SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2018, Vol. 61 032108:1–032108:13
doi: 10.1007/s11432-016-9157-y

Improved meet-in-the-middle attacks on reduced-round Piccolo

Ya LIU[1,2,3], Liang CHENG[1], Zhiqiang LIU[3,2], Wei LI[4,5*], Qingju WANG[3,6] & Dawu GU[3]

[1]*Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai* 200093*, China;*
[2]*State Key Laboratory of Cryptology, P.O. Box* 5159*, Beijing* 100878*, China;*
[3]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[4]*School of Computer Science and Technology, Donghua University, Shanghai* 201620*, China;*
[5]*Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai* 200240*, China;*
[6]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby* 2800*, Denmark*

Received 22 December 2016/Revised 5 April 2017/Accepted 21 June 2017/Published online 20 November 2017

Abstract Piccolo is a lightweight block cipher that adopts a generalized Feistel network structure with 4 branches, each of which is 16 bit long. The key length is 80 or 128 bit, denoted by Piccolo-80 and Piccolo-128, respectively. In this paper, we mounted meet-in-the-middle attacks on 14-round Piccolo-80 without pre- and post-whitening keys and 18-round Piccolo-128 with post-whitening keys by exploiting the properties of the key schedule and Maximum Distance Separable (MDS) matrix. For Piccolo-80, we first constructed a 5-round distinguisher. Then 4 rounds and 5 rounds were appended at the beginning and at the end, respectively. Based on this structure, we mounted an attack on 14-round Piccolo-80 from the 5th round to the 18th round. The data, time, and memory complexities were 2^{52} chosen plaintexts, $2^{67.44}$ encryptions, and $2^{64.91}$ blocks, respectively. For Piccolo-128, we built a 7-round distinguisher to attack 18-round Piccolo-128 from the 4th round to the 21st round. The data, time, and memory complexities were 2^{52} chosen plaintexts, $2^{126.63}$ encryptions, and $2^{125.29}$ blocks, respectively. If not considering results on biclique cryptanalysis, these are currently the best public results on this reduced version of the Piccolo block cipher.

Keywords block cipher, lightweight, Piccolo, meet-in-the-middle attack, distinguisher

Citation Liu Y, Cheng L, Liu Z Q, et al. Improved meet-in-the-middle attacks on reduced-round Piccolo. Sci China Inf Sci, 2018, 61(3): 032108, doi: 10.1007/s11432-016-9157-y

1 Introduction

Recently, resource-constrained devices such as RFID tags and sensor network nodes have been used for access controls, public transportation systems, identification, and electronic health systems. However, traditional block ciphers cannot provide cryptographic security for these tiny computing devices. A number of lightweight block ciphers have been proposed, including PRESENT [1], LBlock [2], LED [3], Piccolo [4], and TWINE [5]. In constrained environments, lightweight block ciphers can be implemented with sufficient computation speed and low power consumption, but they provide only modest security. In other words, they have to be designed with a trade-off between security and performance. Therefore, the security of lightweight block ciphers should be examined carefully.

* Corresponding author (email: liwei.cs.cn@gmail.com)

Piccolo [4] is a lightweight block cipher proposed by Sony Corporation in 2011. It has a 64-bit block length supporting 80- and 128-bit keys, which are denoted by Piccolo-80 and Piccolo-128, respectively. Piccolo-80 and Piccolo-128 use 25 rounds and 31 rounds, respectively. Before the first round and after the last round, pre- and post-whitening keys are added, respectively. The Piccolo cipher has been analyzed by meet-in-the-middle attacks, impossible differential cryptanalysis, and biclique cryptanalysis. In 2012, Isobe and Shibutani [6] performed meet-in-the-middle attacks on 14-round Piccolo-80 and 21-round Piccolo-128 without pre- and post-whitening keys. These two attacks require a full codebook or more. In 2013, Minier [7] performed related-key impossible differential cryptanalysis of 14-round Piccolo-80 and 21-round Piccolo-128 without pre- and post-whitening keys. In 2014, Azimi et al. [8] proposed impossible differential cryptanalysis of 12-round Piccolo-80 without post-whitening keys, 13-round Piccolo-80 without pre- and post-whitening keys, and 15-round Piccolo-128 without pre-whitening keys. Huang and Lai [9] performed meet-in-the-middle attacks on full-round Piccolo-80 and Piccolo-128 with complexities close to an exhaustive search . In 2015, Tolba et al. [10] proposed meet-in-the-middle attacks on 14-round Piccolo-80 without pre- and post-whitening keys and 17-round Piccolo-128 without pre-whitening keys . In addition, there are a number of studies of biclique cryptanalysis of the full-round Piccolo-80 and Piccolo-128 [11–16]. However, biclique cryptanalysis is used to improve exhaustive searches or needs a full codebook, so it is often considered a brute force cryptanalysis.

The meet-in-the-middle attack was proposed by Diffie and Hellman in 1977. It has been used to analyze the security of a substantial number of cryptographic primitives including block ciphers, stream ciphers, and hash functions [10, 17–21]. There are two common implementations of this attack. In the first, a block cipher $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ is treated as a cascade of two sub-ciphers $E = E_1 \circ E_0$. Given a guess for the subkeys used in E_0 and E_1, if a plaintext produces the same value just after E_0 as the corresponding ciphertext produces just before E_1, then this guess for the subkeys is likely to be correct; otherwise, this guess must be incorrect. Thus, we can find the correct subkey given a sufficient number of matching plaintext-ciphertext pairs. However, it is difficult to find two distinct key sets to cover a large number of rounds. Therefore, techniques including splice-and-cut [22] and initial structure [23] have been developed. The second meet-in-the-middle attack implementation was proposed by Demirci and Selçuk [18], and was used to attack 8-round AES-192 and 8-round AES-256 . At this time, the cipher can be split into three sub-ciphers, $E = E_1 \circ E_{\text{mid}} \circ E_0$. In E_{mid}, a distinguisher is constructed in an offline phase. Based on the distinguisher, the attackers guess the keys used in E_0 and E_1 and check in an online phase whether they satisfy the distinguisher or not. However, this method has high memory requirements to save the pre-computation table. Thus, techniques have been proposed to overcome this flaw including differential enumeration [24], multisets [24], efficient tabulation (rebound-like idea) [25], and a key-dependent sieve [26]. Finally, Guo et al. [27–29] presented results on meet-in-the-middle attacks on generic Feistel constructions .

In this paper, we performed meet-in-the-middle attacks on 14-round Piccolo-80 without pre- and post-whitening keys and 18-round Piccolo-128 with post-whitening keys. First, we built a 5-round distinguisher of Piccolo-80. By appending 4 rounds at its top and 5 rounds at its bottom, we mounted an attack on 14-round Piccolo-80 without pre- and post-whitening keys from the 5th round to the 18th round. The data, time, and memory complexities were 2^{52} chosen plaintexts, $2^{67.44}$ encryptions, and $2^{64.91}$ blocks, respectively. Second, we constructed a 7-round distinguisher of Piccolo-128. We then set 4 rounds at the beginning and 7 rounds at the end to present an attack on 18-round Piccolo-128 with post-whitening keys from the 4th round to the 21st round. The data, time, and memory complexities were 2^{52} chosen plaintexts, $2^{126.63}$ encryptions, and $2^{125.29}$ blocks, respectively. In our attacks, we made full use of the redundancy of the key schedule and the property of Maximum Distance Separable (MDS) to reduce the complexity. In Table 1, we list results on the reduced-round Piccolo in a single key scenario, but do not include results on brute force cryptanalysis.

The remainder of this paper is organized as follows. The Piccolo block cipher and notations used are introduced in Section 2. Section 3 proposes a meet-in-the-middle attack on a 14-round Piccolo-80. Section 4 proposes a meet-in-the-middle attack on an 18-round Piccolo-128. Section 5 summarizes this paper.

Table 1 Summary of partial attacks on the Piccolo block cipher in a single key scenario [a)]

Key size	NR	Attack type	Pre/Post	Time (Enc)	Data	Memory (block)	Source
Piccolo-80	14	MITMA	None	2^{73}	2^{64}†	2^{5}	[6]
	12	IDA	Pre	$2^{55.18}$	$2^{36.34}$CC	2^{63}	[8]
	13	IDA	None	$2^{69.7}$	$2^{43.25}$CP	2^{62}	[8]
	14	MITMA	None	$2^{75.39}$	2^{48}CP	$2^{73.49}$	[10]
	14	MITMA	None	$2^{67.44}$	2^{52}CP	$2^{64.91}$	Section 3
Piccolo-128	21	MITMA	None	2^{121}	2^{64}†	2^{6}	[6]
	15	IDA	Post	$2^{125.4}$	$2^{58.7}$CP	2^{61}	[8]
	16	MITMA	Post	2^{123}	2^{48}CP	$2^{113.49}$	[10]
	17	MITMA	Post	$2^{126.87}$	2^{48}CP	$2^{125.99}$	[10]
	18	MITMA	Post	$2^{126.63}$	2^{52}CP	$2^{125.29}$	Section 4

a) NR, Pre, Post, MITMA, IDA, CP, CC, Enc, and † denote the number of rounds, pre-whitening key, post-whitening key, meet-in-the-middle attack, impossible differential attack, chosen plaintexts, chosen ciphertexts, encryption, and full codebook or more, respectively.

2 Preliminaries

2.1 Notations

- $A_{(b)}$: a word A whose length is b bit.
- $A \parallel B$: concatenation of two words A and B.
- M^{T}: transposition of the matrix or the vector M.
- A_b: the word A is represented in base b.
- K: the master key.
- k_i: the ith nibble of K from the left, whose length is 16 bit.
- $rk_{2i} \parallel rk_{2i+1}$: the 32-bit key used in round i.
- $wk_0 \parallel wk_1 \parallel wk_2 \parallel wk_3$: the pre- and post-whitening keys.
- X_i: the 64-bit input of round i, where $0 \leqslant i < 26$ in Piccolo-80 and $0 \leqslant i < 32$ in Piccolo-128.
- Y_i: the 64-bit state before applying the RP function in the ith round.
- P: the plaintext.
- C: the ciphertext.
- $X_i[j]$: the jth nibble of X_i, where $0 \leqslant j < 16$.
- $X_i[j:l]$: from j-th to l-th nibbles of X_i, where $j < l$.
- $X_i[j,l]$: the j-th and l-th nibbles of X_i.
- $\triangle X_i$,$\triangle X_i[j]$: the difference at state X_i and nibble $X_i[j]$, respectively.
- X_i^j: the jth state of the 64-bit input in the round i.

2.2 Description of Piccolo

Piccolo adopts a generalized Feistel network (GFN) structure with 4 branches of 16 bit each. Its round function consists of two Feistel networks, each of which includes an F-function and an XOR operation with a round key. Before the first round and after last round, the pre- and post-whitening keys are added. The detailed structure can be seen in Figures 1–4 of [4]. Based on the notations given above, the encryption algorithm (Algorithm 1) and key schedule of Piccolo are presented below.

The F-function consists of two S-box layers separated by a diffusion matrix M. Each S-box layer applies the same four 4×4-bit bijective S-boxes in parallel. The diffusion function updates the internal state by the matrix M as follows:

$$(x_{0(4)}, x_{1(4)}, x_{2(4)}, x_{3(4)})^{\mathrm{T}} = M \cdot (x_{0(4)}, x_{1(4)}, x_{2(4)}, x_{3(4)})^{\mathrm{T}},$$

where the multiplication is performed over a finite field GF (2^4). The round permutation RP acts at byte level and permutes the bytes of the current block as follows:

$$\mathrm{RP} : (x_{0(8)}, x_{1(8)}, \ldots, x_{7(8)}) \rightarrow (x_{2(8)}, x_{7(8)}, x_{4(8)}, x_{1(8)}, x_{6(8)}, x_{3(8)}, x_{0(8)}, x_{5(8)}).$$

Algorithm 1 Encryption algorithm

Let P be the plaintext.
$P = X_{0(64)} = x_{0(16)} \parallel x_{1(16)} \parallel x_{2(16)} \parallel x_{3(16)}$.
$x_{0(16)} = x_{0(16)} \oplus \omega k_0, x_{2(16)} = x_{2(16)} \oplus \omega k_1$.
for $i = 0$ to $r - 2$ **do**
    $y_{0(16)} = x_{0(16)}$, $y_{1(16)} = x_{1(16)} \oplus F(x_{0(16)}) \oplus rk_{2i}$, $y_{2(16)} = x_{2(16)}$, $y_{3(16)} = x_{3(16)} \oplus F(x_{2(16)}) \oplus rk_{2i+1}$.
    $x_{0(16)} \parallel x_{1(16)} \parallel x_{2(16)} \parallel x_{3(16)} = RP(y_{0(16)} \parallel y_{1(16)} \parallel y_{2(16)} \parallel y_{3(16)})$.
end for
$y_{0(16)} = x_{0(16)} \oplus wk_2$, $y_{1(16)} = x_{1(16)} \oplus F(x_{0(16)}) \oplus rk_{2r-2}$, $y_{2(16)} = x_{2(16)} \oplus wk_3$, $y_{3(16)} = x_{3(16)} \oplus F(x_{2(16)}) \oplus rk_{2r-1}$.
$C = y_{0(16)} \parallel y_{1(16)} \parallel y_{2(16)} \parallel y_{3(16)}$.

Key schedule. The key schedule divides an 80-bit master key K into five 16-bit subkeys such that $K = k_{0(16)} \parallel k_{1(16)} \parallel k_{2(16)} \parallel k_{3(16)} \parallel k_{4(16)}$ or an 128-bit master key K into eight 16-bit subkeys such that $K = k_{0(16)} \parallel k_{1(16)} \parallel k_{2(16)} \parallel k_{3(16)} \parallel k_{4(16)} \parallel k_{5(16)} \parallel k_{6(16)} \parallel k_{7(16)}$. Then it generates whitening keys and round keys as shown in Algorithms 2 and 3.

Algorithm 2 Key schedule employed in Piccolo-80

Input: $K = k_{0(16)} \parallel k_{1(16)} \parallel k_{2(16)} \parallel k_{3(16)} \parallel k_{4(16)}$.
Output: $wk_i, 0 \leqslant i \leqslant 3$ and $rk_i, 0 \leqslant i \leqslant 49$.
$wk_0 = k_0^L \parallel k_1^R$, $wk_1 = k_1^L \parallel k_0^R$, $wk_2 = k_4^L \parallel k_3^R$, $wk_3 = k_3^L \parallel k_4^R$.
for $i = 0$ to 24 **do**

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

end for

Algorithm 3 Key schedule employed in Piccolo-128

Input: $K = k_{0(16)} \parallel k_{1(16)} \parallel k_{2(16)} \parallel k_{3(16)} \parallel k_{4(16)} \parallel k_{5(16)} \parallel k_{6(16)} \parallel k_{7(16)}$.
Output: $wk_i, 0 \leqslant i \leqslant 3$ and $rk_i, 0 \leqslant i \leqslant 61$.
$wk_0 = k_0^L \parallel k_1^R$, $wk_1 = k_1^L \parallel k_0^R$, $wk_2 = k_4^L \parallel k_7^R$, $wk_3 = k_7^L \parallel k_4^R$.
for $i = 0$ to 61 **do**
    if $(i + 2) \bmod 8{=}0$ **then**
        $(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7) = (k_2, k_1, k_6, k_7, k_0, k_3, k_4, k_5)$.
    end if
    $rk_i = k_{(i+2) \bmod 8} \oplus \mathrm{con}_i^{128}$.
end for

In Algorithms 2 and 3, con_i^{80} and con_i^{128} are constants. Detailed information can be found in [4].

3 A meet-in-the-middle attack on 14-round Piccolo-80

In this section, we applied the strategy used by Demirci and Selçuk [18] on AES to attack reduced-round Piccolo-80. We first constructed a 5-round distinguisher on Piccolo-80, which we employed to attack 14-round Piccolo-80 from the 5th to 18th rounds (i.e., rounds 4 to 17) without pre- and post-whitening keys. Finally, we analyzed the complexity of our attack.

3.1 A 5-round distinguisher on Piccolo-80

In our attack, we first chose a δ-set at the second input branch of the Feistel network (FN). Then we evaluated the ordered sequence at the first output branch and constructed a 5-round distinguisher that minimized the number of parameters. During δ-set construction, we utilized the following property of the diffusion matrix M.

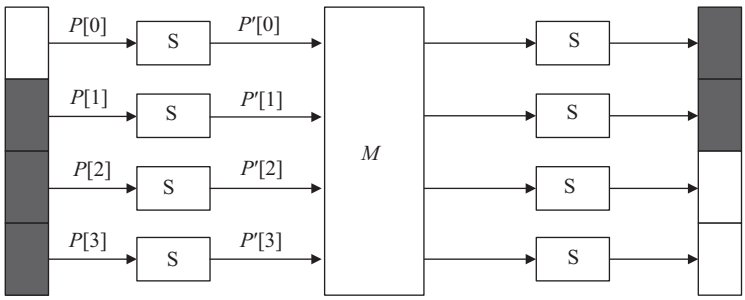

Figure 1 The property of the diffusion matrix M.

Lemma 1 ([10]). If the input of the linear transformation M contains three active nibbles and its output has two active nibbles, as shown in Figure 1, the number of such differences is 15 by enumerating all the possible values.

Based on Lemma 1, we constructed a δ-set as follows.

Proposition 1. Consider the encryption of a δ-set $\{P^0, P^1, \ldots, P^j\}$ through 5 rounds of Piccolo, where $P^i = X_0^i[5, 14, 15] \parallel X_0[0, \ldots, 4] \parallel X_0[6, \ldots, 13]$ $(i = 0, \ldots, j)$, and $X_0^i[5, 14, 15]$ are active nibbles and others are inactive. Then the ordered sequence $X_5^0[6:7] \oplus X_5^1[6:7]$, $X_5^0[6:7] \oplus X_5^2[6:7]$, $\ldots$, $X_5^0[6:7] \oplus X_5^j[6:7]]$ is fully determined by the 16-bit parameters $X_1^0[0:3]$, $X_2^0[0:3]$, $X_2^0[8:11]$, and $X_3^0[8:11]$. The detailed structure has been shown in Figure 2.

Proof. The value of $[X_0^0 \oplus X_0^1, X_0^0 \oplus X_0^2, \ldots, X_0^0 \oplus X_0^j]$ can be computed by the knowledge of the δ-set $\{X_0^0, X_0^1, \ldots, X_0^j\}$. After one-round encryption, we can calculate the partial input difference of the second round $X_1^0[0:3] \oplus X_1^i[0:3]$ $(i = 1, \ldots, j)$. If the value of $X_1^0[0:3]$ is known, the partial output differences of the second round $Y_1^0[0:7] \oplus Y_1^i[0:7]$ $(i = 1, \ldots, j)$ can be calculated, i.e., $X_2^0[0, 1, 6, 7, 10, 11, 12, 13] \oplus X_2^i[0, 1, 6, 7, 10, 11, 12, 13]$ $(i = 1, \ldots, j)$ can be known. Other nibbles of $X_2^0 \oplus X_2^i$ $(i = 1, \ldots, j)$ are zero. In the following, if $X_2^0[0:3]$ and $X_2^0[8:11]$ have been guessed, then we can calculate the value of $Y_2^0[6, 7, 10, 11, 12, 13] \oplus Y_2^i[6, 7, 10, 11, 12, 13]$ $(i = 1, \ldots, j)$, which can be used to calculate $X_3^0[8, 9, 10, 11, 14, 15] \oplus X_3^i[8, 9, 10, 11, 14, 15]$ $(i = 1, \ldots, j)$. If $X_3^0[8:11]$ is known, then we can compute the value of $Y_3^0[14, 15] \oplus Y_3^i[14, 15]$ $(i = 1, \ldots, j)$, i.e., $X_5^0[6:7] \oplus X_5^i[6:7]$ can be computed. Thus the ordered sequence $X_5^0[6:7] \oplus X_5^1[6:7]$, $X_5^0[6:7] \oplus X_5^2[6:7]$, $\ldots$, $X_5^0[6:7] \oplus X_5^j[6:7]$ is fully determined by the 4 16-bit parameters $X_1^0[0:3]$, $X_2^0[0:3]$, $X_2^0[8:11]$, and $X_3^0[8:11]$.

Note. Because $\mathrm{RP}^{-1}[X_0^0[4, 5, 14, 15] \oplus X_0^i[4, 5, 14, 15]] = Y_{-1}^0[8, 9, 10, 11] \oplus Y_{-1}^i[8, 9, 10, 11]$ $(i = 1, \ldots, j)$, we can obtain $Y_{-1}^0[9, 10, 11] \oplus Y_{-1}^i[9, 10, 11] \neq 0$ and $Y_{-1}^0[8] \oplus Y_{-1}^i[8] = 0$ for $i = 1, \ldots, j$. If $F(Y_{-1}^0[8, 9, 10, 11]) \oplus F(Y_{-1}^i[8, 9, 10, 11])$ $(i = 0, 1, \ldots, j)$ has two non-zero nibbles, then the number of such differences is 15 by Lemma 1. Select a random value of $X_0[5, 14, 15]$ and Xor with these 15 differences to obtain 16 plaintexts that construct the δ-set, i.e., $j = 15$ in Proposition 1. At this time, we have $2^{4\times16} = 2^{64}$ 120-bit ordered sequences out of the $2^{15\times8} = 2^{120}$ theoretically possible ones.

3.2 Attack procedure

In this subsection, we mounted a meet-in-the-middle attack on 14-round Piccolo-80 from the 5th to 18th rounds without the pre- and post-whitening keys based on the 5-round distinguisher described in Proposition 1. The attack relied on the property of M and the redundancy in the key schedule. In fact, if we selected the δ-set from the second input branch of the FN and the ordered sequence was computed at its first output branch, only 2 rounds could be added before this distinguisher. If 3 or more rounds were added, the full codebook would be needed because of the diffusion transformation M. Therefore, we utilized the property of the diffusion operation M in the plaintext direction (Lemma 1) so that we could append 4 rounds at the beginning. At the end of the distinguisher, 5 rounds were added. To reduce the data complexity, we moved subkeys rk_8 and rk_9 to the following round in our attack. The detailed structure can be found in Figure 3.

The attack on 14-round Piccolo-80 has two phases as follows.

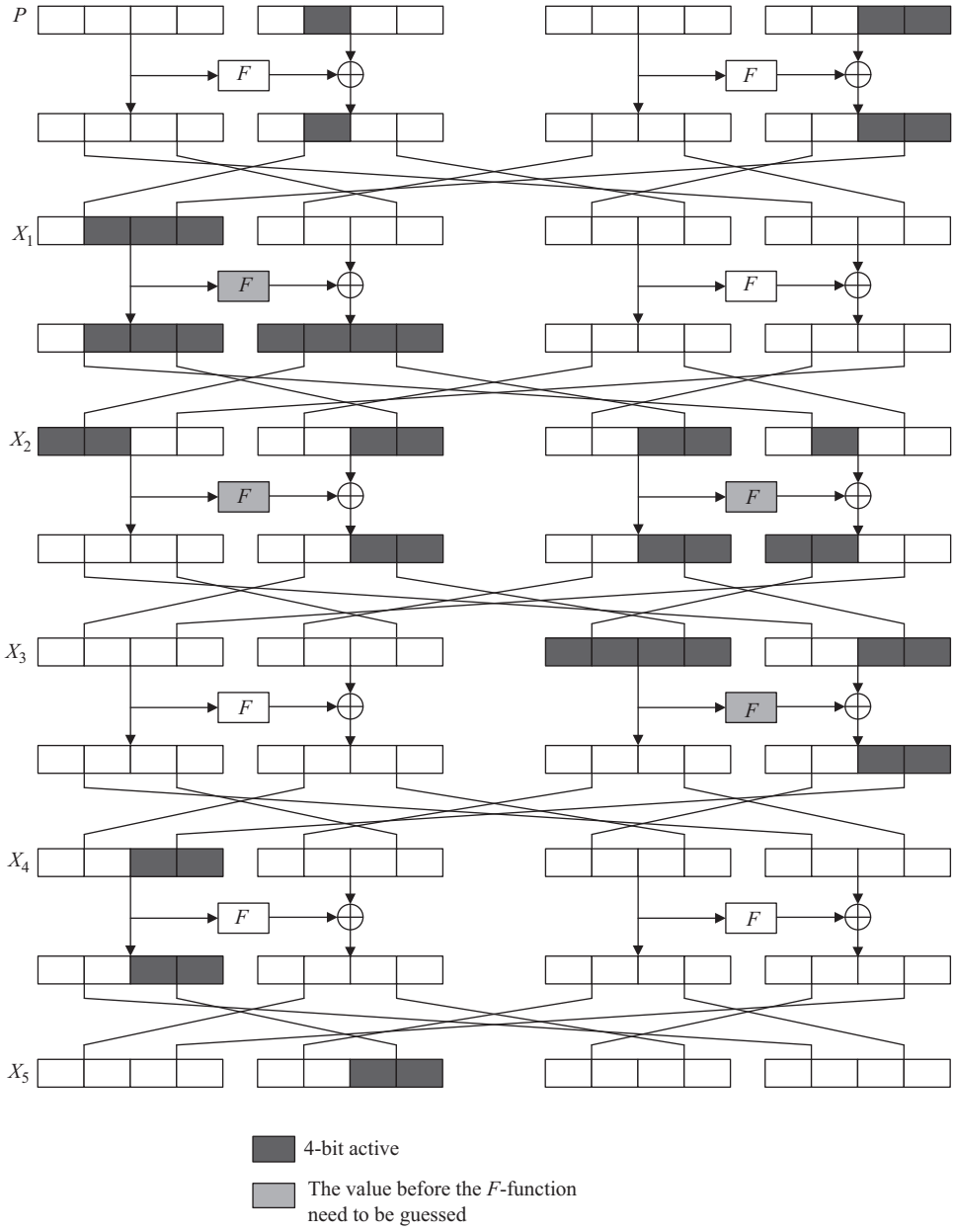

Figure 2 A 5-round distinguisher on Piccolo-80.

• **Offline phase.** By Proposition 1, we determine all 2^{64} ordered sequences and store them in a hash table H.

• **Online phase.**

(1) Select a plaintext P^0. Then we can encrypt one round to obtain the value of X_5^0. Guess the subkeys $rk_8^L \parallel rk_9^R, rk_8^R \parallel rk_9^L, rk_{10}$, and rk_{11}. Then X_6^0 can be calculated. In the following, we guess the value of rk_{12}^R and rk_{13}^L. The value of $Y_6^0[0:3] \parallel Y_6^0[6,7] \parallel Y_6^0[8:13]$ can be computed. Therefore, the value of $X_7^0[8:11]$ can be obtained, and the value of $X_8^0[4,5,14,15]$ can be known.

(2) As noted in Proposition 1, we obtain 15 differences, which equals $X_8^0 \oplus X_8^i \triangleq \Delta X_8^i$ $(i = 1, \ldots, 15)$. Among them, $\Delta X_8^i[5,14,15]$ are non-zero nibbles and others are zero. Therefore, we can know the value of $X_8^i[4,5,14,15]$ $(i = 1, \ldots, 15)$, and the value of $X_7^i[8:11]$ $(i = 1, \ldots, 15)$ is known. We can compute the value of $X_7^0 \oplus X_7^i \triangleq \Delta X_7^i$ $(i = 1, \ldots, 15)$, i.e., $Y_6^i \oplus Y_6^0 \triangleq \Delta Y_6^i$ $(i = 1, \ldots, 15)$.

(3) Since $Y_6^0[0:3] \parallel Y_6^0[8:11]$ and ΔY_6^i $(i = 1, \ldots, 15)$ have been obtained, the value of $X_6^i \oplus X_6^0 \triangleq \Delta X_6^i$ $(i = 1, \ldots, 15)$ can be calculated. The value of $Y_5^i \oplus Y_5^0 \triangleq \Delta Y_5^i$ $(i = 1, \ldots, 15)$ can also be known.

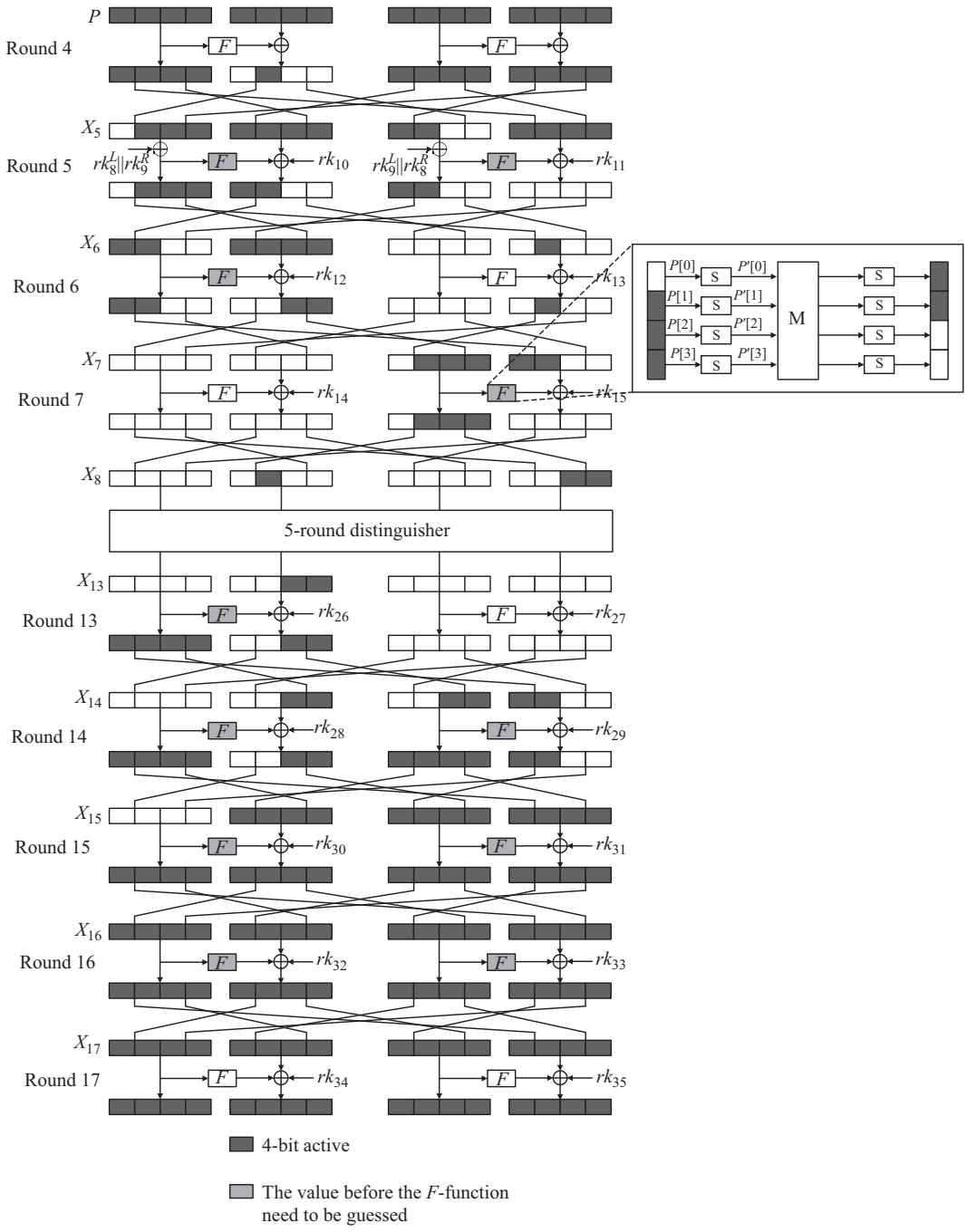

Figure 3 Meet-in-the-middle attacks on 14-round Piccolo-80.

(4) Because Y_5^0(i.e., X_6^0) and ΔY_5^i $(i = 1, \ldots, 15)$ have been obtained, we can calculate the value of $X_5^i \oplus X_5^0 \triangleq \Delta X_5^i$ $(i = 1, \ldots, 15)$. Therefore, the value of X_5^i $(i = 1, \ldots, 15)$ can be obtained. By decrypting one round, we can obtain the other 15 plaintexts $P^1, P^2, \ldots, P^{15}$ by the plaintext P^0.

(5) Ask for the corresponding ciphertexts $C^0, C^1, \ldots, C^{15}$.

(6) Guess subkeys rk_{30}, rk_{31}, rk_{32}, rk_{33}, rk_{34}, rk_{35}. Then we can decrypt the ciphertexts C^0, C^1, $\ldots, C^{15}$ to obtain the value of X_{15}^i (i.e., Y_{14}^i) $(i = 0, \ldots, 15)$.

(7) Next, guess the value of rk_{28}^R, rk_{29}^L. Then we can compute the value of $X_{14}^i[6, 7, 10, 11, 12, 13]$ $(i = 0, \ldots, 15)$. Thus we can know the value of $Y_{13}^i[0:3]|Y_{13}^i[6, 7]$ $(i = 0, \ldots, 15)$. Finally, the ordered sequence $[X_5^0[6:7] \oplus X_5^1[6:7], X_5^0[6:7] \oplus X_5^2[6:7], \ldots, X_5^0[6:7] \oplus X_5^j[6:7]]$ can be computed.

(8) Build the sequence and check if it belongs to the table H.

In our attack, we guessed the subkeys rk_8, rk_9, rk_{10}, rk_{11}, rk_{12}^R, rk_{13}^L, rk_{28}^R, rk_{29}^L, rk_{30}, rk_{31}, rk_{32}, rk_{33}, rk_{34}, and rk_{35}. By the key schedule, we obtained a relationship between some round subkeys as follows.

Proposition 2. The related subkeys have the following relationships:

$$(rk_8, rk_9) = (\mathrm{con}_8^{80}, \mathrm{con}_9^{80}) \oplus (k_0, k_1);\ (rk_{10}, rk_{11}) = (\mathrm{con}_{10}^{80}, \mathrm{con}_{11}^{80}) \oplus (k_2, k_3);$$

$$(rk_{12}, rk_{13}) = (\mathrm{con}_{12}^{80}, \mathrm{con}_{13}^{80}) \oplus (k_0, k_1);\ (rk_{28}, rk_{29}) = (\mathrm{con}_{28}^{80}, \mathrm{con}_{29}^{80}) \oplus (k_0, k_1);$$

$$(rk_{30}, rk_{31}) = (\mathrm{con}_{30}^{80}, \mathrm{con}_{31}^{80}) \oplus (k_2, k_3);\ (rk_{32}, rk_{33}) = (\mathrm{con}_{32}^{80}, \mathrm{con}_{33}^{80}) \oplus (k_0, k_1);$$

$$(rk_{34}, rk_{35}) = (\mathrm{con}_{34}^{80}, \mathrm{con}_{35}^{80}) \oplus (k_2, k_3).$$

By Proposition 2, the guessed subkeys are related to $k_0, k_1, k_2,$ and k_3. Thus we guessed 2^{64} keys. We expect that only $2^{64-(120-64)} = 2^8$ keys remain after Step (8). Finally, we used two plaintext-ciphertext pairs to verify the remaining 2^8 key candidates along with all guessed values of k_4 so as to recover the master key.

3.3 Complexity analysis

The memory complexity was determined by the pre-computation table H built in the offline phase. This hash table consisted of 2^{64} ordered sequences, each of which has 15 8-bit differences. So the memory complexity was $2^{64} \times 120/64 = 2^{64.91}$ 64-bit blocks. The upper bound of data complexity is $2^{13\times4} = 2^{52}$ chosen plaintexts according to X_5. The time complexity contained the time complexities of the offline and online phases. The time complexity of the offline phase was determined by building hash table H, i.e., $2^{64} \times 16 \times 4/(2 \times 14) = 2^{65.19}$. The time complexity of the online phase was about $2^{64} \times 16 \times (6+9)/(2 \times 14) + 2 \times 2^{(64-(120-64))} \times 2^{16} \approx 2^{67.1}$. Hence, the total time complexity of this attack was $2^{65.19} + 2^{67.1} + 2^{25} \approx 2^{67.44}$ 14-round Piccolo-80 encryptions.

4 A meet-in-the-middle attack on 18-round Piccolo-128

In this section, we constructed a 7-round distinguisher to mount a meet-in-the-middle attack on 18-round Piccolo-128 from the 4th to 21st rounds (i.e., rounds 3 to 20) with post-whitening keys.

4.1 A 7-round distinguisher on Piccolo-128

We used Lemma 1 to construct the same δ-set as Proposition 1 to obtain a 7-round distinguisher. The detailed structure is depicted in Figure 4.

Proposition 3. Consider the encryption of a δ-set $\{P^0, P^1, \ldots, P^j\}$ through 7 rounds of Piccolo, where $P^i = X_0^i[5, 14, 15] \parallel X_0^i[0, \ldots, 4] \parallel X_0^i[6, \ldots, 13]$ $(i = 1, \ldots, j)$ and three nibbles $X_0^i[5, 14, 15]$ $(i = 1, \ldots, j)$ are active and other nibbles are inactive. Then the ordered sequence $[X_7^0[5:7] \oplus X_7^1[5:7], X_7^0[5:7] \oplus X_7^2[5:7], \ldots, X_7^0[5:7] \oplus X_7^j[5:7]]$ is fully determined by the following 8 16-bit parameters: $X_1^0[0:3]$, $X_2^0[0:3]$, $X_2^0[8:11]$, $X_3^0[0:3]$, $X_3^0[8:11]$, $X_4^0[0:3]$, $X_4^0[8:11]$, and $X_5^0[8:11]$.

Proof. The proof of this proposition is similar to Proposition 1. According to the value of $\{P^0, P^1, \ldots, P^j\}$, we can compute $P^i \oplus P^0 \triangleq \Delta P^i$ $(i = 1, \ldots, j)$. After one-round encryption, we can calculate the value of $X_1^i \oplus X_1^0 \triangleq \Delta X_1^i$ $(i = 1, \ldots, j)$. Guessing the value of $X_1^0[0:3]$, we can compute the value of $X_2^i \oplus X_2^0 \triangleq \Delta X_2^i$ $(i = 1, \ldots, j)$. Second, guessing the value of $X_2^0[0:3] \parallel X_2^0[8:11]$, we can obtain the value of $X_3^i \oplus X_3^0 \triangleq \Delta X_3^i$ $(i = 1, \ldots, j)$. Third, guessing the value of $X_3^0[0:3] \parallel X_3^0[8:11]$, we can calculate the value of $X_4^i \oplus X_4^0 \triangleq \Delta X_4^i$ $(i = 1, \ldots, j)$. Next, we guess the value of $X_4^0[0:3] \parallel X_4^0[8:11]$, and we can then calculate the value of $X_5^i[8:15] \oplus X_5^0[8:15] \triangleq \Delta X_5^i[8:15]$. Finally, we guess the value of $X_5^0[8:11]$ to obtain the value of $X_7^i[5:7] \oplus X_7^i[5:7] \triangleq \Delta X_7^i[5:7]$. Thus, the ordered sequence $[X_7^0[5:7] \oplus X_7^1[5:7], X_7^0[5:7] \oplus X_7^2[5:7], \ldots, X_7^0[5:7] \oplus X_7^j[5:7]]$ is fully determined by the following 8 16-bit parameters: $X_1^0[0:3]$, $X_2^0[0:3]$, $X_2^0[8:11]$, $X_3^0[0:3]$, $X_3^0[8:11]$, $X_4^0[0:3]$, $X_4^0[8:11]$, and $X_5^0[8:11]$.

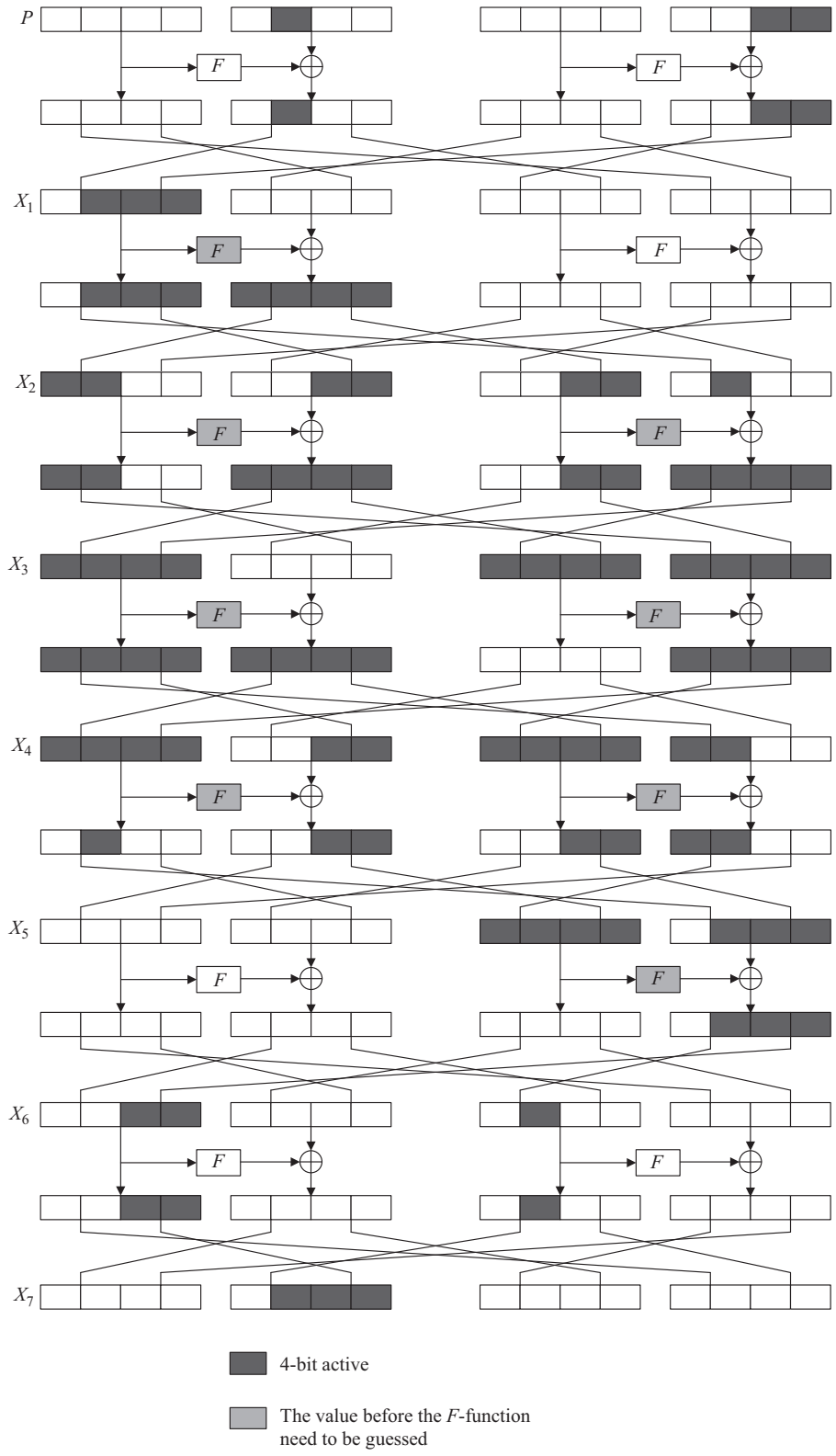

Figure 4 A 7-round distinguisher on Piccolo-128.

We selected the same differences as Proposition 1. Therefore, $j = 15$ in Proposition 3. At this time, we had $2^{8\times16} = 2^{128}$ 180-bit ordered sequences out of the $2^{15\times8} = 2^{180}$ theoretically possible ones in Proposition 3.

4.2 Attack procedure

Based on the 7-round distinguisher, we proposed a meet-in-the-middle attack on 18-round Piccolo-128 from the 4th round to the 21st round by appending 4 rounds at the beginning and 7 rounds at the end. In our attack, we shifted rk_6 and rk_7 from the 4th round to the 5th round equivalently. The attack procedure can be seen in Figure 5.

This attack contains offline and online phases. In the offline phase, we build a pre-computation table on the order sequence $[X_7^0[5:7] \oplus X_7^1[5:7], X_{14}^0[5:7] \oplus X_{14}^2[5:7], \ldots, X_{14}^0[5:7] \oplus X_{14}^{15}[5:7]]$. In the online phase, we encrypt and decrypt some chosen plaintext-ciphertexts and check whether they satisfy H' or not. This attack is similar to the meet-in-the-middle attack on 14-round Piccolo-80.

- **Offline phase.** By Proposition 3, we store all 2^{128} 180-bit ordered sequences in a hash table H'.
- **Online phase.**

(1) Pick a plaintext P^0.

(2) Guess round keys rk_6, rk_7, rk_8, rk_9, rk_{10}^R, and rk_{11}^L to identify a δ-set P^0, P^1, ..., P^{15} containing P^0.

(3) Ask for the corresponding ciphertexts $C^0, C^1, \ldots, C^{15}$.

(4) Guessing round keys rk_{30}^R, rk_{31}^L, rk_{32}, ..., rk_{40}, rk_{41}, wk_2, and wk_3, we decrypt the ciphertexts to determine the ordered sequences $[X_{14}^0[5:7] \oplus X_{14}^1[5:7], X_{14}^0[5:7] \oplus X_{14}^2[5:7], \ldots, X_{14}^0[5:7] \oplus X_{14}^{15}[5:7]]$.

(5) Build the sequence and check if it belongs to the table H'.

In our attack, we needed to guess round subkeys rk_6, rk_7, rk_8, rk_9, rk_{10}^R, rk_{11}^L, rk_{30}^R, rk_{31}^L, rk_{32}, ..., rk_{40}, rk_{41}, wk_2, and wk_3. By the key schedule, we obtained the following relationships between the round subkeys.

Proposition 4. The related subkeys have the following relationships:

$$wk_2 = k_4^L \parallel k_7^R, wk_3 = k_7^L \parallel k_4^R;$$

$$rk_6 = k_2 \oplus \mathrm{con}_6^{128};\ rk_7 = k_1 \oplus \mathrm{con}_7^{128};\ rk_8 = k_6 \oplus \mathrm{con}_8^{128};$$
$$rk_9 = k_7 \oplus \mathrm{con}_9^{128};\ rk_{10} = k_0 \oplus \mathrm{con}_{10}^{128};\ rk_{11} = k_3 \oplus \mathrm{con}_{11}^{128};$$
$$rk_{30} = k_0 \oplus \mathrm{con}_{30}^{128};\ rk_{31} = k_1 \oplus \mathrm{con}_{31}^{128};\ rk_{32} = k_2 \oplus \mathrm{con}_{32}^{128};$$
$$rk_{33} = k_7 \oplus \mathrm{con}_{33}^{128};\ rk_{34} = k_4 \oplus \mathrm{con}_{34}^{128};\ rk_{35} = k_3 \oplus \mathrm{con}_{35}^{128};$$
$$rk_{36} = k_6 \oplus \mathrm{con}_{36}^{128};\ rk_{37} = k_5 \oplus \mathrm{con}_{37}^{128};\ rk_{38} = k_2 \oplus \mathrm{con}_{38}^{128};$$
$$rk_{39} = k_1 \oplus \mathrm{con}_{39}^{128};\ rk_{40} = k_6 \oplus \mathrm{con}_{40}^{128};\ rk_{41} = k_5 \oplus \mathrm{con}_{41}^{128}.$$

According to Proposition 4, the guessed subkeys were related to seven and a half keys k_0^R, k_1, k_2, k_3, k_4, k_5, k_6, and k_7. By adjusting the guessed order of the round subkeys, we minimized the time complexity of our attack.

4.3 Complexity analysis

We estimated that the memory complexity was $2^{8\times16} \times (15 \times 12)/64 \approx 2^{129.49}$ 64-bit blocks. To reduce the memory complexity below 2^{128}, we used a simple time-memory trade-off technique. We chose α=$2^{4.2}$ so that the memory complexity was $2^{125.29}$ 64-bit blocks. The data complexity was 2^{52} chosen plaintexts. The time complexity of the offline phase was estimated to be $2^{128-4.2} \times 16 \times 8/(2 \times 18) \approx 2^{125.63}$. To reduce the time complexity of the online phase, we used a partial computation technique. First, we identified the δ-set by guessing the values of k_2, k_1, k_6, k_7, k_0^R, and k_3^L. This step was evaluated to be $2^{80} \times 16 \times 7/(2 \times 18) \approx 2^{81.64}$. Then, we decrypted rounds 20, 19, and 18, and the first F-function in round 17 by guessing k_4 and k_5. This step was evaluated to be $2^{112} \times 16 \times 7/(2 \times 18) \approx 2^{113.64}$. Finally, guessing k_3^R, we computed the ordered sequence. This step needed $2^{120} \times 16 \times 6/(2 \times 18) \approx 2^{121.42}$ encryptions. Accordingly, the time complexity of the online phase was $2^{81.64} + 2^{113.64} + 2^{121.42} \approx 2^{121.43}$. Moreover, it was repeated $2^{4.2}$ times, so the whole time complexity was about $2^{125.63}$. We used two

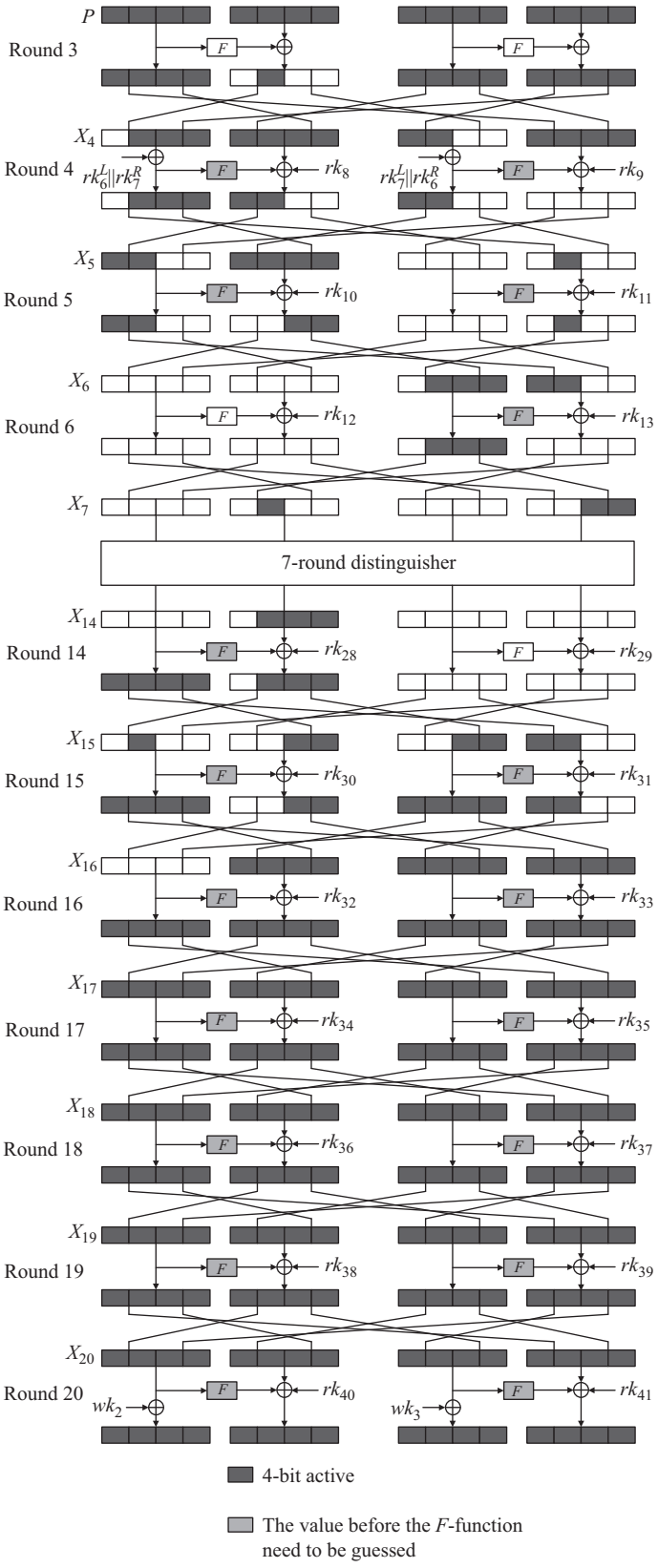

Figure 5 Meet-in-the-middle attacks on 18-round Piccolo-128.

plaintext-ciphertexts to recover the master key, which required $2 \times 2^{120} \times 2^{128-180} \times 2^8 = 2^{77}$ encryptions. Therefore, the total time complexity of this attack was $2^{125.63} + 2^{77} + 2^{125.63} \approx 2^{126.63}$ encryptions. The

success probability of this attack was about $1-(1-2^{-4.2})^{2^{4.2}} \approx 64\%$. Although the success probability was not 100%, we attacked the maximum number of rounds of Piccolo-128. Moreover, the success probability could be improved by attacking several times.

5 Conclusion

In this paper, we studied the properties of the linear diffusion layer and key schedule to mount meet-in-the-middle attacks on 14-round Piccolo-80 and 18-round Piccolo-128. Specifically, we constructed a 5-round distinguisher and a 7-round distinguisher that were used to attack 14-round Piccolo-80 and 18-round Piccolo-128, respectively. The data complexities of the two attacks were the same, i.e., 2^{52} chosen plaintexts. The time complexities of a 14-round Piccolo-80 and an 18-round Piccolo-128 were $2^{67.44}$ and $2^{126.63}$, respectively, and their memory complexities were $2^{64.91}$ and $2^{125.29}$, respectively.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61402288, 61672347, 61772129, 61472250), National Basic Research Program of China (Grant No. 2013CB338004), Shanghai Natural Science Foundation (Grant Nos. 15ZR1400300, 16ZR1401100), Innovation Program of Shanghai Municipal Education Commission (Grant No. 14ZZ066), Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security(Grant No. AGK201703). The authors are grateful to Dr. Lei WANG and the reviewers for their valuable suggestions and comments.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems-CHES 2007. Berlin: Springer-Verlag, 2007. 450–466
2 Wu W, Zhang L. LBlock: a lightweight block cipher. In: Applied Cryptography and Network Security-ACNS 2011. Berlin: Springer-Verlag, 2011. 327–344
3 Guo J, Peyrin T, Poschmann A, et al. The LED block cipher. In: Cryptographic Hardware and Embedded Systems-CHES 2011. Berlin: Springer-Verlag, 2011. 326–341
4 Shibutani K, Isobe T, Hiwatari H, et al. Piccolo: an ultra-lightweight blockcipher. In: Cryptographic Hardware and Embedded Systems-CHES 2011. Berlin: Springer-Verlag, 2011. 342–357
5 Suzaki T, Minematsu K, Morioka S, et al. TWINE: a lightweight block cipher for multiple platforms. In: Selected Areas in Cryptography-SAC 2012. Berlin: Springer-Verlag, 2013. 339–354
6 Isobe T, Shibutani K. Security analysis of the lightweight block ciphers XTEA, LED and Piccolo. In: Proceedings of Australasian Conference on Information Security and Privacy-ACISP 2012. Berlin: Springer-Verlag, 2012. 71–86
7 Minier M. On the security of Piccolo lightweight block cipher against related-key impossible differentials. In: Progress in Cryptology-INDOCRYPT 2013. Berlin: Springer-Verlag, 2013. 308–318
8 Azimi S, Ahmadian Z, Mohajeri J, et al. Impossible differential cryptanalysis of Piccolo lightweight block cipher. In: Proceedings of International ISC Conference on Information Security and Cryptology-ISCISC 2014. Piscataway: IEEE, 2014. 89–94
9 Huang J L, Lai X J. What is the effective key length for a block cipher: an attack on every practical block cipher. Sci China Inf Sci, 2014, 57: 072110
10 Tolba M, Abdelkhalek A, Youssef A M. Meet-in-the-middle attacks on reduced round Piccolo. In: Lightweight Cryptography for Security and Privacy-LightSec 2015. Berlin: Springer-Verlag, 2016. 3–20
11 Jeong K, Kang H, Lee C, et al. Biclique cryptanalysis of lightweight block ciphers PRESENT, Piccolo and LED. IACR Cryptology ePrint Archive, 2012, 2012: 621
12 Wang Y, Wu W, Yu X. Biclique cryptanalysis of reduced-round Piccolo block cipher. In: Information Security Practice and Experience-ISPEC 2012. Berlin: Springer-Verlag, 2012. 337–352
13 Ahmadi S, Ahmadian Z, Mohajeri J, et al. Low-data complexity biclique cryptanalysis of block ciphers with application to Piccolo and HIGHT. IEEE Trans Inf Foren Sec, 2014, 9: 1641–1652
14 Jeong K. Cryptanalysis of block cipher Piccolo suitable for cloud computing. J Supercomput, 2013, 66: 829–840
15 Song J, Lee K, Lee H. Biclique cryptanalysis on lightweight block cipher: HIGHT and Piccolo. Int J Comput Math, 2013, 90: 2564–2580
16 Gong Z, Liu S, Wen Y, et al. Biclique cryptanalysis using balanced complete bipartite subgraphs. Sci China Inf Sci, 2016, 59: 049101
17 Biryukov A, Derbez P, Perrin L. Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In: Fast Software Encryption-FSE 2015. Berlin: Springer-Verlag, 2015. 3–27

18 Demirci H, Selçuk A A. A meet-in-the-middle attack on 8-round AES. In: Fast Software Encryption-FSE 2008. Berlin: Springer-Verlag, 2008. 116-126
19 Chen J, Li L. Low data complexity attack on reduced camellia-256. In: Proceedings of Australasian Conference on Information Security and Privacy-ACISP 2012. Berlin: Springer-Verlag, 2012. 101–114
20 Bogdanov A, Rechberger C. A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. In: Selected Areas in Cryptography-SAC 2010. Berlin: Springer-Verlag, 2011. 229–240
21 Jia K, Yu H, Wang X. A meet-in-the-middle attack on the full kasumi. IACR Cryptol ePrint Archive, 2011, 2011: 466
22 Aoki K, Sasaki Y. Preimage attacks on one-block MD4, 63-step MD5 and more. In: Selected Areas in Cryptography-SAC 2008. Berlin: Springer-Verlag, 2009. 103–119
23 Sasaki Y, Aoki K. Finding preimages in full MD5 faster than exhaustive search. In: Advances in Cryptology-EUROCRYPT 2009. Berlin: Springer-Verlag, 2009. 134–152
24 Dunkelman O, Keller N, Shamir A. Improved single-key attacks on 8-round AES-192 and AES-256. In: Advances in Cryptology-ASIACRYPT 2010. Berlin: Springer-Verlag, 2010. 158–176
25 Derbez P, Fouque P -A, Jean J. Improved key recovery attacks on reduced-round AES in the single-key setting. In: Advances in Cryptology C EUROCRYPT 2013. Berlin: Springer-Verlag, 2013. 371–387
26 Li L, Jia K, Wang X. Improved single-key attacks on 9-round AES-192/256. In: Fast Software Encryption-FSE 2015. Berlin: Springer-Verlag, 2015. 127–146
27 Guo J, Jean J, Nikolic I, et al. Meet-in-the-middle attacks on generic Feistel constructions. In: Advances in Cryptology-ASIACRYPT 2014. Berlin: Springer-Verlag, 2014. 458–477
28 Guo J, Yu S. Extended meet-in-the-middle attacks on some Feistel constructions. Design Code Cryptogr, 2016, 80: 587–618
29 Guo J, Jean J, Nikolic I, et al. Meet-in-the-middle attacks on classes of contracting and expanding Feistel constructions. IACR Transact Symmetric Cryptol, 2017, 2016: 307–337