Improved Meet-in-the-Middle Attacks on Crypton and mCrypton

Jingyi Cui[1], Jiansheng Guo[1,2*], Yanyan Huang[1] and Yipeng Liu[1]
[1] Zhengzhou Information Science and Technology Institute
Zhengzhou, P.R. China
[e-mail: xd_cjy@126.com]
[2] Science and Technology on Information Assurance Laboratory
Beijing, P.R. China
[tsg_31@126.com;]
*Corresponding author: Jiansheng Guo

Received September 14, 2016; revised February 9, 2017; accepted March 4, 2017; published May 31, 2017

Abstract

Crypton is a SP-network block cipher that attracts much attention because of its excellent performance on hardware. Based on Crypton, mCrypton is designed as a lightweight block cipher suitable for Internet of Things (IoT) and Radio Frequency Identification (RFID). The security of Crypton and mCrypton under meet-in-the-middle attack is analyzed in this paper. By analyzing the differential properties of cell permutation, several differential characteristics are introduced to construct generalized δ-sets. With the usage of a generalized δ-set and differential enumeration technique, a 6-round meet-in-the-middle distinguisher is proposed to give the first meet-in-the-middle attack on 9-round Crypton-192 and some improvements on the cryptanalysis of 10-round Crypton-256 are given. Combined with the properties of nibble permutation and substitution, an improved meet-in-the-middle attack on 8-round mCrypton is proposed and the first complete attack on 9-round mCrypton-96 is proposed.

Keywords: Cryptanalysis, Crypton, mCrypton, meet-in-the-middle attack, generalized δ-set, differential enumeration

This research was supported by Chinese Postdoctoral Science Foundation (2014M562582). We give our thanks to Prof. Houbing Song for checking our manuscript.

https://doi.org/10.3837/tiis.2017.05.020 **ISSN : 1976-7277**

1. Introduction

Crypton [1] is one of the fifteen candidates for AES [2]. Inspired by Square [3], Crypton has a SPN structure and owns many advantages that attract much attation, such as: identity of encryption and decryption, well-proven security, excellent performance on hardware and general applicability for various platforms. There are two versions of Crypton, Crypton v0.5 [4] and Crypton v1.0 [1]. The designers proposed Crypton v1.0 by modifying the key schedule and S-boxes in Crypton v0.5. With the development of IoT, RFID, Smart City[5] and Ad Hoc network[6], the security of these systems[7,8] becomes popular and the demand of block ciphers suitable for these resource-constrained environments is increasing. In 2006, Lim et al. [9] designed a lightweight block cipher mCrypton based on Crypton. The analysis on the security of Crypton and mCrypton could further develop the research on SPN-based block ciphers.

Many researchers have done deeply research on Crypton and mCrypton in recent years. In 2010, Mala et al. [10] gave an impossible differential attack on 7-round Crypton. In 2011, Wei et al. [11] proposed a related-key impossible differential attack on Crypton. In 2013, Kang et al. [12] gave the collision attacks on Crypton-192/256 and mCrypton -96/128. In 2014, Song et al. [13] proposed the biclique attacks on full-round Crypton -256 and mCrypton -128. Hao et al. [14] analyzed the security of mCrypton under meet-in-the-middle attack. In 2015, Shakiba et al. improved the biclique attacks on full-round Crypton [15] and mCrypton [16]. Jeong et al. [17] improved the biclique attack on full-round mCrypton. In 2016, Hao [18] introduced several meet-in-the-middle attacks on 10-round Crypton-256. Li and Jin [19] gave meet-in-the-middle attacks on 8-round mCrypton-96 and 9-round mCrypton-128. In CRYPTON 2016, Derbez et al. [20] proposed a search algorithm for generalized meet-in-the-middle attack and impossible differential attack. They simply estimated the security of 11-round Crypton-256, 9-round mCrypotn-96 and 10-round mCrypton-128 under meet-in-the-middle attack but without complete attacks.

Meet-in-the-middle attack was first proposed by Diffie and Hellman [21] in 1977. It is a powerful tool in the analysis of AES-like block cipher. In FSE 2008, Demirci et al. [22] introduced a 5-round meet-in-the-middle distinguisher to attack AES. In ASIACRYPT 2010, Dunkelman et al. [23] proposed several techniques to further improve the attacks on AES which are widely used now. In EUROCRYPT 2013, Derbez et al. [24] improved the meet-in-the-middle attacks on AES with the rebound-like idea. In FSE 2014, Li et al. [25] proposed key-dependent sieve technique to reduce the memory complexity and attacked 9-round AES-192 with a 5-round distinguisher. In 2015, Li et al. [26] gave the first 6-round distinguisher with the property of the linear transformation to attack 10-round AES-256.

In Section 2, notions used in this paper and the descriptions of Crypton and mCrypton are given. Section 3 proposes the related properties of Crypton and mCrypton. Section 4 presents the meet-in-the-middle attacks on Crypton with the usage of generalized δ-sets and properties of bit permutation. Section 5 gives the improved meet-in-the-middle attacks on 8-round and 9-round mCrypton. Section 6 concludes the whole paper.

2. Preliminaries

2.1 Notations

The following notations are used in the rest of this paper.

x_i : The *i*-th round state after key addition σ ;

y_i : The *i*-th round state after nonliner substitution γ ;

z_i : The *i*-th round state after bit permutation π ;

w_i : The *i*-th round state after cell transposition τ ;

$x_{i,col(j)}$: The *j*-th column of x_i ;

$x_{i,row(j)}$: The *j*-th row of x_i ;

k_{ei} : The *i*-th round subkey;

k_{ei}^* : The *i*-th round subkey after $\pi^{-1} \circ \tau^{-1}$, satisfies $k_{ei} = \tau \circ \pi(k_{ei}^*)$;

$x_i[j]$: The *j*-th cell of x_i ;

$<< a$: Left rotation of 32-bit word by a bits;

$<<_b^i$: Left rotation of each byte in 32-bit word by i bits.

2.2 Description of Crypton

Crypton is a SPN-based block cipher family. The block size is 128-bit and the key size is $64+32k (0 \le k \le 6)$. It has 12 rounds and which are numbered 1 to 12. A 128-bit state of Crypton can be indexed as **Fig. 1** shows. The number i means the i-th byte of the 128-bit state, $(i = 0,1,...,15)$.

3	2	1	0
7	6	5	4
11	10	9	8
15	14	13	12

a_{03}	a_{02}	a_{01}	a_{00}
a_{13}	a_{12}	a_{11}	a_{10}
a_{23}	a_{22}	a_{21}	a_{20}
a_{33}	a_{32}	a_{31}	a_{30}

Fig. 1. The state of Crypton

The round function is consisted of four transformations: nonlinear transformation γ, bit permutation π, byte transposition τ, key addition σ. The details are shown below:

Nonlinear Substitution γ : There are 4 different 8-bit S-boxes $S_i(0 \le i \le 3)$, for $S_2 = S_0^{-1}$, $S_3 = S_1^{-1}$. Crypton applies two different layers, γ_o in the odd round and γ_e in the even round.

Bit Permutation π : By using four masking bytes $m_0 = 0\text{xfc}, m_1 = 0\text{xfc}, m_2 = 0\text{xcf}$, $m_3 = 0\text{x3f}$, the bit permutation π mixes each column with $\pi_i(0 \le i \le 3)$:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \pi_i \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \Leftrightarrow b_j = \oplus_{k=0}^{3} (m_{i+j+k \bmod 4} \wedge a_k) .$$

There are two layers applied in Crypton, π_o in the odd round and π_e in the even round.

$$\pi_o(A) = (\pi_3(A^3), \pi_2(A^2), \pi_1(A^1), \pi_0(A^0)),\ \pi_e(A) = (\pi_1(A^3), \pi_0(A^2), \pi_3(A^1), \pi_2(A^0)).$$

Byte Tansposition τ : Move the (i,j)-th byte to the (j,i)-th position as $B = \tau(A) \Leftrightarrow b_{ij} = a_{ji}$.

Key Addition σ : The subkey is XORed to the corresponding state.

The whole encryption of Crypton needs to XOR the pre-whitening key at the beginning and add an output transformation $\phi_e = \tau \circ \pi_e \circ \tau$ to ensure the identity of encrypton and decryption. Similarly, we define $\phi_o = \tau \circ \pi_o \circ \tau$.

The key schedule of Crypton utilizes the master key to generate 52 32-bit words to construct 13 subkeys. The key schedule can be divided into two phase:

Generating Expand Keys Fill 0 to the left of the master key to get a 256-bit key $K = k_{31} ... k_1 k_0$. Divide K into 8 32-bit words:

$$\begin{array}{ll} U[0] = k_6 k_4 k_2 k_0 & V[0] = k_7 k_5 k_3 k_1 \\ U[1] = k_{14} k_{12} k_{10} k_8 & V[1] = k_{15} k_{13} k_{11} k_9 \\ U[2] = k_{22} k_{20} k_{18} k_{16} & V[2] = k_{23} k_{21} k_{19} k_{17} \\ U[3] = k_{30} k_{28} k_{26} k_{24} & V[3] = k_{31} k_{29} k_{27} k_{25} \end{array}$$

Using U, V , generate expand keys E_e:

1 $U' = \tau \circ \pi_o \circ \gamma(U), V' = \tau \circ \pi_e \circ \gamma(V);$

2 for $i = 0$ to 4

$$\begin{array}{l} E_e[i] = U'[i] \oplus (\oplus_{j=0}^{3} V'[j]) \\ E_e[i+4] = V'[i] \oplus (\oplus_{j=0}^{3} U'[j]) \end{array} .$$

Generating Round Keys With the usage of 8 expand keys $E_e(k)$, 13 round constants $C_e[i]$ and 4 masking constants MC_j , generate 13 round keys:

1 Compute the first two round keys $K_e[0,...,7]$. For $i = 0,1,2,3$, compute

$$K_e[i] = E_e[i] \oplus C_e[0] \oplus MC_i, \quad K_e[i+4] = E_e[i+4] \oplus C_e[1] \oplus MC_i ;$$

2 Comput the remaining 11 round keys.

For the even round,

$$\{E_e[3], E_e[2], E_e[1], E_e[0]\} \leftarrow \{E_e[0]^{<<^{6}_{b}}, E_e[3]^{<<^{6}_{b}}, E_e[2]^{<<16}, E_e[1]^{<<24}\} ,$$

$$K_e[4r+i] \leftarrow E_e[i] \oplus C_e[r] \oplus MC_i, (i = 0,1,2,3) ;$$

For the odd round,

$$\{E_e[7], E_e[6], E_e[5], E_e[4]\} \leftarrow \{E_e[6]^{<<16}, E_e[5]^{<<8}, E_e[4]^{<<^{2}_{b}}, E_e[7]^{<<^{2}_{b}}\} ,$$

$$K_e[4r+i] \leftarrow E_e[4+i] \oplus C_e[r] \oplus MC_i, (i = 0,1,2,3) .$$

2.3 Description of mCrypton

mCrypton is a reduced vesion of Crypton with a block size of 64-bit and three key sizes of 64-bit, 96-bit and 128-bit called mCrypton-64, mCrypton-96 and mCrypton-128 respectively. All of them have 12 rounds. The round function is similar with that of Crypton which contains four transformations: nonlinear substitution γ, bit permutation π, nibble transformation τ, key addition σ.

Nonlinear substitution γ contains four 4-bit S-boxes $S_i(0 \le i \le 3)$ that satisfy $S_2 = S_0^{-1}$, $S_3 = S_1^{-1}$.

Bit permutation π applies 4 column permutations $\pi_i(0 \le i \le 3)$ to mix the state with four masking nibbles $m_0 = \text{0xe}, m_1 = \text{0xd}, m_2 = \text{0xb}, m_3 = \text{0x7}$.

Nibble transformation τ and key addition σ are similar with those in Cypton.

The full encryption of mCrypton adds a pre-whitening key at the beginning and adds an output transformation $\phi = \tau \circ \pi \circ \tau$. The key schedules for mCrypton-64/96/128 are different, more details can refer [9].

3. Properties of Crypton and mCrypton

Here are some properties of Crypton and mCrypton used in the rest of paper.

Proposition 1 The transformation $\pi \circ \tau$ combined with the transformation $\phi = \tau \circ \pi \circ \tau$ is equivalent a byte permutation as shows and it satisfies $\pi_e \circ \pi_o = \pi_o \circ \pi_e$.

$$\begin{pmatrix} 3 & 2 & 1 & 0 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 15 & 14 & 13 & 12 \end{pmatrix} \xrightarrow{\tau \circ \pi_e \circ \pi_o} \begin{pmatrix} 4 & 0 & 12 & 8 \\ 5 & 1 & 13 & 9 \\ 6 & 2 & 14 & 10 \\ 7 & 3 & 15 & 11 \end{pmatrix}$$

Proof Taking the odd-round Crypton as an example, considering the truncated difference, we omit the key addition σ and then combine the transformation $\pi \circ \tau$ with $\phi = \tau \circ \pi \circ \tau$ to obtain $\tau \circ \pi_e \circ \pi_o$. Let a be the input difference that satisfies $\tau \circ \pi_e \circ \pi_o(a) = c$. If we set $c' = \tau(c)$ and $b = \pi_o(a)$, there must be $\pi_e \circ \pi_o(a) = c'$.

$$\begin{aligned} b_3 &= m_3a_3 \oplus m_0a_7 \oplus m_1a_{11} \oplus m_2a_{15} & c_3' &= m_1b_3 \oplus m_2b_7 \oplus m_3b_{11} \oplus m_0b_{15} \\ b_7 &= m_0a_3 \oplus m_1a_7 \oplus m_2a_{11} \oplus m_3a_{15} & c_7' &= m_2b_3 \oplus m_3b_7 \oplus m_0b_{11} \oplus m_1b_{15} \\ b_{11} &= m_1a_3 \oplus m_2a_7 \oplus m_3a_{11} \oplus m_0a_{15} & c_{11}' &= m_3b_3 \oplus m_0b_7 \oplus m_1b_{11} \oplus m_2b_{15} \\ b_{15} &= m_2a_3 \oplus m_3a_7 \oplus m_0a_{11} \oplus m_1a_{15} & c_{15}' &= m_0b_3 \oplus m_1b_7 \oplus m_2b_{11} \oplus m_3b_{15} \end{aligned}$$

So we can get $c_3' = a_{11}, c_7' = a_{15}, c_{11}' = a_3, c_{15}' = a_7$ directly. Similarly, each column is a permutation.

Proposition 2 For mCrypotn, the combination of $\pi \circ \tau$ and $\phi = \tau \circ \pi \circ \tau$ is equivalent to a nibble permutation as follows:

$$\begin{pmatrix} 3 & 2 & 1 & 0 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 15 & 14 & 13 & 12 \end{pmatrix} \xrightarrow{\tau \circ \pi \circ \pi} \begin{pmatrix} 12 & 8 & 4 & 0 \\ 13 & 9 & 5 & 1 \\ 14 & 10 & 6 & 2 \\ 15 & 11 & 7 & 3 \end{pmatrix}.$$

Proposition 3 When the input difference of π in Crypton is active only on two bytes and the output difference is active on two bytes, there are only four possible groups in which x presents nonzero difference:

I. For $x = \text{0x1,0x2,0x3}$,

$$(x,x,0,0)\to(x,0,0,x),\quad (0,x,x,0)\to(0,0,x,x),$$
$$(x,0,x,0)\to(x,0,x,0),\quad (0,x,0,x)\to(0,x,0,x),$$
$$(x,0,0,x)\to(x,x,0,0),\quad (0,0,x,x)\to(0,x,x,0);$$

II. For $x=0\text{x}4,0\text{x}8,0\text{xc}$,

$$(x,x,0,0)\to(x,x,0,0),\quad (0,x,x,0)\to(x,0,0,x),$$
$$(x,0,x,0)\to(0,x,0,x),\quad (0,x,0,x)\to(x,0,x,0),$$
$$(x,0,0,x)\to(0,x,x,0),\quad (0,0,x,x)\to(0,0,x,x);$$

III. For $x_1=0\text{x}10,0\text{x}20,0\text{x}30$,

$x_2=0\text{x}10,0\text{x}11,0\text{x}12,0\text{x}13,0\text{x}20,0\text{x}21,0\text{x}22,0\text{x}23,0\text{x}30,0\text{x}31,0\text{x}32,0\text{x}33$,

$$(x_1,x_1,0,0)\to(0,x_1,x_1,0),\quad (0,x_1,x_1,0)\to(x_1,x_1,0,0),$$
$$(0,0,x_1,x_1)\to(x_1,0,0,x_1),\quad (0,x_2,0,x_2)\to(0,x_2,0,x_2),$$
$$(x_1,0,0,x_1)\to(0,0,x_1,x_1),\quad (x_2,0,x_2,0)\to(x_2,0,x_2,0);$$

IV. For $x_1=0\text{x}40,0\text{x}80,0\text{xc}0$,

$x_2=0\text{x}40,0\text{x}44,0\text{x}48,0\text{x}4\text{c},0\text{x}80,0\text{x}84,0\text{x}88,0\text{x}8\text{c},0\text{xc}0,0\text{xc}4,0\text{xc}8,0\text{xcc}$,

$$(x_1,x_1,0,0)\to(0,0,x_1,x_1),\quad (0,x_1,x_1,0)\to(0,x_1,x_1,0),$$
$$(0,0,x_1,x_1)\to(x_1,x_1,0,0),\quad (0,x_2,0,x_2)\to(x_2,0,x_2,0),.$$
$$(x_1,0,0,x_1)\to(x_1,0,0,x_1),\quad (x_2,0,x_2,0)\to(0,x_2,0,x_2).$$

Proposition 4 When the input difference of π in mCrypotn is active only on two nibbles and the output difference is active on two nibbles, there are totally 28 differential characteristics:

I. For $x_1=0\text{x}1$, $x_2=0\text{x}1,0\text{x}5$,

$$(x_1,x_1,0,0)\to(x_1,0,0,x_1),\quad (0,x_1,x_1,0)\to(0,0,x_1,x_1),$$
$$(x_2,0,x_2,0)\to(x_2,0,x_2,0),\quad (0,x_2,0,x_2)\to(0,x_2,0,x_2),$$
$$(x_1,0,0,x_1)\to(x_1,x_1,0,0),\quad (0,0,x_1,x_1)\to(0,x_1,x_1,0);$$

II. For $x_1=0\text{x}2$, $x_2=0\text{x}2,0\text{xa}$,

$$(x_1,x_1,0,0)\to(x_1,x_1,0,0),\quad (0,x_1,x_1,0)\to(x_1,0,0,x_1),$$
$$(x_2,0,x_2,0)\to(0,x_2,0,x_2),\quad (0,x_2,0,x_2)\to(x_2,0,x_2,0),$$
$$(x_1,0,0,x_1)\to(0,x_1,x_1,0),\quad (0,0,x_1,x_1)\to(0,0,x_1,x_1);$$

III. For $x=0\text{x}4$,

$$(x,x,0,0)\to(0,x,x,0),\quad (0,x,x,0)\to(x,x,0,0),$$
$$(x,0,x,0)\to(x,0,x,0),\quad (0,x,0,x)\to(0,x,0,x),$$
$$(x,0,0,x)\to(0,0,x,x),\quad (0,0,x,x)\to(x,0,0,x);$$

IV. For $x=0\text{x}8$,

$$(x,x,0,0)\to(0,0,x,x),\quad (0,x,x,0)\to(0,x,x,0),$$
$$(x,0,x,0)\to(0,x,0,x),\quad (0,x,0,x)\to(x,0,x,0),$$
$$(x,0,0,x)\to(x,0,0,x),\quad (0,0,x,x)\to(x,x,0,0).$$

Proposition 5 When the input difference of π in Crypton is active on three bytes and the output difference is active on one byte, there are four possible groups in which x presents nonzero difference:

I. For $x=0\text{x}1,0\text{x}2,0\text{x}3$,

$$(x,x,x,0)\to(0,x,0,0);$$

II. For $x=0\text{x}4,0\text{x}8,0\text{xc}$,

$$(x,x,x,0)\to(0,0,x,0);$$

III. For $x = 0x10, 0x20, 0x30$,

$$(x, x, x, 0) \to (0, 0, 0, x);$$

IV. For $x = 0x40, 0x80, 0xc0$,

$$(x, x, x, 0) \to (x, 0, 0, 0).$$

Proposition 6 When the input difference of π in mCrypotn is active only on three nibbles and the output difference is active on one nibble, there is only 1 differential characteristic:

$$(0x8, 0x8, 0x8, 0) \to (0x8, 0, 0, 0).$$

Proposition 7 [19] Let $a[0,1,2,3]$ be the input of π in Crypton and $b[0,1,2,3]$ be the output. If $a[3] \| b[2,3]$ are fixed, the remaining five bytes can take 256 values.

Proposition 8 [14] Given one pair of input difference and output difference of S-box, there is one pair of input and output can be determined on average.

4. Meet-in-the-Middle Attacks on Crypton-192/256

We use two generalized δ-sets to construct two new 6-round meet-in-the-middle distinguishers to give the first attack on 9-round Crypton-192 and improve the attack on 10-round Crypton-256.

4.1 A New Meet-in-the-Middle Attack on 9-Round Crypton-192/256

Definition 1 [19] For a set of 256 Crypton states $\{y_2^0, y_2^1, ..., y_2^{255}\}$, if these elements satisfy $y_2^i[j] = y_2^0[j], j \in \{0,1,2,4,5,6,8,9,10,12,13,14,15\}$ and $z_2^i[j] = z_2^0[j], j \in \{0,1,2,4,5,6,8,...,15\}$ for $0 \le i \le 255$, we call this set a generalized δ-set of Crypton.

Combined with the properties of π, the first 6-round meet-in-the-middle distinguisher suitable for Cypton-192 is given by Theorem 1.

Theorem 1 For the generalized δ-set of Crypton $\{y_2^0, y_2^1, ..., y_2^{255}\}$, select the first 32 values $\{y_2^0, y_2^1, ..., y_2^{31}\}$ to encrypt 6 rounds. If the pair $\{y_2^0, y_2^j\}(0 \le j \le 255)$ satisfies the truncated differential characteristic shown in **Fig. 2**, the corresponding 496-bit ordered sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ could take 2^{172} possible values.

The gray bytes depict nonzero difference and the white bytes are inactive in **Fig. 2**.

Proof First, the 496-bit ordered sequence $(x_8^1[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ could be determined by the following 36 bytes:

$$x_3^0[12,13] \| x_4^0[0,1,2,3,4,5,6,7] \| x_5^0 \| k_{e5}[0,2,4,6,8,10,12,14] \| k_{e6}[0,8].$$

Known the value of Δy_2^i, deduce Δz_2^i and Δx_3^i because π and σ are linear. Knowing $x_3^0[12,13]$ can deduce $\Delta y_3^i[12,13]$ and $\Delta x_4^i[0,1,2,3,4,5,6,7]$. Then deduce Δx_5^i with the knowledge of $x_4^0[0,1,2,3,4,5,6,7]$. Knowing x_5^0 can decuce y_5^i and encrypt it to get $y_7^i[0,8]$ and $\Delta y_7^i[0,8]$ with $k_{e5}[0,2,4,6,8,10,12,14] \| k_{e6}[0,8]$. Then because of $\Delta y_7^i[0,8] = \Delta x_8^i[0,2]$, we can get the sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$.

Next, the 36 bytes can be represented by the following 23 bytes:

$$\Delta z_2^j[3,7] \| x_3^0[12,13] \| x_4^0[0,1,2,3,4,5,6,7] \| y_6^0[0,2,4,6,8,10,12,14] \| y_7^0[0,8] \| \Delta y_7^0[0].$$

On the one hand, known $\Delta z_2^j[3,7] \| x_3^0[12,13] \| x_4^0[0,1,2,3,4,5,6,7]$,we can deduce Δx_5^i. On the other hand, knowing $y_7^0[0,8] \| \Delta y_7^0[0,8]$ can deduce $\Delta y_6^i[0,2,4,6,8,10,12,14]$. Then we obtain Δy_5^i. According to Proposition 8, we can get one pair $x_5^i \| y_5^i$ on average to deduce the corresponding keys $k_{e5}[0,2,4,6,8,10,12,14] \| k_{e6}[0,8]$.

As $\Delta z_2^j[3,7]$ can only take 256 possible values and $\Delta y_7^0[0]$ can take 15 values in total, the 496-bit sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ can take 2^{172} possible values.

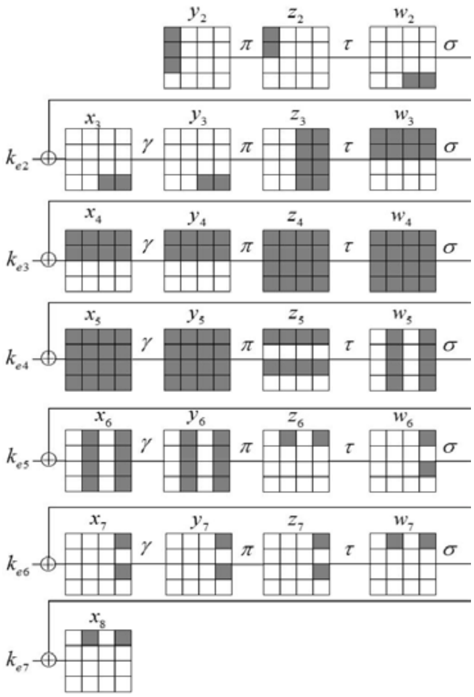

Fig. 2. 6-round meet-in-the-middle distinguisher

When the values of $\Delta y_2^i[3,7,11]$ are 0, $\Delta z_2^j[3,7]$ are equal to 0 with a probability of 2^{-16}. Set the differences $\Delta z_5^i[0,1,2,3,8,9,10,11]$ to nonzero and other bytes to zero with a probability of 2^{-64}. When there are only {0,2}-th byte to be nonzero in Δz_6^j, the probability is 2^{-48}. There are only 15 possible values of $\Delta y_7^i[0,8]$ with a probability of 2^{-12}. So the probability of this distinguisher is 2^{-140}.

With the usage of this distinguisher, we can extend one more round forward and two more rounds backward to give the 9-round attack on Crypton-192 as **Fig. 3** shows. The gray bytes depict nonzero difference and the white bytes are inactive. The slashed bytes are the subkey bytes needed to be guessed. The whole attack can be divided into two phases: Precomputation Phase and Key Recovery Phase.

Precomputation Phase: With the usage of time and memory tradeoff technique, three tables T_0, T_1, T_2 are estibalished to reduce the time complexity of key recovery phase.

Table T_0: Store 2^{172} possible 496-bit sequences $(x_8^1[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ with a time complexity of $2^{176} \times 2^5 \times 6/9 \approx 2^{180.4}$ 9-round Crypton encryptions and a memory complexity of $2^{176} \times 496/128 \approx 2^{178.0}$ Crypton states.

Table T_1: Store the encryption from $y_8[0]$ to $z_{9,row(0)}'$. Decrypt $z_{9,row(0)}'$ to get $y_8[0]$ with a 40-bit key $k_{e9,row(0)} \| k_{e8}[0]$. Take $k_{e9,row(0)} \| k_{e8}[0] \| z_{9,row(0)}'$ as index to store $y_8[0]$ with a time complexity of $2^{32} \times 2^{40} \times 2/9 \approx 2^{69.8}$ 9-round Crypton encryptions and a memory complexity of $2^{72}/16 = 2^{68}$ Crypton states.

Table T_2: Store the encryption from $y_8[2]$ to $z_{9,row(2)}'$ which is simialr with the table T_1.

Key Recovery Phase: We need to find a plaintext pair suitable for **Fig. 3** and construct the ordered sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ to match the table T_0.

1. Choose 2^{21} structures: a set of 2^{96} plaintexts are all possible 128-bit values with {0,1,2,4,5,6,8,9,10,12,13,14}-th running over all values and others fixed constants. We need 2^{212} pairs to get 1 pair that satisfies the truncated differential characteirstic shown in **Fig. 3**.

2. Filter those pairs that the ciphertext difference on {1,3,5,7,9,11,13,15}-th bytes are 0 and others are active. $2^{212} \times 2^{-64} = 2^{148}$ plaintext pairs remain.

3. Do these following substeps for each of 2^{148} pairs:

3.1 Guess $\Delta y_8[0,2]$ to deduce $\Delta x_9[0,1,2,3,8,9,10,11]$. Known the ciphertext difference, deduce $\Delta y_9[0,1,2,3,8,9,10,11]$. Deduce $x_9[0,1,2,3,8,9,10,11] \| y_9[0,1,2,3,8,9,10,11]$ and $k_{e9}^*[0,1,2,3,8,9,10,11]$ according to Property 8.

3.2 For each deduced key $k_{e9}^*[0,1,2,3,8,9,10,11]$, deduce $\Delta y_8[0,2]$. Guess 15 possible values of $\Delta x_8[0,2]$. Then deduce $x_8[0,2] \| y_8[0,2]$ and $k_{e8}^*[0,2]$.

3.3 Guess $\Delta x_2[3,7,11]$ to deduce $\Delta y_1[0,1,2,4,5,6,8,9,10,12,13,14]$. Known the plaintext difference, deduce $\Delta x_1[0,1,2,4,5,6,8,9,10,12,13,14]$. Then according to Property 8, deduce $x_1[0,1,2,4,5,6,8,9,10,12,13,14] \| y_1[0,1,2,4,5,6,8,9,10,12,13,14]$ and the corresponding key $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14]$.

3.4 For each deduced key $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14]$, deduce $\Delta x_2[3,7,11]$. Guess 256 possible values of $\Delta y_2[3,7,11]$. Deduce $x_2[3,7,11] \| y_2[3,7,11]$ and $k_{e1}[3,7,11]$.

3.5 For those keys in 3.3 and 3.4, choose a plaintext P^0 and encrypt it to get $y_2^0[3,7,11]$ and $w_1^0[0,1,2,4,5,6,8,9,10]$. Known $\Delta y_2[3,7,11]$, get $y_2^i[3,7,11]$ and decrypt them to get $w_1^0[3,7,11]$. Because $w_1^i[0,1,2,4,5,6,8,9,10]$ is equal to $w_1^0[0,1,2,4,5,6,8,9,10]$, we can deduce $P^i[0,1,2,4,5,6,8,9,10,12,13,14]$. With the knowledge of $P^i[3,7,11,15] = P^0[3,7,11,15]$, obtain the plaintext P^i and its corresponding ciphertext.

3.6 With the knowledge of $k_{e8}^*[0,2] \| k_{e9}^*[0,1,2,3,8,9,10,11]$, look up the table T_1 and T_2 to obtain the sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$. If the sequence lies in the table T_0, select the key as a candidate. If not, discard the key. A wrong key pass the test with a probability of $2^{176} \times 2^{-496} = 2^{-320}$.

4 There remains $1 + 2^{144} \times 2^{16} \times 15 \times 2^{24} \times 2^{8} \times 2^{-320} \approx 1$ key $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14] \| k_{e1}[3,7,11] \| k_{e8}^*[0,2] \| k_{e9}^*[0,1,2,3,8,9,10,11]$. Exhaust the rest bytes to recover the master key.

The details of this attack are shown in **Fig. 3**. Theorem 2 anlyzes the complexity of this attack.

Theorem 2 With the usage of a 6-round distinguisher, the meet-in-the-middle attack on 9-round Crypton is proposed with a time of complexity of $2^{190.3}$, a memory complexity of 2^{178} and a data complexity of 2^{117}.

Proof In the precomputation phase, the construction of the table T_0 needs a time complexity of $2^{180.4}$ 9-round Crypton encryptions and a memory of $2^{178.0}$ Crypton states.

In the key recovery phase, Step 3.6 contributes the main time complexity. In Step 3.6, we need to look up the table T_0 that we translate the unit of time complexity into 9-round Crypton encrypton [10]. Its time complexity is $2^{148}\times 2^{16}\times 15\times 2^{24}\times 2^{8}\times 2^{5}\times 2^{-14}\times 6/9\approx 2^{190.3}$ 9-round Crypton encryptions.

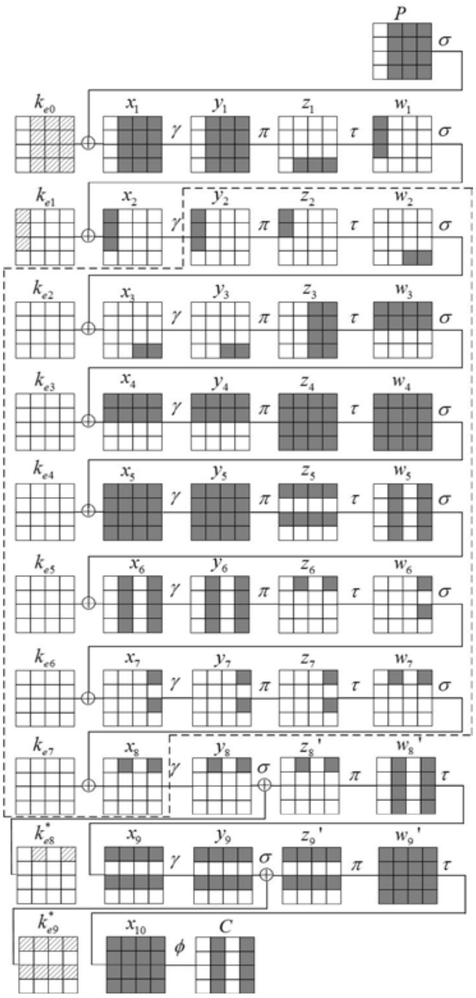

Fig. 3. Meet-in-the-middle attack on 9-round Crypton

4.2 Improved Meet-in-the-Middle Attack on 10-Round Crypton-256

Definition 2 For a set of 256 Crypton states $\{y_2^0, y_2^1, ..., y_2^{255}\}$, if the elements in this set satisfy $y_2^i[j] = y_2^0[j]$, $j\in\{0,1,2,4,5,6,8,...,15\}$ and $z_2^i[j] = z_2^0[j]$, $j\in\{0,1,2,4,5,6,8,9,10,12,13,14,15\}$ for $0\le i\le 255$, we call this set a generalized δ-set of Crypton.

We can construct another 6-round distinguisher with this generalized δ-set. Theorem 3 shows the details of the new 6-round distinguisher in the dashed box of **Fig. 4**. The gray bytes depict non-zero difference and the white bytes are inactive.

Theorem 3 For the generalized δ-set of Crypton $\{y_2^0, y_2^1, ..., y_2^{255}\}$, select the first 32 values $\{y_2^0, y_2^1, ..., y_2^{31}\}$ to encrypt 6 rounds. If the pair $\{y_2^0, y_2^j\}(0 \le j \le 255)$ satisfies the truncated differential characeristic shown in the dashed box of **Fig. 4**, the corresponding 496-bit ordered sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ could take 2^{212} possible values.

Then attack 10-round Crypton-256 by extending one more round forward and three more rounds backward. This attack contains two phases: Precomputation Phase and Key Recovery Phase.

Precomputation Phase: Establish 5 tables T_0, T_1, T_2, T_3, T_4.

Table T_0: Store 2^{212} possible 496-bit sequences $(x_8^1[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ with a time complexity of $2^{212} \times 32 \times 2.5/10 = 2^{215}$ 10-round Crypton encryptions and a memory complexity of $2^{212} \times 496/128 = 2^{214.0}$ Crypton states.

Table T_1: Exhaust $\Delta z_9{'}[2,10]$. Take $\Delta C_{col(0)}$ as index to store $y_{10,row(2)}$.

Table T_2: Exhaust $\Delta z_9{'}[3,11]$. Take $\Delta C_{col(1)}$ as index to store $y_{10,row(3)}$.

Table T_3: Exhaust $\Delta z_9{'}[0,8]$. Take $\Delta C_{col(2)}$ as index to store $y_{10,row(0)}$.

Table T_4: Exhaust $\Delta z_9{'}[1,9]$. Take $\Delta C_{col(3)}$ as index to store $y_{10,row(1)}$.

Key Recovery Phase: 1. Choose 2^{53} structures: a set of 2^{64} plaintexts are all possible 128-bit values with {0,1,4,5,8,9,12,13}-th running over all values and others fixed constants. We need 2^{180} pairs to get 1 pair that satisfies the truncated differential characteirstic shown in **Fig. 4**.

2. Do these following substeps for each of the 2^{180} pairs:

2.1 Guess $\Delta y_9[0,1,2,3,8,9,10,11]$ to deduce Δx_{10}. Known the ciphertext difference, deduce Δy_{10}. Look up the table T_1, T_2, T_3, T_4 to deduce $x_{10} \| y_{10}$ and k_{e10}^*.

2.2 With the usage of key schedule, deduce k_{e0} from k_{e10}^*. Encrypt the 2^{180} plaintext pairs for one round with the 2^{64} keys in 2.1 to filter those keys lead $\Delta z_1[0,1,4,5,8,9]$ nonzero with a probability of 2^{-48}.

2.3 For the remaining deduced keys in 2.2, compute $\Delta y_9[0,1,2,3,8,9,10,11]$ .Guess $\Delta y_8[0,2]$ to deduce $\Delta x_9[0,1,2,3,8,9,10,11]$. Then deduce $x_9[0,1,2,3,8,9,10,11] \| y_9[0,1,2,3,8,9,10,11]$ and $k_{e9}^*[0,1,2,3,8,9,10,11]$ according to Property 8.

2.4 For the remaining keys k_{e10}^*, deduce $k_{e8}^*[0,2]$ according to key schedule .

2.5 Known k_{e0}, we can compute $\Delta x_2[3,7]$. Guess 256 possible values of $\Delta y_2[3,7]$ to deduce $x_2[3,7] \| y_2[3,7]$ and $k_{e1}[3,7]$.

2.6 Choose a plaintext P^0 and encrypt it to get $w_1^0[0,1,2,4,5,6]$ and $y_2^0[3,7]$. Known $\Delta y_2[3,7]$, we can get $y_2^i[3,7,11]$ and decrypt them to get $w_1^0[3,7]$. Because $w_1^i[0,1,2,4,5,6]$ is equal to $w_1^0[0,1,2,4,5,6]$, we can deduce $P^i[0,1,4,5,8,9,12,13]$. With the knowledge of $P^i[j] = P^0[j], \quad j \in \{2,3,6,7,10,11,14,15\}$, obtain the plaintext P^i and its corresponding ciphertext.

2.7 With the knowledge of $k^*_{e8}[0,2] || k^*_{e9}[0,1,2,3,8,9,10,11] || k^*_{e10}$, look up table T_1 and T_2 to obtain the sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$. If the sequence lies in the table T_0, select the key as a candidate. If not, discard the key. A wrong key pass the test with a probability of $2^{212} \times 2^{-496} = 2^{-284}$.

4 There remains $1 + 2^{180} \times 2^{64} \times 2^{16} \times 2^{-48} \times 2^{8} \times 2^{-284} \approx 1$ key $k_{e1}[3,7] || k^*_{e9}[0,1,2,3,8,9,10,11]$ $|| k^*_{e10}$. Exhaust the rest bytes to recover the master key.

Theorem 4 With the usage of a 6-round distinguisher, the meet-in-the-middle attack on 10-round Crypton-256 is proposed with a time of complexity of $2^{240.7}$, a memory complexity of $2^{214.0}$ and a data complexity of 2^{117}.

Proof In the precomputation phase, the comstruction of the table T_0 needs a time complexity of 2^{215} 10-round Crypton encryptions and a memory of 2^{214} Crypton states.

In the key recovery phase, Step 2.2 encrypts 2^{180} pairs for one round with 2^{64} keys. The time complexity for each plaintext is equivalent to 0.5 round Crypton encryption. The total time complexity is $2^{180} \times 2^{64} \times 2 \times 0.5 / 10 \approx 2^{240.7}$ 10-round Crypton encryptions.

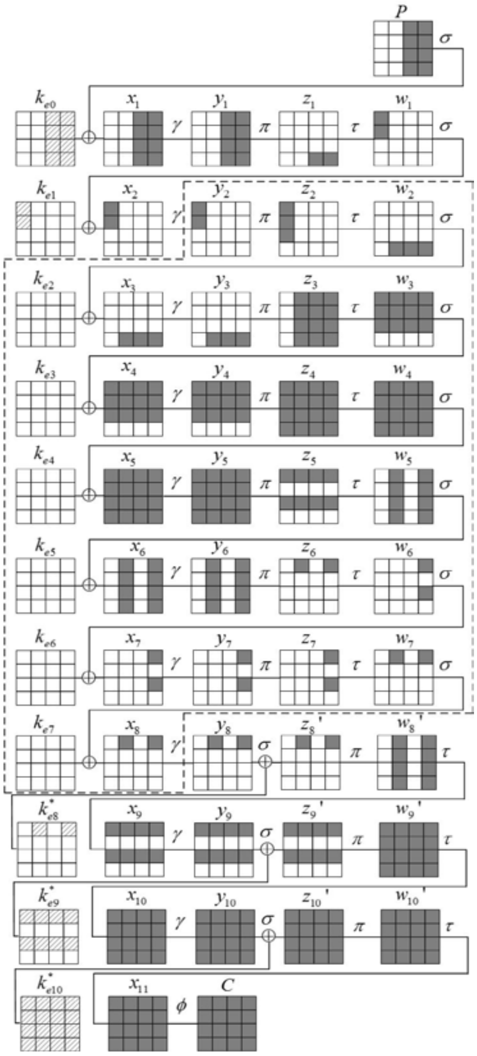

Fig. 4. Meet-in-the-middle attack on 10-round Crypton

5. Meet-in-the-Middle on mCrypton-96/128

We utilize Property 6 to construct a new generalized δ-set and give the attacks on 8-round and 9-round mCrypton with a 5-round and 6-round distinguisher respectively.

Definition 2 For a set of 512 mCrypton states $\{x_2^0, x_2^1, ..., x_2^{511}\}$, if these satisfy $x_2^i[j] = x_2^0[j], j \in \{0,1,2,4,5,6,8,9,10,12,...,15\}$ and $y_2^i[j] \oplus y_2^0[j] = 0\text{x}8, j \in \{3,7,11\}$, we call this set a generalized δ-set of mCrypton.

5.1 Improved Meet-in-the-Middle Attack on 8-Round mCrypton-96/128

Theorem 5 proposes a 5-round meet-in-the-middle distinguisher of mCrypton. The details are shown in **Fig. 5.** The gray nibbles depict non-zero difference and the white nibbles are inactive.

Theorem 5 For the generalized δ-set of mCrypton $\{x_2^0, x_2^1, ..., x_2^{511}\}$, select the first 64 values to encrypt 5 rounds. If the pair $\{x_2^0, x_2^j\}(0 \le j \le 63)$ satisfies the truncated differential path in **Fig. 5**, the 252-bit ordered sequence $(y_7^0[3] \oplus y_7^1[3], y_7^0[3] \oplus y_7^2[3], ..., y_7^0[3] \oplus y_7^{63}[3])$ could take 2^{53} possible values.

Proof First, the 252-bit ordered sequence $(y_7^0[3] \oplus y_7^1[3], y_7^0[3] \oplus y_7^2[3], ..., y_7^0[3] \oplus y_7^{63}[3])$ could be determined by the following 29 nibbles:

$$x_2^0[3,7,11] \,||\, x_3^0[12] \,||\, x_4^0[0,1,2,3] \,||\, x_5^0 \,||\, k_{e5}[0,4,8,12] \,||\, k_{e6}[3].$$

Known Δx_2^i, deduce Δy_2^i and Δx_3^i because π and σ are linear. Knowing $x_3^0[12]$ can deduce $\Delta y_3^i[12]$ and $\Delta x_4^i[0,1,2,3]$. Then deduce Δx_5^i with knowledge of $x_4^0[0,1,2,3]$. Knowing x_5^0 can decuce y_5^i and encrypt it with $k_{e5}[0,4,8,12] \,||\, k_{e6}[3]$ to get $y_7^i[3]$. Then we can get $(y_7^0[3] \oplus y_7^1[3], y_7^0[3] \oplus y_7^2[3], ..., y_7^0[3] \oplus y_7^{63}[3])$.

Next, the 29 bytes can be represented by the following 23 bytes:

$$\Delta x_2^j[3,7,11] \,||\, \Delta y_2^j[3,7,11] \,||\, x_3^0[12] \,||\, x_4^0[0,1,2,3] \,||\, y_6^0[0,4,8,12] \,||\, y_7^0[3] \,||\, \Delta y_7^0[3].$$

As $\Delta x_2^j[3,7,11]$ can only take 512 possible values and $\Delta y_2^j[3,7,11]$ can take 1 value in total, the 252-bit sequence $(y_7^0[3] \oplus y_7^1[3], y_7^0[3] \oplus y_7^2[3], ..., y_7^0[3] \oplus y_7^{63}[3])$ can take 2^{53} possible values.

In our meet-in-the-middle attack on 8-round mCrypton, we store all the possibles distinguishers in a hash table in precomputation phase and match them in key recovery phase.

Precomputation Phase: Establish the table T to store 2^{53} possible values of 252-bit sequences $(y_7^0[3] \oplus y_7^1[3], y_7^0[3] \oplus y_7^2[3], ..., y_7^0[3] \oplus y_7^{63}[3])$.

Key Recovery Phase: 1. Choose 2^{13} structures: a set of 2^{48} plaintexts are all possible 64-bit values with {0,1,2,4,5,6,8,9,10,12,13,14}-th running over all values and others fixed constants. We need 2^{108} pairs to get 1 pair that satisfies the truncated differential characteirstic shown in **Fig. 5**.

2. Filter those pairs that the ciphertext difference on {0,1,2,4,5,6,8,9,10,12,13,14}-th bytes are 0 and others are active.There remains 2^{60} plaintext pairs.

3. Do these following substeps for each of 2^{60} pairs:

3.1 Guess $\Delta y_7[0]$ to deduce $\Delta x_8[12,13,14,15]$. Known the ciphertext difference, deduce $\Delta y_8[12,13,14,15]$. Deduce $x_8[12,13,14,15] \,||\, y_8[12,13,14,15]$ and $k_{e8}^*[12,13,14,15]$ according to Property 8.

3.2 Guess 2^9 possible values of $\Delta x_2[3,7,11]$. Deduce $\Delta y_1[0,1,2,4,5,6,8,9,10,12,13,14]$ and $\Delta x_1[0,1,2,4,5,6,8,9,10,12,13,14]$ with the knowledge of plaintext difference. Then obtain $x_1[0,1,2,4,5,6,8,9,10,12,13,14] \| y_1[0,1,2,4,5,6,8,9,10,12,13,14]$ and $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14]$.

3.3 Choose a plaintext P^0 and encrypt to get $y_1^0[0,1,2,4,5,6,8,9,10,12,13,14]$. Known $\Delta y_1[0,1,2,4,5,6,8,9,10,12,13,14]$, get $y_1^i[0,1,2,4,5,6,8,9,10,12,13,14]$ and decrypt them to get $P^i[0,1,2,4,5,6,8,9,10,12,13,14]$. With the knowledge of $P^i[3,7,11,15] = P^0[3,7,11,15]$, obtain the plaintext P^i and its corresponding ciphertext.

3.4 With the knowledge of $k_{e8}^*[12,13,14,15]$, decrypt the ciphertexts to obtain the sequence $(y_7^0[3] \oplus y_7^1[3], y_7^0[3] \oplus y_7^2[3], ..., y_7^0[3] \oplus y_7^{63}[3])$. If the sequence match in the table T, select the key as a candidate. If not, discard the key. A wrong key pass the test with a probability of $2^{53} \times 2^{-252} = 2^{-199}$.

4 There are $1 + 2^{60} \times 2^9 \times 2^4 \times 2^{-199} \approx 1$ key $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14] \| k_{e8}^*[12,13,14,15]$ remaining.

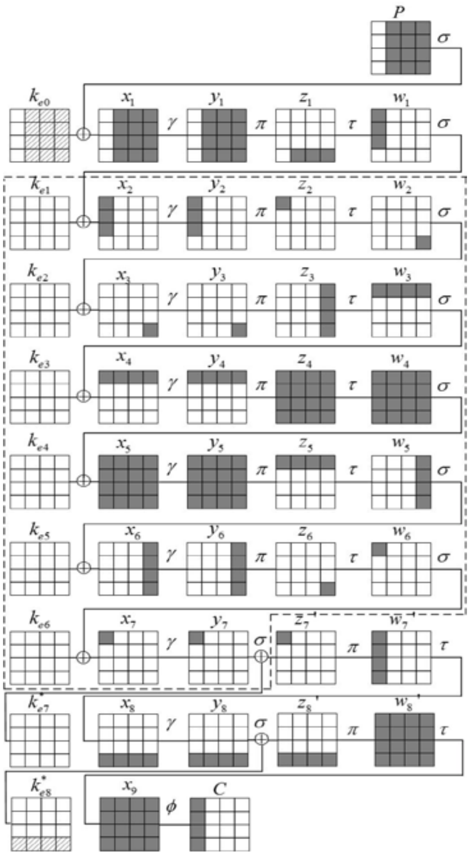

Fig. 5. Meet-in-the-middle attack on 8-round mCrypton

Theorem 6 With the usage of a 5-round distinguisher, the meet-in-the-middle attack on 8-round mCrypton is proposed with a time of complexity of 2^{76}, a memory complexity of 2^{55} and a data complexity of 2^{51}.

Proof In the precomputation phase, the construction of the table T needs a time complexity of $2^{53} \times 2^6 \times 2/8 = 2^{57}$ 8-round mCrypton encryptions and a memory of $2^{53} \times 4 = 2^{55}$ Crypton states.

In the key recovery phase, Step 3.4 owns the main time complexity. We need to look up the table T with a time complexity of $2^{60} \times 2^4 \times 2^9 \times 2^6 / 8 = 2^{76}$ 8-round mCrypton encryptions.

5.2 A New Meet-in-the-Middle Attack on 9-Round mCrypton-96/128

Similarly with Theorem 5, a new meet-in-the-middle attack on 9-round mCrypton-96/128 is proposed in this subsection. The details are shown in **Fig. 6.** The gray nibbles depict non-zero difference and the white nibbles are inactive. The slashed bytes are the recovered subkey nibbles.

Theorem 7 For the generalized δ-set of mCrypton $\{x_2^0, x_2^1, ..., x_2^{511}\}$, select the first 32 values to encrypt 6 rounds. If the pair $\{x_2^0, x_2^j\}(0 \le j \le 31)$ satisfies the truncated differential path in **Fig. 6**, the 248-bit ordered sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ could take 3×2^{77} possible values.

Proof First, the 248-bit ordered sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ could be determined by the following 36 nibbles:

$$x_2^0[3,7,11] \,||\, x_3^0[12] \,||\, x_4^0[0,1,2,3] \,||\, x_5^0 \,||\, k_{e5}[0,2,4,6,8,10,12,14] \,||\, k_{e6}[0,8] \,||\, k_{e7}[0,2].$$

Known Δx_2^i, deduce Δy_2^i with the usage of Property 7 and Δx_3^i because π and σ are linear. Knowing $x_3^0[12]$ can deduce $\Delta y_3^i[12]$ and $\Delta x_4^i[0,1,2,3]$. Then deduce Δx_5^i with knowledge of $x_4^0[0,1,2,3]$. Knowing x_5^0 can decuce y_5^i and encrypt it with $k_{e5}[0,2,4,6,8,10,12,14] \,||\, k_{e6}[0,8] \,||\, k_{e7}[0,2]$ to get $x_8^i[0,2]$. Then get $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$.

Next, the 36 bytes can be represented by the following 23 bytes:

$$\Delta x_2^j[3,7,11] \,||\, \Delta y_2^j[3,7,11] \,||\, x_3^0[12] \,||\, x_4^0[0,1,2,3] \,||\, y_6^0[0,2,4,6,8,10,12,14] \,|| \\ y_7^0[0,8] \,||\, y_8^0[0,2] \,||\, \Delta y_8^0[0].$$

As $\Delta x_2^j[3,7,11]$ can only take 512 possible values, $\Delta y_2^j[3,7,11]$ can take 1 value in total and $\Delta y_8^0[0]$ can take 3 value with the usage of Property 5, the 248-bit sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ can take 3×2^{77} possible values.

This attack contains two phases: precomputation phase and key recovery phase.

Precomputation Phase: Precompute and store 3×2^{77} values of 248-bit sequences $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$ in the table T.

Key Recovery Phase: 1. Choose 2^{14} structures: a set of 2^{48} plaintexts are all possible 64-bit values with {0,1,2,4,5,6,8,9,10,12,13,14}-th running over all values and others fixed constants. We need 2^{109} pairs to get 1 pair that satisfies the truncated differential characteirstic shown in **Fig. 6**.

2. Filter those pairs that the ciphertext difference on {1,3,5,7,9,11,13,15}-th bytes are 0 and others are active. 2^{77} plaintext pairs remain.

3. Do these following substeps for each of 2^{77} pairs:

3.1 Guess $\Delta y_8[0,2]$ to deduce $\Delta x_9[0,1,2,3,8,9,10,11]$. Known the ciphertext difference, deduce $\Delta y_9[0,1,2,3,8,9,10,11]$. Deduce $x_9[0,1,2,3,8,9,10,11] \| y_9[0,1,2,3,8,9,10,11]$ and $k_{e9}^*[0,1,2,3,8,9,10,11]$ according to Property 8.

3.2 Guess 2^9 possible values of $\Delta x_2[3,7,11]$. Deduce $\Delta y_1[0,1,2,4,5,6,8,9,10,12,13,14]$ and $\Delta x_1[0,1,2,4,5,6,8,9,10,12,13,14]$ with the knowledge of plaintext difference. Then obtain $x_1[0,1,2,4,5,6,8,9,10,12,13,14] \| y_1[0,1,2,4,5,6,8,9,10,12,13,14]$ and $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14]$.

3.3 Choose a plaintext P^0 and encrypt it to get $y_1^0[0,1,2,4,5,6,8,9,10,12,13,14]$. Known $\Delta y_1[0,1,2,4,5,6,8,9,10,12,13,14]$, get $y_1^i[0,1,2,4,5,6,8,9,10,12,13,14]$ and decrypt them to get $P^i[0,1,2,4,5,6,8,9,10,12,13,14]$. With the knowledge of $P^i[3,7,11,15] = P^0[3,7,11,15]$, obtain the plaintext P^i and its corresponding ciphertext.

3.4 With the knowledge of $k_{e9}^*[0,1,2,3,8,9,10,11]$, decrypt the ciphertexts to obtain the sequence $(x_8^1[0,2] \oplus x_8^0[0,2], x_8^2[0,2] \oplus x_8^0[0,2], ..., x_8^{31}[0,2] \oplus x_8^0[0,2])$. If the sequence lies in the table T, select the key as a candidate. If not, discard the key. A wrong key pass the test with a probability of $3 \times 2^{77} \times 2^{-248} = 2^{-169}$.

4 There remains $1 + 2^{77} \times 2^9 \times 2^8 \times 2^{-169} \approx 1$ key $k_{e0}[0,1,2,4,5,6,8,9,10,12,13,14]$ $\| k_{e9}^*[0,1,2,3,8,9,10,11]$.

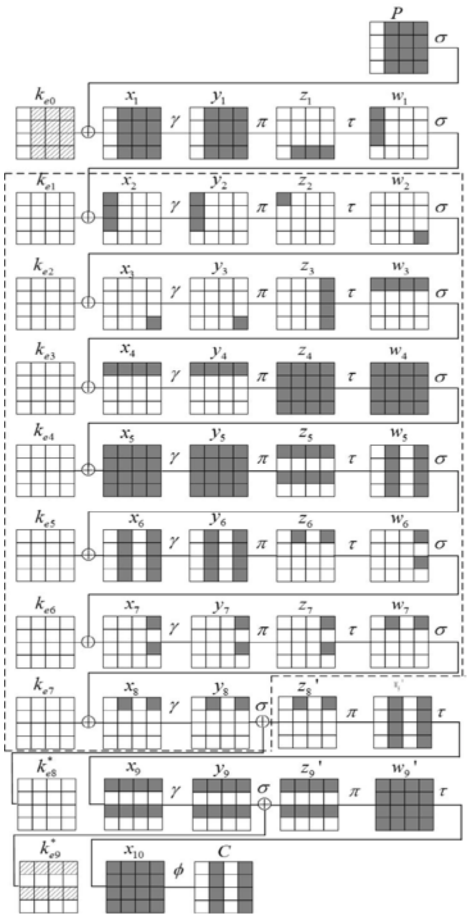

Fig. 6. Meet-in-the-middle attack on 9-round mCrypton

Theorem 8 With the usage of a 6-round distinguisher, the meet-in-the-middle attack on 9-round mCrypton is proposed with a time of complexity of $2^{95.8}$, a memory complexity of $2^{80.6}$ and a data complexity of 2^{62}.

Proof In the precomputation phase, the construction of the table T needs a time complexity of $3\times 2^{77}\times 2^5\times 2/9\approx 2^{81.4}$ 9-round mCrypton encryptions and a memory of $3\times 2^{77}\times 4 = 2^{80.6}$ Crypton states.

In the key recovery phase, Step 3.4 contributes the main time complexity. We need to look up table T with no more than $2^{77}\times 2^8\times 2^9\times 2^5/9\approx 2^{95.8}$ 9-round mCrypton encryptions.

6. Conclusion

The security of Crypton and mCrypton under meet-in-the-middle attack is analyzed in this paper. We concentrate on the differential properties of π and construct various generalized δ-sets by introducing several new differential characteristics. With the usage of a new generalized δ-sets, the first 6-round meet-in-the-middle distinguisher suitable for Crypton-192 is found and the first meet-in-the-middle attack on 9-round Crypton-192 is proposed. We also improve the attack on 10-round Crypton-256. By using the differential properties of π and γ, we give a new generalized δ-set to construct 5-round and 6-round distinguishers to attack 8-round and 9-round mCrypton respectively. At the same condition, the attacks in this paper could attack these two reduced-round ciphers with less resource. Each cipher should be well evaluated before it come into widespread use. The comparison of main meet-in-the-middle attacks on Crypton and mCrypton is shown in **Table 1**.

Table 1. Comparison of Main Meet-in-the-Middle Attacks on Crypton and mCrypton*

Type	Round	Data	Time	Memory	Refer.
Crypton-192	8	2^{113}	2^{155}	2^{138}	[15]
	9	2^{117}	$2^{190.3}$	2^{178}	Sect. 3.1
Crypton-256	9	2^{113}	$2^{245.05}$	$2^{241.17}$	[14]
	9	2^{117}	$2^{190.3}$	2^{178}	Sect. 3.1
	10	2^{113}	$2^{245.05}$	$2^{241.59}$	[14]
	10	2^{113}	2^{246}	$2^{209.59}$	[14]
	10	2^{117}	$2^{240.7}$	2^{214}	Sect. 3.2
	11	-	-	-	[16] **
mCrypton-96	8	2^{53}	2^{91}	2^{82}	[15]
	8	2^{51}	2^{76}	2^{55}	Sect. 4.1
	9	2^{57}	2^{83}	2^{83}	[16] **
	9	2^{62}	$2^{95.8}$	$2^{80.6}$	Sect. 4.2
mCrypton-128	9	2^{53}	2^{112}	2^{106}	[15]
	9	2^{62}	$2^{95.8}$	$2^{80.6}$	Sect. 4.2
	10	2^{55}	2^{117}	2^{103}	[16]

*Precomputation included
**Without complete attack

References

[1] Chae Hoon Lim, "A revised version of Crypton - Crypton V1.0," in *Proc. of 6th Fast Software Encryption Workshop*, pp. 31-45, March 24-26, 1999. Article (CrossRef Link).

[2] Eli Biham, "A note on comparing the AES candidates," Second AES Candidate Conference, 1999. Article (CrossRef Link).

[3] Joan Daemen, Lars R.Knudsen and Vincent Rijmen, "The block cipher Square," in *Proc. of 4th Fast Software Encryption Workshop*, pp. 149-165, January 20-22, 1997. Article (CrossRef Link).

[4] Chae Hoon Lim, "Crypton: A new 128-bit block cipher," NIST AES Proposal, 1998. Article (CrossRef Link).

[5] Maryam Pouryazdan, Burak Kantarci, Tolga Soyata, et al., "Anchor-Assisted and Vote-Based Trustworthiness Assurance in Smart City Crowdsensing," *IEEE Access*, vol. 4, pp. 529-541, 2016. Article (CrossRef Link).

[6] Wenjia Li and Houbing Song, "ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960-969, 2016. Article (CrossRef Link).

[7] Saeed Javanmardi, Mohammad Shojafar, Shahdad Shariatmadari, et al., "FR trust: a fuzzy reputation-based model for trust management in semantic P2P grids," *International Journal of Grid and Utility Computing*, vol. 6, no. 1, pp. 57-66, 2015. Article (CrossRef Link).

[8] Samaher AI-Janabi, Ibrahim AI-Shourbaji, Mohammad Shojafar, et al., "Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications," *Egyptian Informatics Journal*, 2016. Article (CrossRef Link).

[9] Chae Hoon Lim and Tymur Korkishko, "mCrypton – A lightweight block cipher for security of low-cost RFID tags and sensors," in *Proc. of 6th Information Security Applications Workshop*, pp. 243-258, August 22-24, 2006. Article (CrossRef Link).

[10] Hamid Mala, Mohsen Shakiba and Mohammad Dakhilalian, "New impossible differential attacks on reduced-round Crypton," *Computer Standards & Interfaces*, vol. 32, no. 4, pp. 222-227, January, 2010. Article (CrossRef Link).

[11] Yuechuan Wei, Chao Li and Bing Sun, "Related-key impossible differential cryptanalysis on Crypton and Crypton v1.0," in *Proc of the World Congress on Internet Security*, pp. 227-232, February 21-23, 2011. Article (CrossRef Link).

[12] Jinkeon Kang, Kitae Jeong, Jaechul Sung, et al., "Collision Attacks on AES-192/256, Crypton-192/256, mCrypton-96/128, and Anubis," *Journal of Applied Mathematics*, vol. 2013, pp. 1-10, 2013. Article (CrossRef Link).

[13] Junghwan Song, Kwanhyung Lee and Hwanjin Lee, "Biclique Cryptanalysis on the Full Crypton-256 and mCrypton-128," *Journal of Applied Mathematics*, vol. 2014, pp. 1-10, 2013. Article (CrossRef Link).

[14] Yonglin Hao, Dongxia Bai and Leibo Li, "A Meet-in-the-Middle Attack on Round-Reduced mCrypton Using the Differential Enumeration Techniques," in *Proc of the International Conference on Network and System Security*, pp. 166-183, October 15-17, 2014. Article (CrossRef Link).

[15] Mohsen Shakiba, Mohammad Dakhilalian and Hamid Mala, "Cryptanalysis of mCrypton-64," *Internation Journal of Communication Systems*, vol. 28, no. 8, pp. 1401-1418, 2015. Article (CrossRef Link).

[16] Mohsen Shakiba, Mohammad Dakhilalian and Hamid Mala, "Non-isomorphic biclique cryptanalysis of full-round Crypton," *Computer Standards & Interfaces*, vol.41, pp. 72-78, 2015. Article (CrossRef Link).

[17] Kitae Jeong, HyungChul Kang, Changhoon Lee, et al., "Weakness of lightweight block ciphers mCrypton and LED against biclique cryptanalysis," *Peer-to-Peer Networking and Applications*, vol. 8, no. 4, pp. 716-732, 2015. Article (CrossRef Link).

[18] Yonglin Hao. "Improved Meet-in-the-Middle Attack on Round-Reduced Crypton-256," *IACR Cryptology ePrint Archive*, 2016. Article (CrossRef Link).

[19] Rongjia Li and Chenhui Jin. “Improved meet-in-the-middle attacks on Crypton and mCrypton,” *IET Information Security*, vol. 11, no. 2, pp. 97-103, 2017. Article (CrossRef Link).
[20] Patrick Derbez and Pierre-Alain Fouque, “Automatic search of meet-in-the-middle and impossible differential attacks,” in *Proc of the CRYPTO 2016*, pp. 157-184, August 14-18, 2016. Article (CrossRef Link).
[21] Whitfield Diffie and Martin E. Hellman, “Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard,” *IEEE Computer*, vol. 10, no. 6, pp. 74-84, 1977. Article (CrossRef Link).
[22] Hüseyin Demirci and Ali Aydin Selçuk, “A Meet-in-the-Middle Attack on 8-Round AES,” in *Proc. of 15th Fast Software Encryption Workshop*, pp. 116-126, February 10-13, 2008. Article (CrossRef Link).
[23] Orr Dunkelman, Nathan Keller and Adi Shamir, “Improved Single-Key Attacks on 8-Round AES-192 and AES-256,” *Journal of Cryptology*, vol. 28, no. 3, pp. 397-422, 2015. Article (CrossRef Link).
[24] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean, “Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting,” in *Proc. of Eurocrypt 2013*, pp. 371-387, May 26-30, 2013. Article (CrossRef Link).
[25] Leibo Li, Keting Jia and Xiaoyun Wang, “Improved Single-Key Attacks on 9-Round AES-192/256,” in *Proc. of 21st Fast Software Encryption Workshop*, pp. 127-146, March 3-5, 2014. Article (CrossRef Link).
[26] Rongjia Li and Chenhui Jin, “Meet-in-the-middle attacks on 10-round AES-256,” *Des. Codes Cryptology*, vol. 80, no. 3, pp. 459-471, 2016. Article (CrossRef Link).

Jingyi Cui is a M.S. candidate of Zhengzhou Information Science and Technology. His main research interests include the design and cryptanalysis of symmetric cipher.

Jiansheng Guo is a professor of Zhengzhou Information Science and Technology. His main research interests include the theory of information security and quantum cryptology.

Yanyan Huang received M.S. degree in Zhengzhou Information Science and Technology. Her main research insterest includes information theory.

Yipeng Liu is a M.S. cnadidate of Zhengzhou Information Science and Technology. His main research interests include information theory and quantum cryptology.