, we use the same toy cipher and follow the same procedure as introduced in Sect. 3.2 to verify whether these assumptions used in Section 3.1 are acceptable.

On the Assumption of Stuart-Maxwell Test. Given 2^s paired data $(T_l(y;z), T_l(y;z'))$ evaluated under $y \in F_2^s$, we have to test whether these pairs are independent. This is equivalent to checking the autocorrelation of the sequence:

$$(T_l(y_1;z), T_l(y_1;z')),\ (T_l(y_2;z), T_l(y_2;z')),\ (T_l(y_3;z), T_l(y_3;z')),\ \ldots$$

In statistics, testing the autocorrelation of sequences [32] where only one element is involved each time, rather than a pair, can be described as follows. For a given sequence of samples $\{x_1, x_2, x_3, \ldots, x_n\}$, we evaluate its correlation with the sequence $\{x_{t+1}, x_{t+2}, x_{t+3}, \ldots, x_n\}$ that omits the first t samples (i.e., from x_1 to x_t). The autocorrelation of this sequence under the distance t is then defined as

$$R(t) = \sum_{i=1}^{n-t} \frac{(x_i - \bar{x})(x_{i+t} - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$

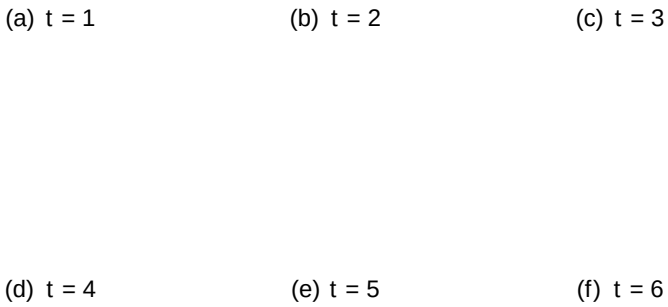
where $\bar{x}$ is the average value of all samples x_i. If these n samples are collected independently, the absolute value of $R(t)$ should fulfill $|R(t)| \approx 0$ for any $t > 0$.

In order to use the above theory to test the sequence of paired data, we mapped each pair into an integer. In experiments, since $T_l(y;z) \in F_2^4$ and $T_l(y;z') \in F_2^4$, we can transform the paired data $(T_l(y;z), T_l(y;z'))$ into $16 T_l(y;z) + T_l(y;z')$. Note that the independence of the transformed samples is equivalent to that of the original ones since it is a bijective mapping. For each t, we evaluate $R(t)$ in 1000 experiments and compare its value with zero. Since there are a lot of possible t, we only present a few of them in Fig. 7. We can see that $\Pr\{|R(t)| \le 0.04\} \ge 98\%$. Therefore, $|R(t)| \approx 0$. In other words, the independence assumption used in the Stuart-Maxwell test is fulfilled.

We also implemented the above experiments when the key difference has a low Hamming weight. In this case, the key difference of each round is set to 0x1. The corresponding results are illustrated in Fig. 8. Similarly, we have $\Pr\{|R(t)| \le 0.04\} \ge 98\%$ and thus the assumption is also fulfilled.

On $A_{i,j} \approx 0$ Used in the Proof of Lemma 1. We collect 1000 values of $A_{i,j}$ here under each $1 \le i \le 2^t - 1$ and $1 \le j \le 2^t - 1$. Some of these experimental results are presented in Fig. 9. Since $\Pr\{|A_{1,1}| \le 0.04\} \ge 95\%$, $\Pr\{|A_{1,2}| \le 0.04\} \ge 98\%$ and $\Pr\{|A_{1,3}| \le 0.04\} \ge 98\%$, we can say $|A_{1,1}| \approx 0$, $|A_{1,2}| \approx 0$ and $|A_{1,3}| \approx 0$. So, the assumption $A_{i,j} \approx 0$ is reasonable, and then $M \approx M^* = I$.

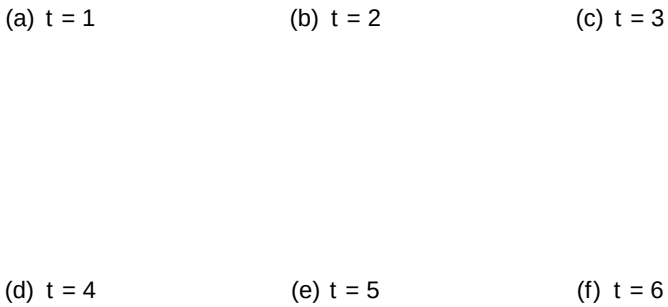
On Hypothesis 1. We collect 1000 values of $q_i q_j$, $q_i' q_j'$ and $q_i + q_j'$ under each $0 \le i \le 2^t - 1$ and $0 \le j \le 2^t - 1$. Then we compare them with $(2^{-t})^2$, $(2^{-t})^2$ and $2 \cdot 2^{-t}$, respectively. Since there are 256 combinations of (i,j) pairs, we only



(a) t = 1 (b) t = 2 (c) t = 3

(d) t = 4 (e) t = 5 (f) t = 6

Fig. 7. Experimental veri cation of the assumption of the Stuart-Maxwell test when the key di erence is randomly chosen.



(a) t = 1 (b) t = 2 (c) t = 3

(d) t = 4 (e) t = 5 (f) t = 6

Fig. 8. Experimental veri cation of the assumption of the Stuart-Maxwell test when the key di erence has a low Hamming weight.

(a) $A_{1;1}$ (b) $A_{1;2}$ (c) $A_{1;3}$

Fig. 9. Experimental results related to $A_{i;j}$ 0.

present several of them here in Fig. 10. For cases (a), (b), (d) and (e), we have $\Pr[j\ q_i q_j \quad (2^{\ t})^2 j \quad 0{:}001g \quad 97\%$ and $\Pr[j\ q_i^0 q_j^0 \quad (2^{\ t})^2 j \quad 0{:}001g \quad 97\%$; for case (c) and (f), we have $\Pr[j\ q_i + q_j^0 \quad 2 \quad 2^{\ t} j \quad 0{:}01g \quad 94\%$. Thus, all these values can be approximated by 0. Hence, Hypothesis 1 is reasonable.

(a) $q_0 q_0 \quad (2^{\ t})^2$ (b) $q_0^0 q_0^0 \quad (2^{\ t})^2$ (c) $q_0 + q_0^0 \quad 2 \quad 2^{\ t}$

(d) $q_0 q_1 \quad (2^{\ t})^2$ (e) $q_0^0 q_1^0 \quad (2^{\ t})^2$ (f) $q_0 + q_1^0 \quad 2 \quad 2^{\ t}$

Fig. 10. Experimental results related to Hypothesis 1.

C Further Discussion on the Probabilistic RKSS Method

As we can see from our applications on Piccolo, the probabilistic RKSS method has shown its ability to require less data and even cover more rounds than the original RKSS method, with only a small reduction in the success probability.

To make a clear comparison between these two methods, we also mounted key recovery attacks on reduced-round SKINNY-128-256 [6] and the full-round

LiCi-2 [20]. Due to space constraints, we omit the details of these two attacks and only list our results here.

Best Integral-Like Attacks on Round-Reduced SKINNY-128-256 in the Basic Related Tweakey Setting. SKINNY [6] is a well-known lightweight tweakable block cipher family designed by Beierle et al. at CRYPTO 2016. The cipher supports two kinds of block sizes $n \in \{64, 128\}$ and three main tweakey sizes n, $2n$ and $3n$, which are usually referred to as SKINNY-n-n, SKINNY-n-$2n$, and SKINNY-n-$3n$, respectively. Here, we only focus on SKINNY-128-256. Apart from the self-analysis by its designers [6], SKINNY has been evaluated under many cryptanalytic methods. Among all of these, the best tweakey recovery attacks on SKINNY-128-256 are given by [15].

Using the probabilistic RKSS method, we can mount tweakey recovery attacks on 20-round and 21-round SKINNY-128-256. While with the original RKSS method, we can only proceed the 20-round attack with much higher data complexities and cannot mount valid 21-round attacks. Note that our attacks are not the best known but they are the best integral-like attack results when the number of distinct tweakeys is limited to 2.

Much Faster Key Recovery Attacks on Full-Round LiCi-2. LiCi-2 [20] is a 64-bit lightweight block cipher designed by Khairnar et al. for IoT devices that supports a 128-bit key. Its full-round security has been recently broken by [38]. However, their attack requires $2^{123.44}$ full-round encryptions.

Using the (probabilistic) RKSS method, we can mount key recovery attacks costing only $2^{93.36}$ full-round encryptions. Compared with the original RKSS method, the probabilistic RKSS method also needs much lower data complexity.

Further Discussion. As we can see from the above applications, the data complexities can be reduced using the probabilistic RKSS method. However, the reduction of the data complexity is different. Denote $Q_{s,t,\epsilon_0,\epsilon_1}$ as the reduction of the data complexity, i.e.,

$$Q_{s,t,\epsilon_0,\epsilon_1} = \frac{2^{-2^s} - 2^{-N}}{2^{-2^s}} = 1 - \frac{1}{2^s}\frac{q_1^{(2^t-1)}}{q_{1-\epsilon_0}^{(2^t-1)}}.$$

For several ciphers, $Q_{s,t,\epsilon_0,\epsilon_1}$ along with other information are compared in Table 3.

From Table 3, we can see that the most important parameter for the data reduction is t, i.e., a larger t often leads to a larger $Q_{s,t,\epsilon_0,\epsilon_1}$. However, when ϵ_1 is chosen to be extremely small, we may not obtain a larger $Q_{s,t,\epsilon_0,\epsilon_1}$ even if t is larger. This is the case for the parameters in Piccolo-80 and LiCi-2. Hence, to reduce the amount of data, we have to choose a large t and ϵ_1. The parameter t is determined by the RKSS distinguisher used, while ϵ_1 is mainly influenced by the time complexity.

Table 3. Parameters in all proposed attacks, where $_0$ and $_1$ are error probabilities, s and t are determined by the RKSS distinguisher.

Cipher	s	t	Data Reduced $Q_{s;t;\ 0;\ 1}$	Chosen $_1$	Maximum of $_1$
SKINNY-128	96	8	44:52%	$2^{\ 32:1}$	$2^{\ 22:97}$
Piccolo-80	40	12	90%	$2^{\ 7:16}$	$2^{\ 7}$
LiCi-2	61	14	88:34%	$2^{\ 60}$	$2^{\ 35:64}$
Piccolo-128	48	16	96:56%	$2^{\ 14:89}$	$2^{\ 14:83}$

Usually, a larger $_1$ leads to a larger time complexity. Let T and T denote the time complexity costs in the probabilistic and original RKSS attacks, respectively. More precisely, T can be computed as

$$T = l_1 \quad N + l_2 + l_3 \quad 2 \quad {}^{g}2^{\ g} \quad {}_1;$$

where l_1 N denotes the cost of generating ciphertexts, l_2 is the cost of recovering round key bits, is the full key length, $_g$ is the number of guessed key bits, and l_3 is the number of plaintext-ciphertext pairs used to lter out the right key. For the original RKSS attack, since we have to use 2^s chosen plaintexts and $_1$ 0, T can be evaluated as

$$T = l_1 \quad 2^s + l_2 + l_3 \quad 2 \quad {}^{g}:$$

Usually, we lean toward choosing $_1$ ful lling T T, since in this case, the new method can reduce the data complexity without increasing the time complexity. When increasing $_1$, we should assure that the term l_3 2 $^{g}2^{\ g}$ $_1$ should never be larger than the maximum between l_1 2^s, l_2 and l_3 2 ^{g}. Thus,

$$_1 \quad \max \quad \frac{l_1 \quad 2^s}{l_3 \quad 2}; \frac{l_2}{l_3 \quad 2}; \frac{l_3 \quad 2 \quad {}^{g}}{l_3 \quad 2} \quad :$$

That is,

$$_1 \quad \frac{1}{l_3 \quad 2} \max \quad l_1 \quad 2^s; l_2; l_3 \quad 2 \quad {}^{g} \quad :$$

This indicates that, when T is close to 2 (i.e., the cost of exhaustive search), it may be possible to choose a larger $_1$, so that the probabilistic RKSS method works e ciently. The maximum of $_1$ in our attacks is given in Table 3.

Another important point to discuss, is the e ectiveness of our new method on QARMA-64 [3], since the RKSS method was originally proposed to attack 10-round QARMA-64. The attack uses four di erent RKSS distinguishers where s = 56 and t = 4 [22]. We can follow a similar procedure shown in [22, Algorithm 2] to recover the 128-bit key using the probabilistic RKSS method. In this case, the time complexity of this attack is T 2N 4 + 2^{128} $\frac{4}{1}$ encryption units. The data needed is D = 2 N 4 = 8 N chosen plaintext-tweak pairs, while the memory requirement is the same as the original attack. The success probability is decreased to $Pr_s = (1 \quad _0)^4$. Taking Pr_s = 99% and $_1 = 2^{\ 18:45}$, we can