Impossible Differential Attack on Reduced-Round TWINE

Xuexin Zheng[1] and Keting Jia[2](✉)

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, School of Mathematics, Shandong University, Jinan, China
zhxuexin@mail.sdu.edu.cn

[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China
ktjia@mail.tsinghua.edu.cn

Abstract. TWINE, proposed at the ECRYPT Workshop on Lightweight Cryptography in 2011, is a 64-bit lightweight block cipher consisting of 36 rounds with 80-bit or 128-bit keys. In this paper, we give impossible differential attacks on both versions of the cipher, which is an improvement over what the designers claimed to be the best possible. Although our results are not the best considering different cryptanalysis methods, our algorithm which can filter wrong subkeys that have more than 80 bits and 128 bits for TWINE-80 and TWINE-128 respectively shows some novelty. Besides, some observations which may be used to mount other types of attacks are given. Overall, making use of some complicated subkey relations and time-memory tradeoff trick, the time, data and memory complexity of attacking 23-round TWINE-80 are $2^{79.09}$ 23-round encryptions, $2^{57.85}$ chosen plaintexts and $2^{78.04}$ blocks respectively. Besides, the impossible differential attack on 24-round TWINE-128 needs $2^{58.1}$ chosen plaintexts, $2^{126.78}$ 24-round encryptions and $2^{125.61}$ blocks of memory.

Keywords: TWINE · Lightweight block cipher · Impossible differential attack

1 Introduction

Impossible differential attack is a powerful cryptanalysis method introduced by Biham et al. [2] and Knudsen [10] independently. It is often used in cryptanalyzing block ciphers with (generalized) Feistel structures and SPN structures. The main trick of this method is to find an impossible differential path as long as possible and then extend two truncated differentials from it. Then any candidate subkey involved in both truncated differentials, which can lead to the impossible differential path is a wrong key and should be discarded. So long as enough

This work is partially supported by the National 973 Program of China (Grant No. 2013CB834205), and the National Natural Science Foundation of China (Grant No. 61133013).

H.-S. Lee and D.-G. Han (Eds.): ICISC 2013, LNCS 8565, pp. 123–143, 2014.
DOI: 10.1007/978-3-319-12160-4_8

Table 1. Summary of attacks on TWINE

Key (bits)	Number of rounds	Data (block)	Time (encryption)	Memory (block)	Attack	Source
80	22	2^{62}	$2^{68.43}$	2^{67}	Saturation attack	[15]
	23	$2^{57.85}$	$2^{79.09}$	$2^{78.04}$	Impossible differential attack	Section 4
	36	2^{60}	$2^{79.10}$	2^{8}	Biclique attack	[6]
128	23	$2^{62.81}$	$2^{106.14}$	2^{103}	Saturation attack	[15]
	24	$2^{58.1}$	$2^{126.78}$	$2^{125.61}$	Impossible differential attack	Section 5
	25	2^{48}	2^{122}	2^{125}	MITM	[3]
	27	$2^{62.95}$	$2^{119.5}$	2^{60}	Key-difference invariant bias attack	[1]
	36	2^{60}	$2^{126.82}$	2^{8}	Biclique attack	[6]

plaintext-ciphertext pairs are collected, an attacker can eliminate all wrong keys and recover the right key.

Due to the requirement of lightweight encryption algorithms which are used in tiny computing devices, such as RFID and sensor network nodes, many lightweight block ciphers have been proposed, for example PRESENT, KATAN, KTANTAN, KLEIN, LED, HIGHT, LBlock, TWINE [4,5,7–9,11–16], and much more. TWINE is a 64-bit lightweight block cipher designed by Suzaki, Minematsu, Morioka and Kobayashi in [15], which has two versions supporting 80-bit and 128-bit keys respectively. Consisting of 36 rounds, TWINE employs Type-2 generalized Feistel structure with 16 nibbles. When TWINE was proposed, the designers presented security evaluation including impossible differential attacks on 23-round TWINE-80 and 24-round TWINE-128 which were the most powerful attacks given by the designers. Unfortunately, the time complexity of their impossible differential attacks may have a flaw and may lead to a complexity of more than exhaustive key search. Besides the designers' security analysis, Çoban et al. gave an biclique analysis of full round TWINE [6], Boztaş et al. gave an multidimensional meet-in-the-middle attack on reduced-round TWINE-128 [3], Bogdanov et al. gave an key-difference invariant bias attack on reduced-round TWINE-128 [1]. All the results are summarized in Table 1. Note that although our results are not the best considering different cryptanalysis methods, our algorithm which can filter wrong subkeys that have more than 80 bits and 128 bits for TWINE-80 and TWINE-128 respectively shows some novelty. Besides, some observations which may be used to mount other types of attacks are given.

Our Contribution. This paper focuses on the security of TWINE against impossible differential attack. The novelty includes the following aspects:

- Propose an algorithm to filter wrong subkeys which exceeds the master key size;

- Several observations on key relations and optimization of our algorithm are given;
- Several tables are precomputed to decrease the time complexity.

This paper is organized as follows. In Sect. 2, we present the necessary notations and a simple description of the TWINE encryption algorithm and the key schedule. Section 3 gives useful observations and the reason for our choice of the impossible differential paths. Section 4 first explains the flaw of attacks in [15], and then shows the impossible differential attack against 23-round TWINE-80. The result of attacking 24-round TWINE-128 is showed in Sect. 5. Section 6 concludes the paper.

2 Preliminaries

Some notations used in this paper and a simple description of the TWINE algorithm are given in this section.

2.1 Notations

$\tilde{0}^m$: the concatenation of m 4-bit 0s. C_L^r, C_H^r: constants used in the Key Schedule of TWINE.
$x||y$: the concatenation of x and y. $k(i,j)$: $k_i \oplus s[k_j]$, where s stands for 4-bit sbox.
$A_{[i_1,...,i_m]}$: $A_{i_1}||...||A_{i_m}$. $RK^r_{[0,...,7]}$: the 32-bit round subkey of round r.
α_{i+1}: one possible value for output difference of sbox with input difference α_i.
β_{i+1}: one possible value for output difference of sbox with input difference β_i.
$\triangle s[b]$: $\{s[x] \oplus s[x \oplus b] | x \in \{0, ..., f\}\}$ the set of output differences of s with input difference b.
$a \in \triangle s[b]$: a is one of the possible output difference of sbox with input difference b.
$(X_0^r, X_1^r, ..., X_{14}^r, X_{15}^r)$: the 64-bit input value of round r.
$\#RK_p^r$: the number of possible values of RK_p^r for each plaintext-ciphertext pair.

2.2 Description of TWINE

TWINE is a 64-bit block cipher with 80-bit or 128-bit key. The global structure of TWINE is a variant of Type-2 generalized Feistel structure with 16 nibbles. Consisting of 8 4-bit S-boxes and a diffusion permutation π as described in Table 2, the round function of TWINE is showed in Fig. 1. Expressed in a formula form, the round function encrypts an input value of round r to the input value of round $r+1$ in the following two steps:

$$X_{2j+1}^r \leftarrow s[X_{2j}^r \oplus RK_j^r] \oplus X_{2j+1}^r (j = 0, ..., 7),$$
$$X_{\pi(i)}^{r+1} \leftarrow X_i^r.$$

For both versions of TWINE, the round function is iterated for 36 times and the diffusion permutation is omitted in the last round.

The key schedules of TWINE-80 and TWINE-128 produce 36 32-bit round subkeys $RK^r_{[0,...,7]}$ $(r = 1, ..., 36)$ from the 80-bit master key (denoted as $k_0, ..., k_{19}$) and 128-bit master key (denoted as $k_0, ..., k_{31}$) respectively as described in Algorithm D.1. and Algorithm D.2. (Appendix D).

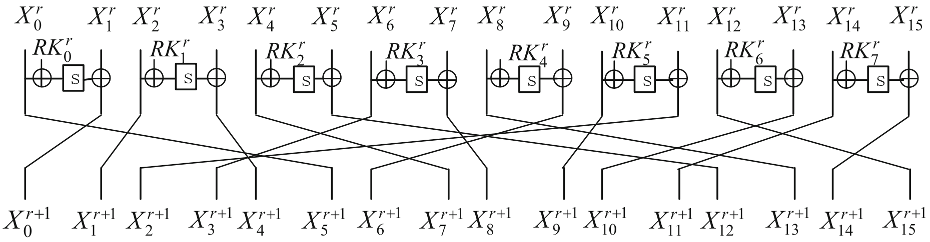

Fig. 1. Round function of TWINE

Table 2. S-box and π permutation

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$s[x]$	C	0	F	A	2	B	9	5	8	3	D	7	1	E	6	4

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(x)$	5	0	1	4	7	12	3	8	13	6	9	2	15	10	11	14

3 Observations and 14-Round Impossible Differentials of TWINE

This section gives several useful observations and the reason for our choice of the impossible differential path. Observation 1 is used in [15]. For the sake of completeness, we describe it here. Observation 2, 3, 4, 5 are about the subkeys. We give the round subkeys of TWINE-80 from round 1 to round 5 and the round subkeys of TWINE-128 from round 1 to round 7 in Table D.1 and Table D.2 (Appendix D).

Observation 1. *For any input difference $a(\neq 0)$ and output difference $b(\in \triangle s[a])$ of the sbox in TWINE, the average number of pairs that satisfy the differential characteristic $(a \to b)$ is $\frac{16}{7}$. Given an 8-bit pair (X_{2i}^r, X_{2i+1}^r) and $(X_{2i}^r \oplus a, X_{2i+1}^r \oplus b)$, the probability that RK_i^r leads to the sbox differential characteristic $(a \to b)$ is 7^{-1}.*

Observation 2. *The round subkeys of TWINE-80 satisfy the following equations among four adjacent rounds.*

$$RK_5^{r+2} = RK_1^r; RK_3^{r+2} = RK_5^r; RK_6^{r+2} = s^{-1}[RK_7^{r+1} \oplus RK_0^r] \oplus C_L^{r+1}, (1 \le r \le 34);$$
$$RK_4^{r+3} = RK_3^r; RK_0^{r+3} = RK_4^r; RK_1^{r+3} = RK_6^r \oplus C_H^{r+2}; RK_2^{r+3} = RK_7^r, (1 \le r \le 33);$$
$$RK_6^{r+3} = RK_2^r \oplus s[RK_7^r] \oplus C_L^{r+2}, (1 \le r \le 33).$$

Observation 3. *The round subkeys of TWINE-80 satisfy the following equations among RK^1, RK^2, RK^{21}, RK^{22} and RK^{23}.*

$$\begin{array}{llll}
f_1(RK^2_{[2,7]}, RK^{22}_2, RK^{23}_1) & = 0; & f_2(RK^1_1, RK^2_4, RK^{21}_7, RK^{22}_{[3,4,6]}, RK^{23}_{[0,4]}) & = 0; \\
f_3(RK^2_6, RK^{22}_{[2,5,6]}) & = 0; & f_4(RK^1_{[5,7]}, RK^{21}_{[4,7]}, RK^{22}_6, RK^{23}_{[0,4]}) & = 0; \\
f_6(RK^1_{[1,6]}, RK^{23}_{[3,4,5]}) & = 0; & f_5(RK^1_5, RK^2_6, RK^{21}_4, RK^{22}_{[1,5]}, RK^{23}_3) & = 0; \\
f_7(RK^1_0, RK^2_7, RK^{23}_{[2,5,6]}) & = 0; & p &
\end{array}$$

The precise expression of functions $f_i (i = 1, ..., 8)$ are shown in Appendix A.

Observation 4. *The round subkeys of TWINE-128 satisfy the following equations among six adjacent rounds.*

$$RK_7^{r+5} = RK_2^{r+1} \oplus s[RK_6^r]; RK_6^{r+5} = RK_4^r \oplus s[RK_2^{r+1} \oplus s[RK_6^r]], (1 \le r \le 31);$$
$$RK_7^{r+4} = RK_2^r \oplus s[RK_2^{r+3}]; RK_3^{r+4} = RK_7^r \oplus C_L^{r+3} \oplus s[RK_1^{r+1}], (1 \le r \le 32);$$
$$RK_4^{r+4} = RK_0^r; RK_5^{r+4} = RK_1^r; RK_0^{r+4} = RK_5^r; RK_2^{r+4} = RK_6^r, (1 \le r \le 32);$$
$$RK_1^{r+3} = RK_3^r \oplus C_H^{r+2}, (1 \le r \le 33).$$

Observation 5. *The round subkeys of TWINE-128 satisfy the following equations among RK^1, RK^2, RK^3, RK^4, RK^{21}, RK^{22}, RK^{23} and RK^{24}.*

$$g_1(RK_1^1, RK_{[2,3]}^{22}, RK_5^{23}) = 0;$$
$$g_2(RK_6^1, RK_2^2, RK_0^{21}, RK_{[6,7]}^{24}) = 0;$$
$$g_3(RK_{[0,1]}^3, RK_0^{21}, RK_2^{22}, RK_{[5,7]}^{23}, RK_2^{24}) = 0;$$
$$g_4(RK_5^1, RK_3^2, RK_1^3, RK_2^{21}, RK_6^{22}, RK_0^{23}, RK_{[2,3]}^{24}) = 0;$$
$$g_5(RK_{[0,1]}^1, RK_5^3, RK_0^4, RK_{[0,2]}^{22}, RK_{[1,2,4]}^{23}, RK_{[5,7]}^{24}) = 0;$$
$$g_6(RK_{[0,7]}^1, RK_{[4,5]}^2, RK_5^3, RK_{[0,2]}^{22}, RK_{[1,2,3,4,7]}^{23}, RK_{[5,7]}^{24}) = 0;$$
$$g_7(RK_{[2,4,6]}^1, RK_{[0,2,3,7]}^2, RK_{[1,3]}^3, RK_2^{21}, RK_6^{22}, RK_{[0,3]}^{23}, RK_{[4,5]}^{24}) = 0;$$
$$g_8(RK_{[2,4,6]}^1, RK_{[0,2,6,7]}^2, RK_{[1,3,5]}^3, RK_0^{22}, RK_{[0,1,2,4]}^{23}, RK_{[4,5,7]}^{24}) = 0;$$
$$g_9(RK_{[2,4,5,6]}^1, RK_{[2,3,7]}^2, RK_{[0,1,3]}^3, RK_{[0,2]}^{21}, RK_6^{22}, RK_{[0,5]}^{23}, RK_{[1,4]}^{24}) = 0.$$

The precise expression of functions $g_i (i = 1, ..., 9)$ are shown in Appendix A.

The 14-Round Impossible Differential Paths. Several 14-round impossible differential paths are given in [15]. This paper uses $(0||\alpha||\tilde{0}^{14}) \overset{14r}{\nrightarrow} (\tilde{0}^7||\beta||\tilde{0}^8)$ and $(\tilde{0}^5||\alpha||\tilde{0}^{10}) \overset{14r}{\nrightarrow} (\tilde{0}^{11}||\beta||\tilde{0}^4)$ in attacking TWINE-80 and TWINE-128 respectively. Our choice of the impossible differential paths is determined by the following two reasons. Making use of the relations in Observation 2 and Observation 4, the truncated differential paths involve the least number of round subkeys. What's more, the truncated differential paths involve subkeys that have less complicated equations in Observation 3 and Observation 5. Observation 6 is used in [15]. For the sake of completeness, we give a clear description. Observation 6 and 7 are useful in selecting more accurate plaintext/ciphertext pairs for attacking TWINE-80 and TWINE-128 respectively. Observation 8 is used in key recovery phase of our attacking TWINE-80. Its proof gives a detailed computation and analysis of the number of co responding subkeys that passing the differential path.

Observation 6. *If the impossible differential $(0||\alpha||\tilde{0}^{14}) \overset{14r}{\nrightarrow} (\tilde{0}^7||\beta||\tilde{0}^8)$ is extended 4 rounds ahead and 5 rounds behind, then the input difference is of the form*

$$(\alpha_3, \alpha_4, 0, \alpha_2, \tilde{0}^6, \alpha_1, \alpha_2^{''}, \alpha_1^{'}, \alpha_2^{'}, 0, \alpha)$$

where $\alpha \neq 0$, $\alpha_2^{'} \in \triangle s[\alpha_1^{'}]$, $\alpha_1^{'} \in \triangle s[\alpha]$, $\alpha_3 \in \triangle s[\alpha_2]$, $\alpha_2^{''} \in \triangle s[\alpha_1]$, $\alpha_4 \in \triangle s[\alpha_3]$, $\alpha_2 \in \triangle s[\alpha_1]$, $\alpha_1 \in \triangle s[\alpha]$;
and the output difference is of the form

$$(0,\beta_1^{'},0,\beta_3,\beta_2^{'},\beta_3^{'},\beta,x,\beta_4,\beta_5,\beta_2,\beta_3^{'''},\beta_2^{''},\beta_3^{''},\tilde{0}^2)$$

where $\beta \neq 0$, $\beta_3^{'} \in \triangle s[\beta_2^{'}], \beta_5 \in \triangle s[\beta_4], \beta_3^{'''} \in \triangle s[\beta_2], \beta_3^{''} \in \triangle s[\beta_2^{''}], \beta_2^{'} \in \triangle s[\beta_1^{'}], \beta_4 \in \triangle s[\beta_3], \beta_3 \in \triangle s[\beta_2], \beta_1^{'} \in \triangle s[\beta]$;
$Pr(\alpha\beta \neq 0,$ *and all the relations hold*$) = (\frac{15}{16})^2 \cdot (\frac{7}{16})^{15} = 2^{-18.08}$.

Observation 7. *If the impossible differential* $(\tilde{0}^5||\alpha||\tilde{0}^{10}) \overset{14r}{\nrightarrow} (\tilde{0}^{11}||\beta||\tilde{0}^4)$ *is extended 5 rounds on the top and the bottom of it respectively, then the input difference is of the form*

$$(\alpha_4,\alpha_5,0,\alpha_3,\alpha_2^{'},\alpha_3^{'},\tilde{0}^3,\alpha_1^{'},\alpha_2,\alpha_3^{'''},\alpha_2^{''},\alpha_3^{''},\alpha,y)$$

where $\alpha \neq 0$, $\alpha_5 \in \triangle s[\alpha_4]$, $\alpha_3^{'} \in \triangle s[\alpha_2^{'}]$, $\alpha_3^{'''} \in \triangle s[\alpha_2]$, $\alpha_3^{''} \in \triangle s[\alpha_2^{''}]$, $\alpha_2^{'} \in \triangle s[\alpha_1^{'}]$, $\alpha_1^{'} \in \triangle s[\alpha]$, $\alpha_3 \in \triangle s[\alpha_2]$, $\alpha_4 \in \triangle s[\alpha_3]$;
and the output difference is of the form

$$(\beta_2^{'},\beta_3^{'},\beta_4,\beta_5,0,\beta_1^{'},\beta_2^{''},\beta_3^{''},0,\beta_3,\tilde{0}^2,\beta,x,\beta_2,\beta_3^{'''})$$

where $\beta \neq 0$, $\beta_3^{'} \in \triangle s[\beta_2^{'}], \beta_5 \in \triangle s[\beta_4], \beta_3^{'''} \in \triangle s[\beta_2], \beta_3^{''} \in \triangle s[\beta_2^{''}], \beta_2^{'} \in \triangle s[\beta_1^{'}], \beta_4 \in \triangle s[\beta_3], \beta_3 \in \triangle s[\beta_2], \beta_1^{'} \in \triangle s[\beta]$;
$Pr(\alpha\beta \neq 0,$ *and all the belonging relations holds*$) = (\frac{15}{16})^2 \cdot (\frac{7}{16})^{16} = 2^{-19.27}$.

Observation 8. *For a plaintext-ciphertext pair satisfying the input-output difference relations in Observation 6, the following can be deduced according to the differential path in attacking TWINE-80:*

(1) Given $RK^1_{[1,6,7]}, RK^2_6$ *that pass the differential path, then* $\frac{16}{7}$ *values of* RK^1_2 *on average can pass the path and be computed;*
(2) Given $RK^{23}_{[2,3,4,5]}$ *that pass the differential path, then* $\frac{16}{7}$ *values of* RK^{22}_0 *on average can pass the path and be computed;*
(3) Given $RK^{23}_{[3,6]}$ *that pass the differential path, then* $\frac{16}{7}$ *values of* RK^{22}_4 *on average can pass the path and be computed;*
(4) Given $RK^{23}_{[1,3,4,5]}, RK^{22}_{[0,5]}$ *that pass the differential path, then* $(\frac{16}{7})^2$ *values of* RK^{21}_7 *on average can pass the path and be computed.*

Proof

(1) Compute X^4_2 using $RK^4_1 = RK^1_6 \oplus C^3_H$ and $(\triangle X^4_2, \triangle X^4_3)$, where we get $\#X^4_2 = 16/7$ for every RK^1_6. Besides, $X^3_{11} = X^2_{14}$ is computed using RK^1_7 by partial encryption. Then X^3_{10} is computed using $RK^3_5 = RK^1_1$ by partial decryption, where we get $\#X^3_{10} = 16/7$ for every $RK^1_{[1,6,7]}$. After that, together with the known $X^2_{13} = X^1_8$ and RK^2_6, we get the values of X^2_{12} where $\#X^2_{12} = 16/7$ for every $(RK^1_{[1,6,7]}, RK^2_6)$. Finally, with the knowledge of $X^1_{[4,5]}$, we can compute RK^1_2 with $\#RK^1_2 = 16/7$ for every $(RK^1_{[1,6,7]}, RK^2_6)$.
(2) Compute $X^{21}_3 = X^{20}_6$ using $RK^{20}_3 = RK^{23}_4$ and $(\triangle X^{20}_6, \triangle X^{20}_7)$, where we get $\#X^{21}_3 = 16/7$ for every RK^{23}_4. Besides, X^{22}_4 is computed using RK^{23}_3. Then X^{22}_1 is computed using $RK^{21}_1 = RK^{23}_5$, where we get $\#X^{22}_1 = 16/7$ for

every $RK^{23}_{[3,4,5]}$. What's more, X^{22}_0 is computed using RK^{23}_2. Then together with the known $X^{22}_0 = X^{23}_0$, we can compute RK^{22}_0 with $\#RK^{22}_0 = 16/7$ for every $RK^{23}_{[2,3,4,5]}$.

(3) Compute $X^{22}_9 = X^{21}_{10}$ using $RK^{21}_5 = RK^{23}_3$ and $(\triangle X^{21}_{10}, \triangle X^{21}_{11})$, where we get $\#X^{22}_9 = 16/7$ for every RK^{23}_3. Besides, X^{22}_8 is computed using RK^{23}_6. Then together with X^{23}_6, we can compute RK^{22}_4 with $\#RK^{22}_4 = 16/7$ for every $RK^{23}_{[3,6]}$.

(4) As just mentioned, 16/7 values of X^{22}_9 is computed for every RK^{23}_3. Since $X^{22}_9 = X^{21}_{10}$, we get 16/7 values of X^{21}_{10} for every RK^{23}_3. Besides, Compute $X^{20}_{13} = X^{19}_8$ using $RK^{19}_4 = RK^{22}_0$ and $(\triangle X^{19}_8, \triangle X^{19}_9)$, where we get $\#X^{20}_{13} = 16/7$ for every RK^{22}_0. Then X^{20}_{12} is computed using $RK^{20}_6 = RK^{23}_1 \oplus C^{22}_H$, where $\#X^{20}_{12} = (16/7)^2$ for every $(RK^{23}_{[1,3]}, RK^{22}_0)$. Furthermore, compute X^{22}_{14} using RK^{23}_5, compute X^{22}_{10} using RK^{23}_4, then compute X^{22}_{11} using RK^{22}_5. With the knowledge of X^{20}_{12}, X^{22}_{14} and X^{22}_{11}, we can compute RK^{21}_7 with $\#RK^{21}_7 = (16/7)^2$ for every $(RK^{23}_{[1,3,4,5]}, RK^{22}_{[0,5]})$. □

4 Impossible Differential Cryptanalysis of 23-Round TWINE-80

4.1 Analysis of Suzaki et al.'s Attack on TWINE-80

In the last paragraph of page 9 in the TWINE-80 attack [15], the authors said that *In the key elimination we need to COMPUTE some other subkeys (64 bits in total), which is uniquely determined by the key of Eq. (5). These keys contain* RK^{19}_4, RK^{21}_4, *and* RK^{23}_6 *and they can cause a contradiction with other keys*. Therefore, an attacker has to compute these *other subkeys* using the 80-bit $(\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3)$, and then check whether there is a contradiction. Unfortunately, it seems that this part is omitted in their time complexity formula $2^{50.11+10} \cdot 2^{20} \cdot 22/(23 \cdot 8) = 2^{77.04}$. Because we notice that $2^{50.11+10}$ means the number of plaintext/ciphertext pairs, 2^{20} stands for the time regarding $\mathcal{K}_1$, and $22/(23 \cdot 8)$ is the time regarding $(\mathcal{K}_2, \mathcal{K}_3)$. If the omitted time is considered, the time complexity is supposed to be bigger than exhaustive key search. Take the computation of $RK^{23}_6 = s[RK^{23}_2] \oplus s[RK^{21}_1] \oplus s^{-1}[RK^2_7 \oplus RK^1_0]$ as an example[1], we know that the numbers of RK^{23}_2, RK^{21}_1, RK^2_7, and RK^1_0 that pass the differential path are all 16/7 for one right plaintext/ciphertext pair. Hence the time for checking whether there is a contradiction regarding RK^{23}_6 is $(16/7)^4$. Multiplied by the extra $(16/7)^4$, the time complexity is $2^{77.04} \cdot (16/7)^4 = 2^{81.81}$. It seems that there is a similar problem in the analysis of their attack on TWINE-128.

4.2 Impossible Differential Attack on 23-Round TWINE-80

In this section, we present an impossible differential attack on 23-round TWINE-80 using the impossible differential $(0||\alpha||\tilde{0}^{14}) \overset{14r}{\nrightarrow} (\tilde{0}^7||\beta||\tilde{0}^8)$. This paper uses

1 Reference [15] ignores some known constants C^r_H, C^r_L in their subkey relations.

the same impossible differential as in [15] for TWINE-80, because it leads to the least number of involved round subkeys. The 14-round impossible differential is extended 4 rounds on the top and 5 rounds on the bottom. The extended truncated differential paths are showed in Fig. 2. Making use of Observation 2, eight equations $RK_3^3 = RK_5^1$, $RK_5^3 = RK_1^1$, $RK_1^4 = RK_6^1 \oplus C_H^3$, $RK_4^{19} = RK_0^{22}$, $RK_3^{20} = RK_4^{23}$, $RK_6^{20} = RK_1^{23}$, $RK_1^{21} = RK_5^{23}$ and $RK_5^{21} = RK_3^{23}$ are discovered. Hence the added 9 rounds involve $44 + 68 = 112$ bits round subkeys (see Tables 3 and 4). Therefore, $112 - 80 = 32$ bits subkey information are redundant, which are described in Observation 3.

The idea of attacking is to discard these $\mathcal{K}_{112}$ which pass the truncated differential paths under the condition that $\mathcal{K}_{112}$ is indeed generated from one 80-bit master key according to the key schedule. Denote $\mathcal{K}_0 = (RK^1_{[1,7]}, RK^{23}_{[0,1,7]})$, $\mathcal{K}_1 = (RK^1_{[0,2,3,5,6]}, RK^2_{[2,4,6,7]}, RK^{22}_{[1,3,5]}, RK^{23}_{[2,3,5]})$, $\mathcal{K}_2 = (RK^{21}_{[4,7]}, RK^{22}_{[0,2,4,6]}, RK^{23}_{[4,6]})$. The main steps of our attack are as follows. Firstly, some tables are computed in the precomputation phase for the sake of time and memory balance. Secondly, for every guess of $\mathcal{K}_0$, combine $(\mathcal{K}_1, \mathcal{K}_2)$ which pass the truncated differentials and all the subkeys equations. And then the $\mathcal{K}_1$ in the combined $(\mathcal{K}_1, \mathcal{K}_2)$ is removed from an initialized subkey table. After all the chosen plaintext-ciphertext pairs are utilized, store $\mathcal{K}_0$ and the finally remained $\mathcal{K}_1$. (Notice that once $(\mathcal{K}_0,\mathcal{K}_1)$ is known, $\mathcal{K}_2$ can be computed uniquely according to the subkey equations.) Finally, do trial encryptions for the remaining keys.

Table 3. Subkeys involved in the extended head path of attacking 23-r TWINE-80

Round r	RK_0^r	RK_1^r	RK_2^r	RK_3^r	RK_4^r	RK_5^r	RK_6^r	RK_7^r
Round 1	k_1	k_3	k_4	k_6		k_{14}	k_{15}	k_{16}
Round 2			k_8		k_{17}		$k_{19} \oplus C_L^1$	$k(1,0)$
Round 3				k_{14}		k_3		
Round 4		$k_{15} \oplus C_H^3$						

Table 4. Subkeys involved in the extended tail path of attacking 23-r TWINE-80

Round r	RK_0^r	RK_1^r	RK_2^r	RK_3^r	RK_4^r	RK_5^r	RK_6^r	RK_7^r
Round 19					$RK_4^{19} = RK_0^{22}$			
Round 20				$RK_3^{20} = RK_4^{23}$			$RK_6^{20} = RK_1^{23} \oplus C_H^{22}$	
Round 21		$RK_1^{21} = RK_5^{23}$			RK_4^{21}	$RK_5^{21} = RK_3^{23}$		RK_7^{21}
Round 22	RK_0^{22}	RK_1^{22}	RK_2^{22}	RK_3^{22}	RK_4^{22}	RK_5^{22}	RK_6^{22}	
Round 23	RK_0^{23}	RK_1^{23}	RK_2^{23}	RK_3^{23}	RK_4^{23}	RK_5^{23}	RK_6^{23}	RK_7^{23}

Table 5. KT_i tables

Table	Index	Content[a]
KT_2	$(RK_1^1, RK_0^{23}, RK_4^{23}, RK_6^{22}, RK_7^{21}, RK_{[3,4]}^{22})$	RK_4^2
KT_3	$RK_{[2,5,6]}^{22}$	RK_6^2
KT_4	$(RK_7^1, RK_0^{23}, RK_4^{23}, RK_6^{22}, RK_5^1, RK_4^{21})$	RK_7^{21}
KT_5	$(RK_3^{23}, RK_6^2, RK_5^{22}, RK_5^1, RK_4^{21})$	RK_1^{22}
KT_8	$(RK_7^1, RK_7^{23}, RK_1^{22}, RK_7^{21}, RK_0^{22})$	RK_2^1

[a]The number of possible values of the subkey stored in content is 1 for each index.

Precomputation. Firstly, two tiny tables are precomputed for sbox. A difference distribution table for sbox is computed to facilitate choosing more accurate plaintext-ciphertext pairs using Observation 6. So that $\alpha_1 \in \triangle s[\alpha]$ can be examined by looking up the table. Besides, another tiny table is needed in computing round subkeys, which stores the input pairs of sbox with input and output difference as index. Take the computation of RK_0^1 as an example, suppose a plaintext pair satisfies $\triangle X_1^1 \in \triangle s[\triangle X_0^1]$, looking up this table with index $(\triangle X_0^1, \triangle X_1^1)$ gives the input pair (In1, In2) for sbox, and then $RK_0^1 = In1 \oplus X_0^1$.

Secondly, in order to decrease time complexity at the cost of a little memory in key recovery phase, five tables KT_i (i = 2,3,4,5,8) are precomputed for functions f_i. Hence the computation of f_i can be replaced by one table looking up. A detailed description of these tables is showed in Table 5.

Data Collection. Choose 2^n structures of plaintexts, and each structure contains plaintexts with the following form $(p_0, p_1, \gamma_0, p_2, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, p_3, p_4, p_5, p_6, \gamma_7, p_7)$, where $\gamma_i (i = 0, ..., 7)$ are constants in each structure and $p_i (i = 0, ..., 7)$ take all possible values. As a result, there are 2^{32} plaintexts in each structure and we can get 2^{n+63} plaintext pairs.

Ask for encryptions of the plaintexts in each structure and get the corresponding ciphertexts. The ciphertext is denoted as $(C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15})$. A hash table with index $C_{[0,2,14,15]}$ is built to choose the pairs that satisfy the condition $\triangle C_{[0,2,14,15]} = 0$. The pairs that do not satisfy the condition are discarded. Hence there are $2^{n+63-16} = 2^{n+47}$ pairs remained.

Furthermore, filter the pairs using the plaintext and ciphertext difference relations listed in Observation 6. Therefore, $2^{n+47-18.08} = 2^{n+28.92}$ pairs are finally obtained.

Key Recovery. A detailed key recovery procedure is showed in the following Algorithm 1. It's main steps are as follows. Firstly, 20-bit $\mathcal{K}_0$ is guessed. And then for each plaintext-ciphertext pair, substeps (1.2.1) to (1.2.10) compute some round subkeys that pass the differential path. And then substep (1.2.11) combines all the subkeys according to f_6, f_7, f_1, f_3, f_5, f_4, f_8 and f_2 in sequence and the differential characteristic to obtain 92-bit round subkeys. After these done, the combined 112-bit ($\mathcal{K}_0$, $\mathcal{K}_1$, $\mathcal{K}_2$) pass the differential path and contains

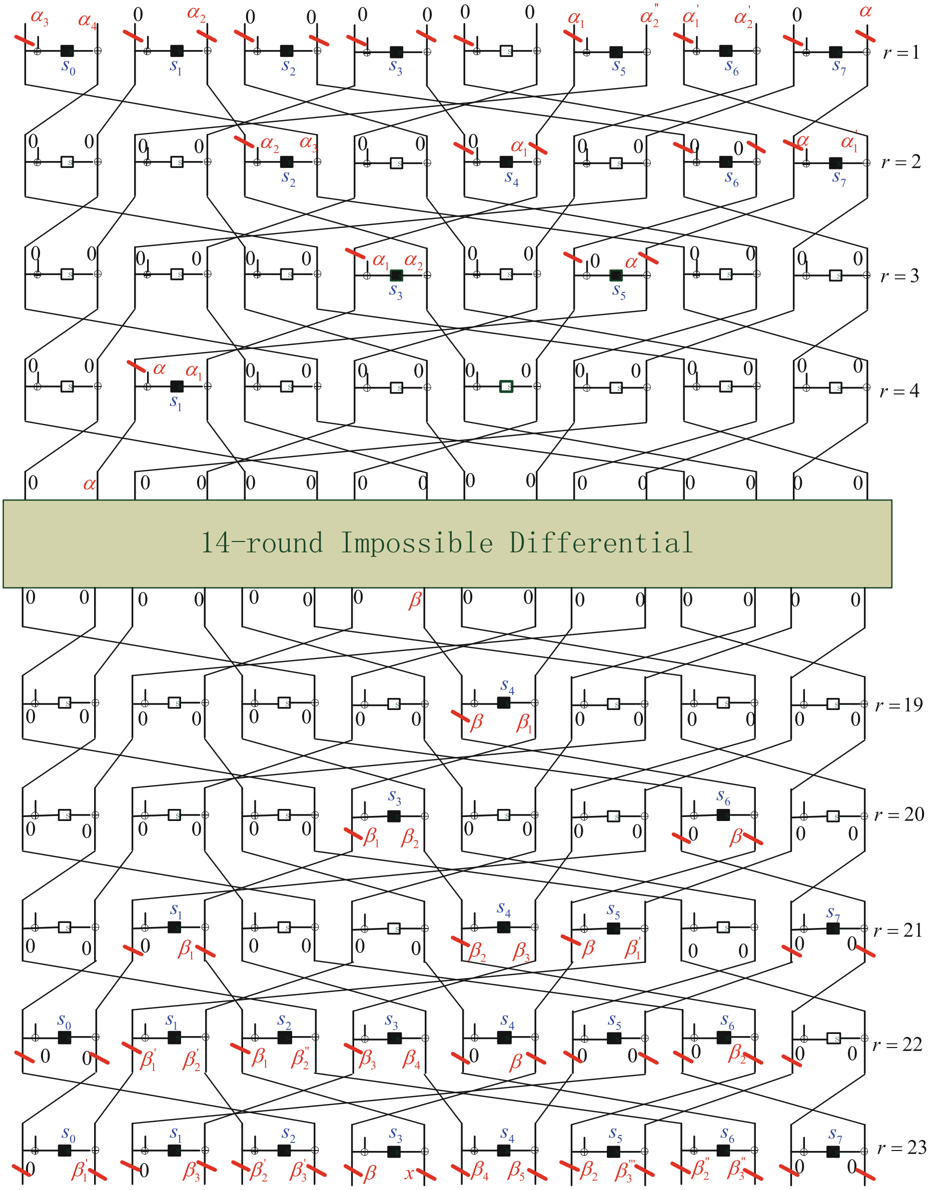

Fig. 2. Attack path for 23-round TWINE-80 (Input (output) values marked with short sloping line and the round subkeys corresponding to black s-box are involved in the attack.)

exactly 80-bit key information which can be expressed by ($\mathcal{K}_0$, $\mathcal{K}_1$). Therefore, the obtained $\mathcal{K}_1$ in the combined 92-bit round Sunkeys are wrong keys and then be discarded in substep (1.2.12). After step 1, the right round subkey is in the remained ones. Hence step 2 aims to recover the right key by trial encryptions. After the candidate master key is computed in substeps (2.1.1) and (2.1.2), a trial encryption is done in substep (2.1.3) to find the right master key.

Algorithm 1. TWINE-80 Key Recovery

Input: chosen plaintext-ciphertext pairs, functions f_i (i = 1,...,8), differential characteristic

Output: right key used in TWINE-80

1: **For** every possible value of $\mathcal{K}_0 = (RK^1_{[1,7]}, RK^{23}_{[0,1,7]})$, **do**

(1.1): Initialize a table Γ of 2^{60} all possible values of $\mathcal{K}_1$;

(1.2): **For** each chosen plaintext-ciphertext pair, **do**

(1.2.1): Compute $X^2_{[4,14]}$ using $RK^1_{[1,7]}$ by partial encryption of plaintext;

(1.2.2): Compute $X^{22}_{[2,6,12]}$ using $RK^{23}_{[0,1,7]}$ by partial decryption of ciphertext;

(1.2.3): Compute $RK^1_{[0,5,6]}$, (RK^{23}_2, X^{22}_0), (RK^{23}_4, X^{22}_{10}), (RK^{23}_5, X^{22}_{14}), (RK^{23}_6, X^{22}_8) using the plaintext-ciphertext pair and differential characteristic;

(1.2.4): Compute RK^2_7 using X^2_{14} and $(\triangle X^2_{14}, \triangle X^2_{15})$;

(1.2.5): Compute RK^{22}_3 using X^{22}_6 and $(\triangle X^{22}_6, \triangle X^{23}_8)$;

/* each 4-bit subkey computed above has $\frac{16}{7}$ values */

(1.2.6): **For** every possible value of RK^{23}_3, **do** /* 2^4 loops */

Compute X^{22}_4 using partial decryption for the ciphertext pair;

If $\triangle X^{22}_4 \in \triangle s[\triangle X^{23}_6]$, $\triangle X^{23}_{10} \in \triangle s[\triangle X^{22}_4]$ and $\triangle X^{23}_{12} \in \triangle s[\triangle X^{22}_4]$ all holds, /* $Pr = (\frac{7}{16})^3$ */

then store (RK^{23}_3, X^{22}_4)

(1.2.7): Compute RK^1_2 using Observation 8, and then store RK^1_2 in Q_0 with index (RK^1_6, RK^2_6);

(1.2.8): Compute RK^{22}_4 using Observation 8, and then store RK^{22}_4 in Q_1 with index $RK^{23}_{[3,6]}$;

(1.2.9): Compute RK^{22}_0 using Observation 8, and then store RK^{22}_0 in Q_2 with index $RK^{23}_{[2,3,4,5]}$;

(1.2.10): Compute RK^{21}_7 using Observation 8, and then store RK^{21}_7 in Q_3 with index $(RK^{23}_{[3,4,5]}, RK^{22}_{[0,5]})$;

(1.2.11): Combine all the involved subkeys using **Algorithm 2** to obtain $(\mathcal{K}_1, \mathcal{K}_2)$ with known $\mathcal{K}_0$;

(1.2.12): Remove $\mathcal{K}_1$ in the combined $(\mathcal{K}_1, \mathcal{K}_2)$ from Γ;

(1.3): Store $\mathcal{K}_0$ and the finally remained $\mathcal{K}_1$ from Γ.

2: After the above steps, suppose there are 2^m ($\mathcal{K}_0$, $\mathcal{K}_1$).

(2.1): **For** each value of ($\mathcal{K}_0$, $\mathcal{K}_1$), **do**

(2.1.1): compute the value of $\mathcal{K}_2$ using f_i (i = 1,...,8);

(2.1.2): and then compute the 9 partial master keys k_2, k_5, k_7, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{18} using ($\mathcal{K}_0$, $\mathcal{K}_1$, $\mathcal{K}_2$);

/* the other 11 partial master keys are known in ($\mathcal{K}_0$, $\mathcal{K}_1$) */

(2.1.3): And then do a trial encryption. If it is correct, then return the right key and abort the loop.

Complexity Analysis. As can be seen from Fig. 2, there are 36 active sboxes. Among these sboxes, 17 sboxes with zero input difference let the corresponding subkey pass the truncated differential with probability 1. Any of the 15 sboxes whose input and output difference appeared in the plaintext/ciphertext difference make the corresponding subkey pass the truncated differential with probability 7^{-1}. The subkey RK^{23}_3 passes the truncated differential with probability $(\frac{7}{16})^3$ as described in substep (1.2.6). After RK^{23}_3 passing, any of the 3 sboxes who has $\triangle X^{22}_4$ as its input(output) difference and nonzero output(input) difference let the corresponding subkey pass the truncated differential with probability 7^{-1}. Therefore, the proportion of removing wrong subkeys for each pair is $7^{-18} \cdot (\frac{7}{16})^3 = 2^{-54.11}$. Hence the number of remained 80-bit subkey after analyzing all $2^{n+28.92}$ pairs is $\sigma = 2^{80}(1 - 2^{-54.11})^{2^{n+28.92}} = 2^m$.

Algorithm 2. Subkeys Combining Procedure

Input: a plaintext-ciphertext pair, $\mathcal{K}_0 = (RK^1_{[1,7]}, RK^{23}_{[0,1,7]})$, tables $\{KT_j\}(j = 2, 3, 4, 5, 8)$, $\{Q_i\}(i = 0, ..., 3)$, and the already computed subkeys $RK^1_{[0,5,6]}$, $RK^{23}_{[2,3,4,5,6]}$, RK^2_7, RK^{22}_3

Output: combined 92-bit subkeys ($\mathcal{K}_1$, $\mathcal{K}_2$) which pass the path and all the subkey equations

1: **For** $(RK^1_6, RK^{23}_{[3,4,5]})$ **do**: /∗ $l_1 = (\frac{16}{7})^3 \cdot (2^4 \cdot (\frac{7}{16})^3) = 2^4$ loops ∗/
  Compute f_6 with the above subkeys;
  If the result is zero, then store $RK = (RK^1_6, RK^{23}_{[3,4,5]})$; /∗ holds with $Pr = 2^{-4}$ ∗/
  otherwise, try next $(RK^1_6, RK^{23}_{[3,4,5]})$;
2: **For** every obtained $(RK^1_0, RK^2_7, RK^{23}_2)$, **do**: /∗ $l_2 = (\frac{16}{7})^3$ loops ∗/
  Compute RK^{23}_6 using f_7; and then compute X^{22}_8 using RK^{23}_6 by partial decryption;
  If $\triangle X^{22}_8 = 0$, then add $(RK^1_0, RK^2_7, RK^{23}_{[2,6]})$ to RK; /∗ $Pr = \frac{16}{7} \cdot 2^{-4}$ ∗/
  otherwise, try next $(RK^1_0, RK^2_7, RK^{23}_2)$;
  compute RK^{22}_2 using the obtained X^{22}_4 and $(\triangle X^{22}_4, \triangle X^{23}_{12})$; /∗ $\frac{16}{7}$ values ∗/
3: **For** every obtained RK^{22}_2, **do**: /∗ $l_3 = \frac{16}{7}$ loops ∗/
  Compute RK^2_2 using f_1; and then compute X^3_{12} using RK^2_2 by partial encryption;
  If $\triangle X^3_{12} = 0$, then add (RK^2_2, RK^{22}_2) to RK; /∗ $Pr = \frac{16}{7} \cdot 2^{-4}$ ∗/
  otherwise, try next RK^{22}_2;
4: **For** every guessed $RK^{22}_{[5,6]}$, **do**: /∗ $l_4 = 2^8$ loops ∗/
  Look up KT_3 to get the value of RK^2_6, then add $(RK^2_6, RK^{22}_{[5,6]})$ to RK;
  Compute X^{21}_8 using RK^{22}_6 and $X^{23}_{[10,15]}$ by partial decryption, and then compute RK^{21}_4;
5: **For** every obtained (RK^1_5, RK^{21}_4), **do**: /∗ $l_5 = (\frac{16}{7})^2$ loops ∗/
  Look up KT_5 to obtain RK^{22}_1, and then compute X^{21}_6 using RK^{22}_1 by partial decryption;
  If $\triangle X^{21}_6 = 0$, then add $(RK^1_5, RK^{21}_4, RK^{22}_1)$ to RK; /∗ $Pr = \frac{16}{7} \cdot 2^{-4}$ ∗/
  otherwise, try next (RK^1_5, RK^{21}_4);
6: Look up KT_4 to get the value for RK^{21}_7;
  For every RK^{22}_0 in Q_2, **do**: /∗ $l_6 = \frac{16}{7}$ loops ∗/
  If RK^{21}_7 appears in Q_3 with index $(RK^{23}_{[1,3,4,5]}, RK^{22}_{[0,5]})$, /∗ $Pr = (\frac{16}{7})^2 \cdot 2^{-4}$ ∗/
  then add (RK^{21}_7, RK^{22}_0) to RK; otherwise, try next RK^{22}_0;
7: Look up KT_8 to get the value for RK^1_2;
  If it appears in Q_0 with index (RK^1_6, RK^2_6), then add RK^1_2 to RK; /∗ $Pr = \frac{16}{7} \cdot 2^{-4}$ ∗/
  otherwise, try next RK^{22}_0;
8: **For** every RK^{22}_4 (from Q_1) and RK^{22}_3, **do**: /∗ $l_8 = (\frac{16}{7})^2$ loops ∗/
  Look up KT_2 to get the value for RK^2_4;
  compute X^3_6 using $RK^3_3 = RK^1_5$ and $(\triangle X^3_6, \triangle X^3_7)$,
  and then X^2_8 is computed using RK^2_4 by partial decryption, and then RK^1_3 is computed using the plaintext pair and X^2_8; and then add $(RK^{22}_{[3,4]}, RK^2_4, RK^1_3)$ to FK.
9: Return the combined $RK = (RK^1_6, RK^{23}_{[3,4,5]}, RK^1_0, RK^2_7, RK^{23}_{[2,6]}, RK^2_2, RK^{22}_2, RK^2_6, RK^{22}_{[5,6]}, RK^1_5, RK^{22}_1, RK^{21}_4, RK^{21}_7, RK^{22}_0, RK^1_2, RK^{22}_{[3,4]}, RK^2_4, RK^1_3)$.

The time complexity of data collection contains: 2^{n+32} to build the hash table, and $2^{n+47}(\frac{15}{16} \cdot \sum_{i=0}^{7}(\frac{7}{16})^i + (\frac{15}{16})^2 \cdot \sum_{i=8}^{14}(\frac{7}{16})^i) = 2^{n+47.737}$ looking up difference distribution table to choose the pairs with required ciphertext/plaintext difference, which is $2^{n+38.628}$ encryptions.

The time complexity of computing the tables in precomputation phase can be omitted compared to the time in key recovery phase.

Notice that the time for substep (1.2.11) dominates the time of step (1.2). Hence the complexity of step (1.2) is $l_1 \cdot (11 + 2^{-4} \cdot l_2 \cdot (9 + \frac{16}{7} + 1 + 7^{-1} \cdot (1 + \frac{16}{7} + l_3 \cdot (7 + 3 + 1 + 7^{-1} \cdot l_4 \cdot (1 + 3 + \frac{16}{7} + l_5 \cdot (1 + 1 + \frac{16}{7} + 1 + 7^{-1} \cdot (1 + l_6 \cdot (1 + (\frac{16}{7})^2 \cdot 2^{-4} \cdot (2 + 7^{-1} \cdot l_8(2 + \frac{16}{7} \cdot 7)))))))))) = 2^{12.73}$ xor, where the computation of f_6, f_7, f_1 needs 11, 9, 7 xor or looking up sbox respectively. (The computation of values l_i (i = 1,...,10) and time estimation for substeps (1.2.7) to (1.2.10) is showed in Appendix B.) Hence the time complexity of step 1 in Key Recovery is $\mathcal{T}_1 = 2^{20+n+28.92+12.73} \cdot \frac{1}{23 \cdot 24}$ 23-round encryptions $= 2^{n+52.54}$ encryptions.

The time complexity of step 2 in Key Recovery is $\mathcal{T}_2 = 2^m$ encryptions, because the time of computing $\mathcal{K}_2$ and nine partial master key (k_2, k_5, k_7, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{18}) is much less than one encryption for each $\mathcal{K}_1$ (see Appendix A). Let $n = 25.85$, $m = 77.72$, then the time complexity of this attack is $\mathcal{T}_1 + \mathcal{T}_2 = 2^{79.09}$ encryptions. Hence, the data complexity is $2^{57.85}$ blocks and the memory complexity is $2^m \cdot 80/64 + 2^{60}/64 = 2^{78.04}$ blocks.

5 Impossible Differential Attack on 24-Round TWINE-128

Attack on 24-round TWINE-128 uses the impossible differential $(\tilde{0}^5||\alpha||\tilde{0}^{10}) \overset{14r}{\nrightarrow} (\tilde{0}^{11}||\beta||\tilde{0}^4)$, because it involves the least number of round subkeys. What's more, subkeys involved in the truncated differential paths have less complicated equations which are showed in Observation 5. We extend 5 rounds on the top and the bottom of the 14-round impossible differential respectively. Table 6 and Table 7 show that the top 5 rounds involve 80-bit subkey information and the bottom 5 rounds involve 84-bit subkey information respectively. Therefore, $80 + 84 - 128 = 36$ bits subkey information are redundant, which are described in Observation 5.

Attacking TWINE-128 is similar to attack on TWINE-80. Suppose 2^n structures are used in this attack, and each structure contains plaintexts with the form $(p_0, p_1, \gamma_0, p_2, p_3, p_4, \gamma_1, \gamma_2, \gamma_3, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11})$, where $\gamma_i (i = 0, ..., 3)$

Table 6. Subkeys involved in the extended head path of attacking TWINE-128

Round r	RK_0^r	RK_1^r	RK_2^r	RK_3^r	RK_4^r	RK_5^r	RK_6^r	RK_7^r
Round 1	k_2	k_3	k_{12}	k_{15}	k_{17}	k_{18}	k_{28}	k_{31}
Round 2	k_6		k_{16}	$k_{19} \oplus C_L^1$	k_{21}	k_{22}	$k(1,0)$	k_0
Round 3	k_{10}	$k_{11} \oplus C_H^2$		$k(23,30) \oplus C_L^2$		k_{26}		
Round 4	k_{14}	$k_{15} \oplus C_H^3$						
Round 5	k_{18}							

Table 7. Subkeys involved in the extended tail path of attacking TWINE-128

Round r	RK_0^r	RK_1^r	RK_2^r	RK_3^r	RK_4^r	RK_5^r	RK_6^r	RK_7^r
Round 20		$RK_1^{20} = RK_5^{24}$						
Round 21	RK_0^{21}		RK_2^{21}					
Round 22	RK_0^{22}		RK_2^{22}	RK_3^{22}			RK_6^{22}	
Round 23	RK_0^{23}	RK_1^{23}	RK_2^{23}	RK_3^{23}	RK_4^{23}	RK_5^{23}		RK_7^{23}
Round 24	RK_0^{24}	RK_1^{24}	RK_2^{24}	RK_3^{24}	RK_4^{24}	RK_5^{24}	RK_6^{24}	RK_7^{24}

are constants and $p_i (i = 0, ..., 11)$ take all possible values in each structure. As a result, there are 2^{48} plaintexts in each structure and 2^{n+95} pairs are obtained. And then select the pairs that satisfy Observation 7, $2^{n+95-16-19.27} = 2^{n+59.73}$ pairs are finally obtained. The complexity of data collection is $2^{n+70.6278}$ encryptions.

Let $\mathcal{K}_0 = (RK^1_{[1,4]}, RK^{24}_{[2,4,5]}), \mathcal{K}_1 = (RK^1_{[0,2,3,5,6,7]}, RK^2_{[0,2,3,4,5,6,7]}, RK^3_{[0,1,3,5]}, RK^4_0, RK^{21}_2, RK^{22}_6, RK^{23}_{[0,1,2,4]}, RK^{24}_{[0,6,7]}), \mathcal{K}_2 = (RK^{21}_0, RK^{22}_{[0,2,3]}, RK^{23}_{[3,5,7]}, RK^{24}_{[1,3]})$, Since the main idea of key recovery is similar to that in TWINE-80, we give the detailed description of key recovery algorithm in Appendix C. Combining $(\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2)$ that pass the truncated differentials and the equations in Observation 5 can be done in $2^{45.48}$ xor operations according to $g_1, g_2, g_3, g_4, g_9, g_7, g_8, g_5, g_6$ in sequence (see Appendix C).

Therefore, the time for filtering wrong keys is $\mathcal{T}_1 = 2^{20+n+59.73+45.48} \cdot \frac{1}{24 \cdot 24}$ 24-round encryptions $= 2^{n+116.04}$ encryptions, followed by $\mathcal{T}_2 = 2^m$ encryptions to do trial encryptions. Since the probability of differential path is Pr$= (7^{-11} \cdot (\frac{7}{16})^3)^2 = 2^{-68.92}$, let $\sigma = 2^{128} \cdot (1 - 2^{-68.92})^{2^{n+59.73}} = 2^m$. Take $n = 10.1$, $m = 125.29$, then the time complexity is $\mathcal{T}_1 + \mathcal{T}_2 = 2^{126.78}$ encryptions. And the memory complexity and data complexity are $2^m \cdot 80/64 + 2^{108}/64 = 2^{125.61}$ blocks and $2^{58.1}$ blocks respectively.

6 Conclusion

This paper gives an impossible differential cryptanalysis of reduced-round TWINE-80 and TWINE-128. In the attacks, we present some key relations, and then an optimal algorithm is proposed to recovery subkeys using these relations, which may be used in other types of attacks. According to the known results, it seems that TWINE currently remains immune to impossible differential attack.

A

The following equations are deduced from the TWINE-80 key schedule.

$$f_1 = RK^2_2 \oplus s[RK^2_7] \oplus RK^{22}_2 \oplus s[RK^{23}_1 \oplus C^{22}_H \oplus C^{19}_L] \oplus C^7_H \oplus C^4_L = 0$$

$$f_2 = RK^{22}_4 \oplus RK^2_4 \oplus C^{14}_H \oplus C^{11}_L \oplus s[C^9_H \oplus C^6_L \oplus RK^{21}_7 \oplus s[RK^{22}_6 \oplus C^{21}_L]] \oplus s[RK^{22}_3 \oplus C^{17}_H \oplus C^{14}_L \oplus s[RK^{23}_0 \oplus C^{12}_H \oplus C^9_L] \oplus s[RK^1_1 \oplus s[RK^{23}_4 \oplus C^{15}_H \oplus C^{12}_L]] \oplus s[RK^{23}_0 \oplus C^{12}_H \oplus C^9_L]] = 0$$

$$f_3 = RK^2_6 \oplus C^4_H \oplus C^1_L \oplus C^{21}_L \oplus RK^{22}_6 \oplus s[RK^{22}_5 \oplus C^{19}_H \oplus C^{16}_L] \oplus s[RK^{22}_2] = 0$$

$$f_4 = RK^{23}_0 \oplus RK^{23}_4 \oplus C^{15}_H \oplus C^{12}_L \oplus s[RK^1_5 \oplus s[C^{13}_H \oplus C^{10}_L \oplus RK^{21}_4]] \oplus C^{12}_H \oplus C^9_L \oplus s^{-1}[RK^1_7 \oplus C^9_H \oplus C^6_L \oplus RK^{21}_7 \oplus s[RK^{22}_6 \oplus C^{21}_L]] = 0$$

$$f_5 = RK^{23}_3 \oplus RK^1_5 \oplus C^{18}_H \oplus C^{15}_L \oplus s[RK^{21}_4 \oplus C^{13}_H \oplus C^{10}_L] \oplus s[RK^{22}_1 \oplus s[RK^2_6 \oplus C^4_H \oplus C^1_L \oplus s[RK^{22}_5 \oplus C^{19}_H \oplus C^{16}_L]] \oplus C^{21}_H \oplus C^{18}_L] = 0$$

$$f_6 = RK^{23}_5 \oplus s[C^{15}_H \oplus C^{12}_L \oplus RK^{23}_4] \oplus C^{20}_H \oplus C^{17}_L \oplus RK^1_1 \oplus s[RK^1_6 \oplus C^3_H \oplus s[C^{18}_H \oplus C^{15}_L \oplus RK^{23}_3]] = 0$$

$$f_7 = RK^{23}_6 \oplus s[C^{20}_H \oplus C^{17}_L \oplus RK^{23}_5] \oplus s[RK^{23}_2] \oplus s^{-1}[RK^2_7 \oplus RK^1_0] \oplus C^5_H \oplus C^2_L \oplus C^{22}_L = 0$$

$$f_8 = s^{-1}[RK^{23}_7 \oplus RK^{22}_0] \oplus s[RK^{21}_7] \oplus s[C^{21}_H \oplus C^{18}_L \oplus RK^{22}_1] \oplus RK^1_2 \oplus C^6_H \oplus C^3_L \oplus s[RK^1_7] = 0$$

As can be seen from the above equations, $\mathcal{K}_2 = (RK^{21}_{[4,7]}, RK^{22}_{[0,2,4,6]}, RK^{23}_{[4,6]})$ can be computed from $(\mathcal{K}_0, \mathcal{K}_1) = (RK^{1}_{[0,1,2,3,5,6,7]}, RK^{2}_{[2,4,6,7]}, RK^{22}_{[1,3,5]}, RK^{23}_{[0,1,2,3,5,7]})$ successively according to equations f_1, f_3, f_5, f_6, f_7, f_4, f_8, f_2 in $87/(23 \cdot 24)$ Xor $= 2^{-2.67}$ encryptions.

$$
\begin{aligned}
k_9 &= s^{-1}[RK^1_7 \oplus C^9_H \oplus C^6_L \oplus RK^{21}_7 \oplus s[RK^{22}_6 \oplus C^{21}_L]] \oplus s[RK^2_2 \oplus s[RK^2_7]] \\
k_{10} &= RK^{22}_3 \oplus C^{17}_H \oplus C^{14}_L \oplus s[RK^{23}_0 \oplus C^{12}_H \oplus C^9_L] \oplus s[RK^1_1 \oplus s[RK^{23}_4 \oplus C^{15}_H \oplus C^{12}_L]] \\
k_5 &= RK^{22}_0 \oplus C^{11}_H \oplus C^8_L \oplus s[RK^1_2 \oplus s[RK^1_7]] \oplus s[RK^2_4 \oplus s[RK^1_7 \oplus s[k_9 \oplus s[RK^2_2 \oplus s[RK^2_7]]]]] \\
k_{11} &= RK^{23}_1 \oplus C^2_H \oplus C^{22}_H \oplus C^{19}_L \oplus s[RK^{22}_3 \oplus C^{17}_H \oplus C^{14}_L] \oplus s[s^{-1}[RK^2_7 \oplus RK^1_0] \\
&\quad \oplus C^5_H \oplus C^2_L \oplus s[RK^{23}_5 \oplus C^{20}_H \oplus C^{17}_L]] \\
k_{18} &= RK^{22}_5 \oplus C^{19}_H \oplus C^{16}_L \oplus s[RK^{22}_4 \oplus C^{14}_H \oplus C^{11}_L] \oplus s[k_{11} \oplus C^2_H \oplus s[RK^{22}_3 \oplus C^{17}_H \oplus C^{14}_L]] \\
k_7 &= RK^{22}_1 \oplus C^1_H \oplus C^{21}_H \oplus C^{18}_L \oplus s[RK^1_3 \oplus s[RK^{22}_0 \oplus C^{11}_H \oplus C^8_L] \oplus s[k_{18} \oplus s[RK^{22}_4 \\
&\quad \oplus C^{14}_H \oplus C^{11}_L]]] \oplus s[RK^2_6 \oplus C^4_H \oplus s[RK^{22}_5 \oplus C^{19}_H \oplus C^{16}_L]] \\
k_2 &= RK^{23}_4 \oplus C^{15}_H \oplus C^{12}_L \oplus s[RK^2_7 \oplus s[RK^{21}_4 \oplus C^{13}_H \oplus C^{10}_L \oplus s[RK^1_3 \oplus s[RK^{22}_0 \\
&\quad \oplus C^{11}_H \oplus C^8_L]]]] \oplus s[RK^1_5 \oplus s[RK^{21}_4 \oplus C^{13}_H \oplus C^{10}_L]] \\
k_{12} &= RK^{23}_2 \oplus C^8_H \oplus C^5_L \oplus s[k_5 \oplus s[RK^1_2 \oplus s[RK^1_7]]] \oplus s[RK^1_6 \oplus C^3_H \oplus s[RK^{23}_3 \\
&\quad \oplus C^{18}_H \oplus C^{15}_L] \oplus s[RK^1_2 \oplus C^6_H \oplus C^3_L \oplus s[RK^1_7] \oplus s[RK^{22}_1 \oplus C^{21}_H \oplus C^{18}_L]]] \\
k_{13} &= RK^{21}_4 \oplus C^{13}_H \oplus C^{10}_L \oplus s[k_{12} \oplus s[k_5 \oplus s[RK^1_2 \oplus s[RK^1_7]]]] \oplus s[RK^1_3 \oplus s[RK^{22}_0 \oplus C^{11}_H \oplus C^8_L]]
\end{aligned}
$$

As can be seen from the above equations, the nine partial master key $(k2, k5, k7, k9, k10, k11, k12, k13, k18)$ can be computed in $114/(23 \cdot 24)$ encryptions $= 2^{-2.276}$ encryptions.

The following equations are deduced from the TWINE-128 key schedule.

$$
\begin{aligned}
g_1 &= RK^{22}_3 \oplus s[RK^{23}_5] \oplus C^{21}_L \oplus s^{-1}[RK^{22}_2 \oplus RK^1_1] = 0 \\
g_2 &= RK^{21}_0 \oplus s[RK^{24}_6 \oplus s[RK^{24}_7]] \oplus C^{12}_H \oplus C^9_L \oplus RK^2_2 \oplus s[RK^1_6] = 0 \\
g_3 &= s^{-1}[RK^3_1 \oplus RK^{24}_2] \oplus s[RK^{23}_7 \oplus s[RK^{22}_2]] \oplus RK^3_0 \oplus s[RK^{23}_5 \oplus C^{18}_H \oplus C^{15}_L \oplus s[RK^{21}_0]] = 0 \\
g_4 &= C^{20}_H \oplus C^{17}_L \oplus s[RK^{23}_0] \oplus s^{-1}[s^{-1}[RK^{24}_2 \oplus RK^3_1] \oplus C^{23}_L \oplus RK^{24}_3] \oplus s^{-1}[RK^1_5 \oplus s^{-1}[RK^{22}_6 \\
&\quad \oplus C^4_H \oplus RK^2_3] \oplus s[RK^{21}_2]] = 0 \\
g_5 &= RK^1_0 \oplus s^{-1}[RK^1_1 \oplus RK^{22}_2] \oplus s[RK^4_0 \oplus s[RK^{24}_5 \oplus C^{19}_H \oplus C^{16}_L \oplus s[RK^{22}_0]]] \oplus s[C^{16}_H \oplus C^{13}_L \oplus s[RK^{23}_4] \\
&\quad \oplus s^{-1}[RK^{23}_1 \oplus C^{22}_H \oplus C^{19}_L \oplus s^{-1}[RK^{24}_7 \oplus RK^3_5 \oplus s[RK^{23}_2]]]] = 0 \\
g_6 &= RK^2_4 \oplus s[RK^{22}_0 \oplus C^{13}_H \oplus C^{10}_L \oplus s[C^7_H \oplus C^4_L \oplus RK^1_7 \oplus s[RK^{23}_2 \oplus s[RK^{23}_3 \oplus C^{22}_L \oplus s[RK^{24}_5]]]]] \\
&\quad \oplus s[RK^1_0 \oplus s[C^{16}_H \oplus C^{13}_L \oplus s[RK^{23}_4] \oplus s^{-1}[RK^{23}_1 \oplus C^{22}_H \oplus C^{19}_L \oplus s^{-1}[RK^{24}_7 \oplus RK^3_5 \oplus s[RK^{23}_2]]]]] \\
&\quad \oplus s^{-1}[RK^{23}_7 \oplus RK^2_5 \oplus s[RK^{22}_2]] = 0 \\
g_7 &= C^{22}_L \oplus RK^2_0 \oplus RK^{23}_3 \oplus s[RK^{24}_5] \oplus s[s^{-1}[RK^{22}_6 \oplus C^4_H \oplus RK^2_3] \oplus s[RK^{21}_2]] \oplus s[s^{-1}[RK^{23}_0 \oplus C^{14}_H \\
&\quad \oplus C^{11}_L \oplus s^{-1}[RK^{24}_4 \oplus C^{11}_H \oplus C^8_L \oplus RK^1_2 \oplus s[C^5_H \oplus RK^3_3]] \oplus s[C^8_H \oplus C^5_L \oplus RK^2_7 \oplus s[RK^3_1]]] \\
&\quad \oplus s[RK^1_4 \oplus s[RK^2_2 \oplus s[RK^1_6]]]] = 0
\end{aligned}
$$

$$\begin{aligned} g_8 = {} & s^{-1}[RK_5^3 \oplus RK_7^{24} \oplus s[RK_2^{23}]] \oplus s^{-1}[RK_5^{24} \oplus C_H^{19} \oplus C_L^{16} \oplus s^{-1}[RK_6^2 \oplus C_H^{16} \oplus C_L^{13} \oplus s[RK_4^{23}] \\ & \oplus s^{-1}[RK_1^{23} \oplus C_H^{22} \oplus C_L^{19} \oplus s^{-1}[RK_7^{24} \oplus RK_5^3 \oplus s[RK_2^{23}]]]] \oplus s[RK_0^{22}]] \oplus s[RK_0^2 \oplus s[\\ & s^{-1}[RK_0^{23} \oplus C_H^{14} \oplus C_L^{11} \oplus s^{-1}[RK_4^{24} \oplus C_H^{11} \oplus C_L^8 \oplus RK_2^1 \oplus s[C_H^5 \oplus RK_3^3]] \oplus s[C_H^8 \oplus C_L^5 \\ & \oplus RK_7^2 \oplus s[RK_1^3]]] \oplus s[RK_4^1 \oplus s[RK_2^2 \oplus s[RK_6^1]]]]] = 0 \\ g_9 = {} & s^{-1}[RK_4^1 \oplus s[RK_2^2 \oplus s[RK_6^1]] \oplus s^{-1}[RK_5^1 \oplus s^{-1}[RK_6^{22} \oplus C_H^4 \oplus RK_3^2] \oplus s[RK_2^{21}]]] \oplus s[RK_0^3 \oplus s[\\ & RK_5^{23} \oplus C_H^{18} \oplus C_L^{15} \oplus s[C_H^{12} \oplus C_L^9 \oplus RK_0^{21} \oplus C_H^{12} \oplus C_L^9]]] \oplus s[C_H^{17} \oplus C_L^{14} \oplus s^{-1}[RK_0^{23} \oplus C_H^{14} \oplus C_L^{11} \\ & \oplus s^{-1}[RK_4^{24} \oplus C_H^{11} \oplus C_L^8 \oplus RK_2^1 \oplus s[C_H^5 \oplus RK_3^3]] \oplus s[C_H^8 \oplus C_L^5 \oplus RK_7^2 \oplus s[RK_1^3]]] \\ & \oplus s[RK_4^1 \oplus s[RK_2^2 \oplus s[RK_6^1]]] \oplus s[RK_4^{24}]] \oplus C_H^{23} \oplus C_L^{20} \oplus RK_1^{24} = 0 \end{aligned}$$

B

It is obvious that the value of $\#RK_0^1$, $\#RK_5^1$, $\#RK_6^1$, $\#RK_2^{23}$, $\#RK_4^{23}$, $\#RK_5^{23}$, $\#RK_6^{23}$, $\#RK_1^{22}$ are all $\frac{16}{7}$ for each plaintext-ciphertext pair when these subkeys pass the differential path with known RK_0^{23}. Besides, RK_3^{23} passes the truncated differential with probability $(\frac{7}{16})^3$, so $\#RK_3^{23} = 2^4 \cdot (\frac{7}{16})^3$ for each accurate plaintext-ciphertext pair. Furthermore, once RK_7^1 that pass the differential path is known, $\#RK_7^2 = \frac{16}{7}$; once RK_1^1 that pass the differential path is known, $\#RK_2^2 = \frac{16}{7}$; once RK_3^{23} that pass the differential path is known, $\#RK_2^{22} = \frac{16}{7}$; once RK_6^{22} that pass the differential path is known, $\#RK_4^{21} = \frac{16}{7}$ with the known RK_7^{23}; once RK_1^{23} that pass the differential path is known, $\#RK_3^{22} = \frac{16}{7}$.

Therefore, it is easy to compute the value of loops l_i with the above knowledge and Observation 8.

The following is a time estimation for substep (1.2.7) to substep (1.2.10) in key recovery algorithm.

As showed in the proof of Observation 8, the computation of RK_2^1 for each (RK_6^1, RK_6^2) can be done in much less than one encryption. Therefore, $\#RK_6^1 = \frac{16}{7}$ and $\#RK_6^2 = 2^4$ indicate that the time for computing RK_2^1 is less than $\frac{16}{7} \cdot 2^4$ encryptions.

Similarly, since $\#RK_3^{23} = 2^4 \cdot (\frac{7}{16})^3$, $\#RK_6^{23} = \frac{16}{7}$, the time for computing RK_4^{22} is less than $2^4 \cdot (\frac{7}{16})^2$ encryptions. Because $\#RK_2^{23}$, $\#RK_4^{23}$ and $\#RK_5^{23}$ are all $\frac{16}{7}$, and $\#RK_3^{23} = 2^4 \cdot (\frac{7}{16})^3$, the time for computing RK_0^{22} is less than 2^4 encryptions. Known from Observation 8, the number of values of RK_0^{22} is $\frac{16}{7}$ for each $RK_{[2,3,4,5]}^{23}$. Hence the time for computing RK_7^{21} is less than $\frac{16}{7} \cdot 2^4$ encryptions.

C

This appendix gives a detailed description of the Key Recovery algorithm for TWINE-128. Before introducing the algorithm, an observation similar to Observation 8 used in attacking TWINE-80 is given, followed by some precomputed tables for g_i functions.

Observation C.1. *For a plaintext-ciphertext pair satisfying the input-output difference relations in Observation 7, the following can be deduced according to the differential path in attacking TWINE-128.*

(1) Given $RK_2^{21}, RK_3^{22}, RK_0^{24}, RK_6^{24}$ that pass the differential path, then $\frac{16}{7}$ values of RK_1^{23} on average can pass the path and be computed;
(2) Given $RK_{[1,5,7]}^{24}, RK_3^{23}, RK_2^{22}, RK_0^{21}$ that pass the differential path, then $(\frac{16}{7})^2$ values of RK_0^{22} on average can pass the path and be computed; and then if RK_3^{24} is also known, then $\frac{16}{7}$ values of RK_2^{23} on average can pass the path and be computed;
(3) Given $RK_0^1, RK_0^2, RK_0^3, RK_5^1, RK_1^3$ that pass the differential path, then $(\frac{16}{7})^2$ values of RK_0^4 on average can pass the path and be computed;
(4) Given RK_6^1, RK_1^3 that pass the differential path, then $\frac{16}{7}$ values of RK_5^2 on average can pass the path and be computed;
(5) Given $RK_2^1, RK_7^1, RK_6^2, RK_5^3$ that pass the differential path, then $\frac{16}{7}$ values of RK_3^1 on average can pass the path and be computed; and then if RK_3^3 is also known, then $(\frac{16}{7})^2$ values of RK_4^2 on average can pass the path and be computed;

Proof. Making use of the differential path and the equations $RK_1^4 = RK_3^1$, $RK_0^5 = RK_5^1$ and $RK_1^{20} = RK_5^{24}$, it is easy to prove the above observation similarly to the proof in Observation 8.

The following tables $KT_i^{'} (i = 3, ..., 9)$ are precomputed for equations g_i respectively.

Table	Index	Content
$KT_3^{'}$	$(RK_{[0,1]}^3, RK_0^{21}, RK_2^{22}, RK_5^{23}, RK_2^{24})$	RK_7^{23}
$KT_4^{'}$	$(RK_5^1, RK_3^2, RK_1^3, RK_6^{22}, RK_0^{23}, RK_{[2,3]}^{24})$	RK_2^{21}
$KT_5^{'}$	$(RK_{[0,1]}^1, RK_5^3, RK_{[0,2]}^{22}, RK_{[1,2,4]}^{23}, RK_{[5,7]}^{24})$	RK_0^4
$KT_6^{'}$	$(RK_{[0,7]}^1, RK_{[4,5]}^2, RK_5^3, RK_{[0,2]}^{22}, RK_{[1,2,3,4,7]}^{23}, RK_{[5,7]}^{24})$	RK_4^2
$KT_7^{'}$	$(RK_{[2,4,6]}^1, RK_{[0,2,3,7]}^2, RK_{[1,3]}^3, RK_2^{21}, RK_6^{22}, RK_{[0,3]}^{23}, RK_{[4,5]}^{24})$	RK_3^{23}
$KT_8^{'}$	$(RK_{[2,4,6]}^1, RK_{[0,2,6,7]}^2, RK_{[1,3,5]}^3, RK_0^{22}, RK_{[0,1,2,4]}^{23}, RK_{[4,5,7]}^{24})$	RK_5^3
$KT_9^{'}$	$(RK_{[2,4,5,6]}^1, RK_{[2,3,7]}^2, RK_{[0,1,3]}^3, RK_{[0,2]}^{21}, RK_6^{22}, RK_{[0,5]}^{23}, RK_{[1,4]}^{24})$	RK_3^3

As can be seen from Algorithm C.2, the time for combining all the subkeys involved in attacking TWINE-128 is $l_1 \cdot (5 + l_2 \cdot (13 + l_3 \cdot (1 + 3 + 1 + \frac{16}{7} + l_4 \cdot (1 + l_{5.1} \cdot (1 + \frac{16}{7} + l_{5.2} \cdot (1 + l_6 \cdot (1 + 1 + \frac{16}{7} + 1 + l_{7.1} \cdot (1 + l_{7.2} \cdot (1 + l_8 \cdot (2 + (\frac{16}{7})^2 \cdot 2^{-4} \cdot l_9 \cdot 2))))))))) = 2^{45.48}$ xor $= 2^{36.31}$ 24-round encryptions.

Algorithm C.1. TWINE-128 Key Recovery

Input: chosen plaintext-ciphertext pairs, functions g_i $(i = 1, ..., 9)$, differential characteristic

Output: right key used in TWINE-128

1: **For** every possible value of $\mathcal{K}_0 = (RK^1_{[1,4]}, RK^{24}_{[2,4,5]})$, **do**

(1.1): Initialize a table Γ of 2^{108} all possible values of $\mathcal{K}_1$;

(1.2): **For** each chosen plaintext-ciphertext pair, **do**

(1.2.1): Compute $X^2_{[4,6]}$ using $RK^1_{[1,4]}$ by partial encryption of plaintext;

(1.2.2): Compute $X^{23}_{[0,10,14]}$ using $RK^{24}_{[2,4,5]}$ by partial decryption of ciphertext;

(1.2.3): Compute (RK^1_0, X^2_0), (RK^1_2, X^2_{12}), (RK^1_5, X^2_2), (RK^1_6, X^2_{10}), (RK^{24}_0, X^{23}_2), (RK^{24}_1, X^{23}_6), (RK^{24}_3, X^{23}_4), (RK^{24}_7, X^{23}_{12}) using the plaintext-ciphertext pair and differential characteristic;

(1.2.4): Compute RK^2_2 using X^2_4 and $(\triangle X^2_4, \triangle X^2_5)$; Compute RK^2_3 using X^2_6 and $(\triangle X^2_6, \triangle X^2_7)$;

(1.2.5): Compute RK^{23}_0 using X^{23}_0 and $(\triangle X^{23}_0, \triangle X^{23}_1)$; Compute RK^{23}_5 using X^{23}_{10} and $(\triangle X^{23}_{10}, \triangle X^{23}_{11})$;

/* each 4-bit subkey computed above has $\frac{16}{7}$ values */

(1.2.6): **For** every possible value of RK^1_7, **do** /* 2^4 loops */

Compute X^2_{14};

If $\triangle X^2_{15} \in \triangle s[\triangle X^2_{14}]$, $\triangle X^1_{10} \in \triangle s[\triangle X^2_{14}]$ and $\triangle X^2_{14} \in \triangle s[\triangle X^1_{14}]$ all holds, /* $Pr = (\frac{7}{16})^3$ */

then store (RK^1_7, X^2_{14});

(1.2.7): **For** every possible value of RK^{24}_6, **do** /* 2^4 loops */

Compute X^{23}_8;

If $\triangle X^{23}_8 \in \triangle s[\triangle X^{24}_{12}]$, $\triangle X^{24}_6 \in \triangle s[\triangle X^{23}_8]$ and $\triangle X^{24}_{14} \in \triangle s[\triangle X^{23}_8]$ all holds, /* $Pr = (\frac{7}{16})^3$ */

then store (RK^{24}_6, X^{23}_8);

(1.2.8): Compute RK^{23}_1 using Observation C.1, and then store it in Q_0 with index $(RK^{21}_2, RK^{22}_3, RK^{24}_0, RK^{24}_6)$;

(1.2.9): Compute (RK^{22}_0, RK^{23}_2) using Observation C.1, and then store it in Q_1 with index $(RK^{24}_{[1,3,5,7]}, RK^{23}_3, RK^{22}_2, RK^{21}_0)$;

(1.2.10): Compute RK^4_0 using Observation C.1, and then store it in Q_2 with index $(RK^1_0, RK^2_0, RK^3_0, RK^1_5, RK^3_1)$;

(1.2.11): Compute RK^2_5 using Observation C.1, and then store it in Q_3 with index (RK^1_6, RK^3_1);

(1.2.12): Compute (RK^2_4, RK^1_3) using Observation C.1, and then store it in Q_4 with index $(RK^1_2, RK^1_7, RK^2_6, RK^3_{[3,5]})$;

(1.2.13): Combine all the involved subkeys using **Algorithm C.2** to obtain $(\mathcal{K}_1, \mathcal{K}_2)$ with known $\mathcal{K}_0$;

(1.2.14): Remove $\mathcal{K}_1$ in the combined $(\mathcal{K}_1, \mathcal{K}_2)$ from Γ;

(1.3): Store $\mathcal{K}_0$ and the finally remained $\mathcal{K}_1$ from Γ.

2: After the above steps, suppose there are 2^m $(\mathcal{K}_0, \mathcal{K}_1)$.

(2.1): **For** each value of $(\mathcal{K}_0, \mathcal{K}_1)$, **do**

(2.1.1): compute the value of $\mathcal{K}_2$ using g_i $(i = 1,...,9)$;

(2.1.2): and then compute the 12 partial master keys k_4, k_5, k_7, k_8, k_9, k_{13}, k_{20}, k_{23}, k_{24}, k_{25}, k_{27}, k_{29} using $(\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2)$; /* the other 20 partial master keys are known in $(\mathcal{K}_0, \mathcal{K}_1)$ */

(2.1.3): And then do a trial encryption. If it is correct, then return the right key and abort the loop.

Algorithm C.2. Subkeys Combining Procedure for TWINE-128

Input: a plaintext-ciphertext pair, $\mathcal{K}_0 = (RK^1_{[1,4]}, RK^{24}_{[2,4,5]})$, functions g_i $(i = 1, 2)$, tables $KT_i^{'}$ $(i = 3, ..., 9)$, $\{Q_i\}(i = 0, ..., 4)$, and the already computed subkeys $RK^1_{[0,2,5,6,7]}$, $RK^{24}_{[0,1,3,6,7]}$, $RK^2_{[2,3]}$, $RK^{23}_{[0,5]}$

Output: combined 144-bit subkeys $(\mathcal{K}_1, \mathcal{K}_2)$ which pass the path and all the subkey equations

1: **For** every (RK^{23}_5, RK^{22}_2) **do**: /* $l_1 = \frac{16}{7} \cdot 2^4$ loops */

Compute RK^{22}_3 using g_1; and then store $RK = (RK^{23}_5, RK^{22}_{[2,3]})$;

2: **For** every $(RK^1_6, RK^2_2, RK^{24}_{[6,7]})$, **do**: /* $l_2 = (\frac{16}{7})^3 \cdot (2^4 \cdot (\frac{7}{16})^3) = 2^4$ loops */

Compute RK^{21}_0 using g_2; and then add $(RK^1_6, RK^2_2, RK^{24}_{[6,7]}, RK^{21}_0)$ to RK;

3: **For** every $RK^3_{[0,1]}$, **do**: /* $l_3 = 2^8$ loops */

Look up $KT_3^{'}$ to get the value of RK^{23}_7; and then add $(RK^3_{[0,1]}, RK^{23}_7)$ to RK;

Compute X^{23}_{15} using RK^{23}_7, and then compute RK^{22}_6 using X^{23}_{15} and $(\triangle X^{23}_{15}, \triangle X^{23}_{10})$;

4: **For** every $(RK^1_5, RK^2_3, RK^{22}_6, RK^{23}_0, RK^{24}_3)$, **do**: /* $l_4 = (\frac{16}{7})^5$ loops */

Look up $KT_4^{'}$ to get the value of RK^{21}_2, then add $(RK^1_5, RK^2_3, RK^{22}_6, RK^{23}_0, RK^{24}_3, RK^{21}_2)$ to RK;

5: **For** every (RK^1_7, X^2_{14}), **do**: /* $l_{5.1} = 2^4 \cdot (\frac{7}{16})^3$ loops */

Compute RK^2_7 using X^2_{14} and $(\triangle X^2_{14}, \triangle X^2_{15})$;

For every $(RK^1_2, RK^2_7, RK^{24}_1)$, **do**: /* $l_{5.2} = (\frac{16}{7})^3$ loops */

Look up $KT_9^{'}$ to obtain RK^3_3, and then add $(RK^1_{[2,7]}, RK^2_7, RK^{24}_1, RK^3_3)$ to RK;

6: **For** every RK^2_0, **do**: /* $l_6 = 2^4$ loops */

Look up $KT_7^{'}$ to get the value for RK^{23}_3; and then add (RK^2_0, RK^{23}_3) to RK;

7: Compute RK^{23}_4 using X^{23}_8 and $(\triangle X^{23}_8, \triangle X^{23}_9)$; Look up Q_1 to obtain (RK^{22}_0, RK^{23}_2);

For every RK^{24}_0, **do**: /* $l_{7.1} = \frac{16}{7}$ loops */

Look up Q_0 to obtain RK^{23}_1;

For every $(RK^{23}_{[1,2,4]}, RK^{22}_0, RK^2_6)$, **do**: /* $l_{7.2} = (\frac{16}{7})^5 \cdot 2^4$ loops */

Look up $KT_8^{'}$ to get RK^3_5; and then add $(RK^{24}_0, RK^{23}_{[1,2,4]}, RK^{22}_0, RK^2_6, RK^3_5)$ to RK;

8: **For** every RK^1_0, **do**: /* $l_8 = \frac{16}{7}$ loops */

Look up $KT_5^{'}$ to get the value for RK^4_0;

If it appears in Q_2 with index $(RK^1_0, RK^2_0, RK^3_0, RK^1_5, RK^3_1)$, /* $Pr = (\frac{16}{7})^2 \cdot 2^{-4}$ */

then add (RK^1_0, RK^4_0) to RK; otherwise, try next RK^1_0;

9: **For** every RK^2_5 from Q_3, **do**: /* $l_9 = \frac{16}{7}$ loops */

Look up $KT_6^{'}$ to get the value for RK^2_4;

If it appears in Q_4 with index $(RK^1_2, RK^1_7, RK^2_6, RK^3_5)$, /* $Pr = (\frac{16}{7})^2 \cdot 2^{-4}$ */

then add RK^2_5, RK^2_4 together with RK^1_3 (from Q_3) to RK; otherwise, try next RK^2_5;

10: Return the combined $RK = (RK^{23}_5, RK^{22}_{[2,3]}, RK^1_6, RK^2_2, RK^{24}_{[6,7]}, RK^{21}_0, RK^3_{[0,1]}, RK^{23}_7, RK^1_5, RK^2_3, RK^{22}_6, RK^{23}_0,$ $RK^{24}_3, RK^{21}_2, RK^1_{[2,7]}, RK^2_7, RK^{24}_1, RK^3_3, RK^2_0, RK^{23}_3, RK^{24}_0, RK^{23}_{[1,2,4]}, RK^{22}_0, RK^2_6, RK^3_5, RK^1_0, RK^4_0, RK^2_5, RK^2_4, RK^1_3)$.

D

Algorithm D.1. Algorithm 2.3: TWINE.KeySchedule-80($(k_0, ..., k_{19})$, $RK^r_{[0,...,7]}$) in [15]

$(WK_0||WK_1||...||WK_{18}||WK_{19}) \leftarrow (k_0, ..., k_{19})$

for r $\leftarrow$ 1 to 35 **do**

$RK^r_{[0,...,7]} \leftarrow (WK_1||WK_3||WK_4||WK_6||WK_{13}||WK_{14}||WK_{15}||WK_{16})$

$WK_1 \leftarrow WK_1 \oplus s[WK_0], WK_4 \leftarrow WK_4 \oplus s[WK_{16}],$

$WK_7 \leftarrow WK_7 \oplus C^r_H, WK_{19} \leftarrow WK_{19} \oplus C^r_L,$

$(WK_0||WK_1||WK_2||WK_3) \leftarrow (WK_1||WK_2||WK_3||WK_0)$

$(WK_0||...||WK_{19}) \leftarrow (WK_4||...||WK_{19}||WK_0||WK_1||WK_2||WK_3)$

end for

$RK^{36}_{[0,...,7]} \leftarrow (WK_1||WK_3||WK_4||WK_6||WK_{13}||WK_{14}||WK_{15}||WK_{16})$

Algorithm D.2. Algorithm A.1: TWINE.KeySchedule-128($(k_0, ..., k_{31})$, $RK^r_{[0,...,7]}$) in [15]

$(WK_0||WK_1||...||WK_{18}||WK_{31}) \leftarrow (k_0, ..., k_{31})$
for r $\leftarrow$ 1 to 35 **do**
  $RK^r_{[0,...,7]} \leftarrow (WK_2||WK_3||WK_{12}||WK_{15}||WK_{17}||WK_{18}||WK_{28}||WK_{31})$
  $WK_1 \leftarrow WK_1 \oplus s[WK_0]$, $WK_4 \leftarrow WK_4 \oplus s[WK_{16}]$, $WK_{23} \leftarrow WK_{23} \oplus s[WK_{30}]$,
  $WK_7 \leftarrow WK_7 \oplus C^r_H$, $WK_{19} \leftarrow WK_{19} \oplus C^r_L$,
  $(WK_0||WK_1||WK_2||WK_3) \leftarrow (WK_1||WK_2||WK_3||WK_0)$
  $(WK_0||...||WK_{31}) \leftarrow (WK_4||...||WK_{31}||WK_0||WK_1||WK_2||WK_3)$
end for
$RK^{36}_{[0,...,7]} \leftarrow (WK_2||WK_3||WK_{12}||WK_{15}||WK_{17}||WK_{18}||WK_{28}||WK_{31})$

Table D.1. Subkeys of round 1–5 in TWINE-80

Round r	RK^r_0	RK^r_1	RK^r_2	RK^r_3	RK^r_4	RK^r_5	RK^r_6	RK^r_7
Round 1	k_1	k_3	k_4	k_6	k_{13}	k_{14}	k_{15}	k_{16}
Round 2	k_5	$k_7 \oplus C^1_H$	k_8	k_{10}	k_{17}	k_{18}	$k_{19} \oplus C^1_L$	$k(1,0)$
Round 3	k_9	$k_{11} \oplus C^2_H$	k_{12}	k_{14}	k_2	k_3	$k_0 \oplus C^2_L$	$k(5,(4,16))$
Round 4	k_{13}	$k_{15} \oplus C^3_H$	k_{16}	k_{18}	k_6	$k_7 \oplus C^1_H$	$k(4,16) \oplus C^3_L$	$k(9,(8,(1,0)))$
Round 5	k_{17}	$k_{19} \oplus C^4_H \oplus C^1_L$	$k(1,0)$	k_3	k_{10}	$k_{11} \oplus C^2_H$	$k(8,(1,0)) \oplus C^4_L$	$k(13,(12,(5,(4,16))))$

Table D.2. Subkeys of round 1–7 in TWINE-128

Round r	RK^r_0	RK^r_1	RK^r_2	RK^r_3	RK^r_4	RK^r_5	RK^r_6	RK^r_7
Round 1	k_2	k_3	k_{12}	k_{15}	k_{17}	k_{18}	k_{28}	k_{31}
Round 2	k_6	$k_7 \oplus C^1_H$	k_{16}	$k_{19} \oplus C^1_L$	k_{21}	k_{22}	$k(1,0)$	k_0
Round 3	k_{10}	$k_{11} \oplus C^2_H$	k_{20}	$k(23,30) \oplus C^2_L$	k_{25}	k_{26}	$k(5,(4,16))$	$k(4,16)$
Round 4	k_{14}	$k_{15} \oplus C^3_H$	k_{24}	$k(27,3) \oplus C^3_L$	k_{29}	k_{30}	$k(9,(8,20))$	$k(8,20)$
Round 5	k_{18}	$k_{19} \oplus C^4_H \oplus C^1_L$	k_{28}	$k_{31} \oplus s[k_7 \oplus C^1_H] \oplus C^4_L$	k_2	k_3	$k(13,(12,24))$	$k(12,24)$
Round 6	k_{22}	$k(23,30) \oplus C^2_L \oplus C^5_H$	$k(1,0)$	$k_0 \oplus s[k_{11} \oplus C^2_H] \oplus C^5_L$	k_6	$k_7 \oplus C^1_H$	$k(17,(16,28))$	$k(16,28)$
Round 7	k_{26}	$k(27,3) \oplus C^3_L \oplus C^6_H$	$k(5,(4,16))$	$k(4,16) \oplus s[k_{15} \oplus C^3_H] \oplus C^6_L$	k_{10}	$k_{11} \oplus C^2_H$	$k(21,(20,(1,0)))$	$k(20,(1,0))$

References

1. Bogdanov, A., Boura, C., Rijmen, V., Wang, M., Wen, L., Zhao, J.: Key difference invariant bias in block ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 357–376. Springer, Heidelberg (2013)
2. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer, Heidelberg (1999)
3. Boztaş, Ö., Karakoç, F., Çoban, M.: Multidimensional meet-in-the-middle attacks on reduced-round TWINE-128. In: Avoine, G., Kara, O. (eds.) LightSec 2013. LNCS, vol. 8162, pp. 55–67. Springer, Heidelberg (2013)
4. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
5. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
6. Çoban, M., Karakoç, F., Boztaş, Ö.: Biclique cryptanalysis of TWINE. In: Pieprzyk, J., Sadeghi, A.-R., Manulis, M. (eds.) CANS 2012. LNCS, vol. 7712, pp. 43–55. Springer, Heidelberg (2012)
7. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: a new family of lightweight block ciphers. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 1–18. Springer, Heidelberg (2012)
8. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
9. Hong, D., et al.: HIGHT: a new block cipher suitable for low-resource device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
10. Knudsen, L.R.: DEAL - a 128-bit block cipher. Technical report, Department of Informatics, University of Bergen, Norway (1998)
11. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: a block cipher for IC-printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)
12. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New lightweight DES variants. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 196–210. Springer, Heidelberg (2007)
13. Mace, F., Standaert, F.X., Quisquater, J.J.: ASIC implementations of the block cipher SEA for constrained applications. In: Proceedings of the Third International Conference on RFID Security (2007). http://www.rfidsec07.etsit.uma.es/confhome.html
14. 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, Heidelberg (2011)
15. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: a lightweight, versatile block cipher. In: ECRYPT Workshop on Lightweight Cryptography, Louvain-la-Neuve, Belgium, 28–29 November 2011
16. Wu, W., Zhang, L.: LBLOCK: a lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)