A Meet-in-the-Middle Attack on Round-Reduced mCrypton Using the Differential Enumeration Technique

Yonglin Hao[1]⋆, Dongxia Bai[1], Leibo Li[2]

[1] Department of Computer Science and Technology, Tsinghua Universtiy, Beijing 100084, China
haoyl12@mails.tsinghua.edu.cn, baidx10@mails.tsinghua.edu.cn

[2] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, School of Mathematics, Shandong University, Jinan, 250100, China
lileibo@mail.sdu.edu.cn

Abstract. This paper describes a meet-in-the-middle (MITM) attack against the round reduced versions of the block cipher mCrypton-64/96/128. We construct a 4-round distinguisher and lower the memory requirement from 2^{100} to 2^{44} using the differential enumeration technique. Based on the distinguisher, we launch a MITM attack on 7-round mCrypton-64/96/128 with complexities of 2^{44} 64-bit blocks and 2^{57} encryptions. Then we extend the basic attack to 8 rounds for mCrypton-128 by adding some key-bridging techniques. The 8-round attack on mCrypton-128 requires a time complexity 2^{100} and a memory complexity 2^{44}. Furthermore, we construct a 5-round distinguisher and propose a MITM attack on 9-round mCrypton-128 with a time complexity of 2^{115} encryptions and a memory complexity of 2^{113} 64-bit blocks.

1 Introduction

mCrypton is a 64-bit block cipher introduced in 2006 by Lim and Korkishko [1]. It is a reduced version of Crypton [2]. It is specifically designed for resource-constrained devices like RFID tags and sensors in wireless sensor networks. According to key length, mCrypton has three versions namely mCrypton-64/96/128.

Quite a few methods of cryptanalysis were applied to attack mCrypton. Under the related-key model, there are two main results. Park [3] launched a related-key rectangle attack on 8-round mCrypton-128 in the year 2009. Then, in 2012, Mala, Dakhilalian and Shakiba [4] gave a related-key impossible differential cryptanalysis on 9-round mCrypton-96/128. These related-key attacks are important basis in estimating the security of a block cipher, but they are not regarded as a real threat to the application of the cipher in practice since they require a powerful assumption that the adversary can ask to modify the unknown key used in the encryption.

For the attacks under the single-key model, there are only two biclique results on mCrypton: [5] managed to attack mCrypton-96/128 and [6] further adapted the methods to all three versions. Like the biclique result on AES [7], the two attacks mount to the full mCrypton but only with a marginal complexity over exhaustive search. In this paper, we try to attack mCrypton under the single-key model using the meet-in-the-middle method.

The meet-in-the-middle (MITM) attack was first introduced by Diffie and Hellman in 1977 [8]. In the past decade, the MITM scenario has become one of the most fruitful cryptanalysis method. It has been used to analyze block ciphers such as DES [9], KASUMI [10], IDEA [11],XTEA [12], KTANTAN [13] and Camellia [14,15]. It also shows good efficiency in the cryptanalysis of hash functions [16,17,18] and is adapted to attack against public key cryptosystem NTRU [19].

Among all the results of MITM attack, the most impressive ones come from the cryptanalysis on AES block cipher in single-key setting [20,21,22,23,24].

Demirci and Selçuk launched the first MITM attack on AES at FSE 2008 [20]. They constructed a 5-round distinguisher and managed to analyze 7-round AES-192 and 8-round AES-256 using data/time/memory tradeoff. Their attack needs a small data complexity of 2^{32}. But its memory complexity reaches 2^{200} since it requires to store a precomputation determined by 25 intermediated variable bytes. The number of parameters

⋆ Corresponding author.

can be reduced to 24 by storing the differentials instead of values in the precomputation table. Although modifications was made in [21] and [22], the crisis of memory requirement remained severe.

At ASIACRYPT 2010, Dunkelman, Keller and Shamir [23] introduced the differential enumeration and multiset ideas to MITM attacks and reduced the high memory complexity in the precomputation phase. They proved that if a pair conforms a truncated differential characteristic, the number of desired 24 intermediate variable bytes will descend to 16. Furthermore, at EUROCRYPT 2013, Derbez, Fouque and Jean [24] modified Dunkelman et al.'s attack with the rebound-like idea. They proved that many values in the precomputation talbe are not reached under the constraint of the truncated differential. They further lower the number of desired intermediate variable bytes to 10 and diminish the size of precomputation table by a large scale. Based on the 4-round distinguisher, they gave the most efficient attacks on 7-round AES-128 and 8-round AES-192/256. They also introduced a 5-round distinguisher to analyze 9-round AES-256. In this paper, we apply [24]'s method to mCrypton.

Table 1. Summary of the Attacks on mCrypton-64/96/128 under the Single-Key Model.

Version	Rounds	Data	Time	Memory	Method	Reference
64	**7**	$\mathbf{2^{57}}$	$\mathbf{2^{57}}$	$\mathbf{2^{44}}$	**MITM**	**Section 4**
	12	2^{48}	$2^{63.38}$	–	Biclique	[6]
96	**7**	$\mathbf{2^{57}}$	$\mathbf{2^{57}}$	$\mathbf{2^{44}}$	**MITM**	**Section 4**
	12	$2^{27.54}$	$2^{94.09}$	2^{20}	Biclique	[5]
	12	2^{48}	$2^{94.81}$	–	Biclique	[6]
128	**7**	$\mathbf{2^{57}}$	$\mathbf{2^{57}}$	$\mathbf{2^{44}}$	**MITM**	**Section 4**
	8	$\mathbf{2^{57}}$	$\mathbf{2^{100}}$	$\mathbf{2^{44}}$	**MITM**	**Section 5**
	9	$\mathbf{2^{57}}$	$\mathbf{2^{115}}$	$\mathbf{2^{113}}$	**MITM**	**Section 6**
	12	$2^{20.1}$	$2^{125.84}$	2^{20}	Biclique	[5]
	12	2^{48}	$2^{126.26}$	–	Biclique	[6]

Our contribution. We construct a 4-round distinguisher of mCrypton and, using the differential enumeration technique, we prove that such a distinguisher can be determined by 11 intermediate variable nibbles. Based on these ideas, we launch a MITM attack on 7 rounds for all three versions of mCrypton, which recovers 36 subkey bits with a low memory complexity of 2^{44} 64-bit blocks. Then, we find some properties of key schedule and extend the basic attack to 8 rounds for mCrypton-128. The 8-round attack recovers 100 subkey bits. Furthermore, we construct a 5-round distinguisher and mount to 9 rounds for mCrypton-128.. This 9-round attack can recover 116 subkey bits with a time complexity of 2^{115} and a memory complexity 2^{113}. Table 1 summarizes our results along with the other previous results of mCrypton-64/96/128 under the single-key model.

Organization of the Paper. Section 2 provides the description of the block cipher mCrypton and some related works. Section 3 describes the 4-round distinguisher of our basic attack and the way in which the differential enumeration method lower the memory complexity. Section 4 describes our basic MITM attack on 7-round mCrypton-64/96/128. In Section 5, we extend the basic attack to 8-round mCrypton-128 using some key bridging techniques. Then, in Section 6 we present a 5-round distinguisher and attack 9-round mCrypton-128. Finally, we summarize our paper in Section 7.

2 Preliminary

This part contains some background information of our attack. It also gives the notations and units used in this article. As is commonly accepted, the plaintexts are denoted by p and ciphertexts by c.

2.1 Description of mCrypton

mCrypton is a 64-bit lightweight block cipher based on SPN design. It consists of 16 4-bit nibbles which are represented by a 4×4 matrix as follows:

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix} \tag{1}$$

It has three versions, categorized by key length, namely mCrypton-64/96/128. All the three versions have 12 rounds and each round consists of 4 transformations as follows.
Nonlinear Substitution γ. This transformation consists of nibble-wise substitutions using four 4-bit S-boxes $S_i (0 \leq i \leq 3)$. The four S-boxes has relationship:

$$S_0 = S_2^{-1}, \quad S_1 = S_3^{-1}.$$

According to our experiments, the S-boxes of mCrypton have the same property with the S-box of AES (Property 1).

Property 1. Given Δ_i and Δ_o two non-zero differences in $\mathbb{F}_{16}$, the equation

$$S_t(x) \oplus S_t(x \oplus \Delta_i) = \Delta_o, \quad \forall t \in [0, 3]$$

has one solution on average.

Bit Permutation π. The bit permutation transformation π has the same function with the MixColumns transformation of AES. It mixes each column of the 4×4 matrix A. For column $i (0 \leq i \leq 3)$, it uses the corresponding column permutations π_i. Suppose

$$A = (A_0, A_1, A_2, A_3)$$

where A is the 4×4 matrix and A_i is its i-th column. Then, we have

$$\pi(A) = (\pi_0(A_0), \pi_1(A_1), \pi_2(A_2), \pi_3(A_3)).$$

According to [1], each π_i is defined for nibble columns $a = (a_0, a_1, a_2, a_3)^t$ and $b = (b_0, b_1, b_2, b_3)^t$ by

$$b = \pi_i(a) \Leftrightarrow b_j = \bigoplus_{k=0}^{3} (m_{(i+j+k) mod 4} \bullet a_k).$$

The symbol $\bullet$ means bit-wise AND and the masking nibbles m_i are given by

$$m_0 = 0xe = 1110_2, m_1 = 0xd = 1101_2, m_2 = 0xb = 1011_2, m_2 = 0x7 = 0111_2.$$

π transformation is an involution, which means $\pi = \pi^{-1}$. It has a differential brunch number of 4.
Column-To-Row Transposition τ. This is simply the ordinary matrix transposition. It moves the nibble from the position (i, j) to position (j, i).
Key Addition σ. It is a simple bit-wise XOR operation and resembles the AddRoundKey operation of AES. The r-th round $(1 \leq r \leq 12)$ of mCrypton applied to a 64-bit state x can be denoted by

$$\rho_{k_r}(x) = \sigma_{k_r} \circ \tau \circ \pi \circ \gamma(x).$$

Like AES, mCrypton also performs an initial key addition transformation (σ_{k_0}) before round 1. In addition, mCrypton adds a linear operation $\phi = \tau \circ \pi \circ \tau$ after round 12. So, the whole process of mCrypton encryption is

$$c = \phi \circ \rho_{k_{12}} \circ ... \circ \rho_{k_1} \circ \sigma_{k_0}(p)$$

Since we use some key bridging skills to analyze mCrypton-128, we briefly introduce the key schedule of mCrypton-128:

Key Schedule of mCrypton-128. The 128-bit internal register

$$U = (U_0, U_1, U_2, U_3, U_4, U_5, U_6, U_7)$$

is first initialized with the 128-bit user key. Each $U_i (0 \leq i \leq 7)$ is a 16-bit (4-nibble) word, occupying a row of the 4×4 matrix. Round keys $k_r (0 \leq r \leq 12)$ are computed consecutively as follows:

$$T \leftarrow S(U_0) \oplus C_r, T_i \leftarrow T \bullet M_i$$

$$k_r = (U_1 \oplus T_0, U_2 \oplus T_1, U_3 \oplus T_2, U_4 \oplus T_3)$$

$$U \leftarrow (U_5, U_6, U_7, U_0^{<<3}, U_1, U_2, U_3, U_4^{<<8}).$$

S is the nibble-wise S-box operation using S-box S_0. C_r is the round constant word for round r. Masking words M_i is to take the i-th nibble of a word:

$$M_0 = 0xf000, M_1 = 0x0f00, M_2 = 0x00f0, M_3 = 0x000f.$$

The symbol $X^{<<n}$ means left rotation of a 16-bit word X by n bits.

2.2 Notations and Units

Here, we summarize the notations that we use through this paper.

State $\mathbf{x_r^i}$: The 64-bit mCrypton state is represented by different small letters. Plaintexts and ciphertexts are represented by p and c. In the r-th round, we denote the internal state after σ_{k_r} transformation by x_r, after γ by y_r, after π by z_r and after τ by w_r. k_r represents the round key while u_r is calculated linearly from k_r with $u_r = \pi \circ \tau(k_r)$. The difference of state x is denoted by Δx. Besides, the superscript represents the position that the state lies in a sequence (or set).

Nibble $\mathbf{x[i]}$: We refer to the i-th nibble of a state x by $x[i]$, and use $x[i, \cdots, j]$ for nibbles at positions from i to j. The nibbles of the state is numbered as the matrix in equation (1).

Bit $\mathbf{x[i]|_k}$: Each nibble has 4 bits numbered 4,3,2,1 from left to right. If we refer to bit k of nibble $x[i]$, we denote it by $x[i]|_k$.

Bit-wise operators:

$\|$ concatenate two strings of bits.

$\oplus$ bit-wise XOR.

$\bullet$ bit-wise AND.

In this paper, memory complexities of our attacks are measured by the number of 64-bit mCrypton blocks and time complexities by mCrypton encryptions (decryptions).

2.3 The Related Works

From the generic view of meet-in-the-middle attack, the cipher E_K is treated as the combination of three parts $E_K = E^2_{K_2} \circ E^m \circ E^1_{K_1}$. The E^m part in the middle has some particular property (such as a differential characteristic), according to which we can identify the correct key by finding the appearance of the property under each guess of subkey (K_1, K_2). The following definition will be used in this part.

Definition 1. ***(σ-set of AES, [25])*** *The σ-set is a set of 256 intermediate states of AES that one byte traverses all values (the active byte) and the other bytes are constants (the inactive bytes).*

We denote the σ-set by $(x^0, \cdots, x^{255})$. After we encrypt the σ-set by an encryption function E_K, the i-th byte of the output values will form a 2048-bit ordered sequence $(E_K(x^0)[i], \cdots, E_K(x^{255})[i])$. The sequences with particular properties will be stored in a precompted lookup table for distinguishing the correct key guess during the attack.

In the first MITM attack on AES, Demirci et al. [20] build a distinguisher in E^m associated with σ-set. When a σ-set is encrypted, a certain byte of the 256 output values will form an ordered sequence. Demirci et al. found that such an ordered sequence can be expressed as a function of 25 (or 24 if they only store the differences rather than the values) intermediate byte parameters of E^m. In precomputation phase of their attack, the adversary precomputes the $2^{8\times 25}$ ordered sequences and stores them in a table. In the online phase, the adversary mounts an attack by guessing the value of K_1, choosing suitable plaintexts to construct a σ-set of E^m, then partially decrypting the ciphertexts by guessing K_2 to get the corresponding ordered sequence, and checking whether the value lies in the precomputed table. If the value is found in the table, the key guess (K_1, K_2) will be kept, otherwise discarded.

We also consider applying the differential enumeration technique proposed by Dunkelman et al. [23] to reduce the memory requirement of the attack. This technique based on the observation that if a message of the σ-set belongs to a pair conforming a spacial truncated differential characteristic, the possible values of the ordered sequence will be restricted to a small subset of the value space. The essence of this technique is fixing some values of intermediate parameters utilizing the truncated differential so that the size of the precomputed table can be diminished by a large scale. On the other hand, additional steps has to be taken in the online phase in order to find a pair satisfying the truncated differential characteristic because the σ-set is constructed only for this kind of pairs. Apparently, the differential enumeration technique reduce the memory requirement but also increase the data and time complexity. It is noticeable that Derbez et al. [24] improved this technique at EUROCRYPT 2013. They further reduced the possible values in the precomputed table by introducing some rebound-like ideas.

3 The 4-Round Distinguisher and The Differential Enumeration Tehcnique

The meet-in-the-middle strategy combined with the differential enumeration technique is the basis of our attack. Imitating those of AES, we define the σ-set of the mCrypton as Definition 2.

Definition 2. ***(σ-set of mCrypton).*** *A σ-set is a set of 16 64-bit mCrypton-states that are all different in one nibble (the active nibble) and all equal in the other state nibbles (the inactive nibbles).*

In our basic MITM attack, the middle part E^m starts from x_1 and ends at x_5. So, in the following parts of this paper, we denote the σ-set with an active nibble at position $j (0 \leq j \leq 15)$ by

$$A_j = (x_1^0, ..., x_1^{15}), \tag{2}$$

and the corresponding ordered sequence constituted by the l-th nibble ($l \in [0, 15]$) of x_5 is denoted by

$$B_j^l = (\Delta^1 x_5[l], \cdots, \Delta^{15} x_5[l]), \tag{3}$$

where $\Delta^i x_r = x_r^i \oplus x_r^0 (1 \leq i \leq 15, 0 \leq r \leq 12)$.

Proposition 1 shows that 25 intermediate variable nibbles are required to deduce B_j^l from A_j. However, if the x_1^0 of σ-set A_0 belongs to a pair conforming the truncated differential characteristic in Figure 1, we can prove that the corresponding sequence B_0^0 can only have 2^{44} values determined by 11 nibble parameters (Proposition 2). This is the differential enumeration method used in [23] and [24] to lower the memory complexities of their attacks on AES.

Proposition 1. $\forall j \in [0, 15]$ *and* $\forall l \in [0, 15]$. *Let the σ-set be*

$$A_j = (x_1^0, ..., x_1^{15})$$

Then, the corresponding sequence

$$B_j^l = (\Delta^1 x_5[l], \cdots, \Delta^{15} x_5[l])$$

can be fully determined by 25 nibble parameters:

- *1 nibble of x_1^0.*
- *The full 16-nibble state x_3^0;*
- *4 nibbles of x_2^0.*
- *4 nibbles of x_4^0.*

Proof. We just let $j = 0$ and $l = 0$. Then, the 25 nibbles required are:

$$x_1^0[0], x_2^0[0,1,2,3], x_3^0[0,\cdots,15], x_4^0[0,4,8,12].$$

For the t-th element of B_0^0 ($t \in [1,15]$), the difference $\Delta^t x_5[0]$ can be deduced from $x_4^0[0,4,8,12]$ and $\Delta^t x_4[0,4,8,12]$.

$\Delta^t x_4[0,4,8,12]$ requires the knowledge of $x_3^0[0,\cdots,15]$ and $\Delta^t x_3[0,\cdots,15]$.

$\Delta^t x_3[0,...,15]$ is generated linearly from $\Delta^t y_2[0,\cdots,3]$, which can be deduced from $x_2^0[0,1,2,3]$ and $\Delta^t x_2[0,1,2,3]$.

$\Delta^t x_2[0,1,2,3]$ is generated linearly from $\Delta^t y_1[0]$, which requires the knowledge of $x_1^0[0]$ and $\Delta^t x_1[0]$. $\Delta^t x_1[0]$ can be deduced directly from A_0. Hence, all the nibble parameters required are: $x_1^0[0]$, $x_2^0[0,1,2,3]$, $x_3^0[0,\cdots,15]$, $x_4^0[0,4,8,12]$. □

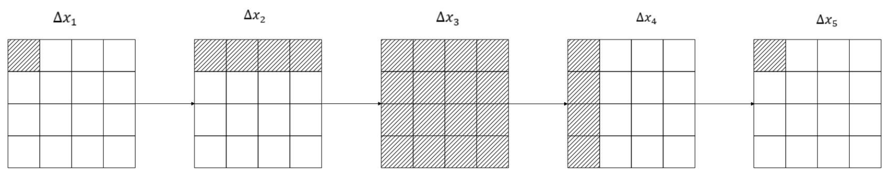

Figure 1. The 4-round truncated differential characteristic. Dashed nibbles are active.

Proposition 2. *If the x_1^0 of a σ-set A_0 belongs to a pair satisfying the differential characteristic in Figure 1, the corresponding sequence B_0^0 can only take 2^{44} values.*

Proof. According to Proposition 1, B_0^0 is determined by 25 nibbles namely:

$$x_1^0[0], x_2^0[0,1,2,3], x_3^0[0,\cdots,15], x_4^0[0,4,8,12].$$

But if x_0 of A_0 belongs to one of the pairs conforming the truncated differential characteristic in Figure 1, the corresponding sequence B_0^0 can only take 2^{44} values determined by 11 nibbles namely

$$x_1^0[0], \Delta x_1[0], x_2^0[0,1,2,3], x_4^0[0,4,8,12], \Delta z_4[0],$$

where Δ refers to the difference of the pair conforming the differential characteristic.

The knowledge of $x_1^0[0]$ and $\Delta x_1[0]$ is sufficient to deduce $\Delta x_2[0,1,2,3]$. Combining $x_2^0[0,1,2,3]$ and $\Delta x_2[0,1,2,3]$, we get the 16-nibble difference Δx_3.

Similarly, we can deduce $\Delta y_4[0,4,8,12]$ from $\Delta z_4[0]$. Adding the knowledge of $x_4^0[0,4,8,12]$, the 16-nibble differential Δy_3 is determined.

Since $y_3 = \gamma(x_3)$, according to the property of mCrypton S-boxes (Property 1), we can only get one value on average for each of the 16-nibble state x_3 using super-box matches technique [26].

This is the way we deduce the sequence B_0^0 from the σ-set A_0. □

The 4-round truncated differential character in Figure 1 is the distinguisher that we use in the following section.

4 The Basic Attack on 7-Round mCrypton-64/96/128

In this part, we describe our basic attack on 7-round mCrypton-64/96/128. This attack can recovery 36 subkey bits. The complete differential path used in this attack can be seen in Figure 4 in Appendix A. This attack is composed of two phases: the precomputation phase and the online phase.

Precomputation Phase: In the precomputation phase, we set up a lookup table containing 2^{44} ordered sequences described as Proposition 2. The procedure is similar to the proof of Proposition 1 and Proposition 2, and is described as follows.

1. For each 44-bit string $x_1^0[0] \| \Delta x_1[0] \| x_2^0[0, \cdots, 3] \| x_4^0[0, 4, 8, 12] \| \Delta z_4[0]$, we compute the possible value of the 25 nibbles, namely $x_1^0[0], x_2^0[0, \cdots, 3], x_3^0, x_4^0[0, 4, 8, 12]$, only with which can we determine the ordered sequence B_0^0. The procedure is as follows:
   (a) Compute $\Delta x_2[0, \cdots, 3]$ with the knowledge of $x_1^0[0]$ and $\Delta x_1[0]$;
   (b) Compute Δx_3 with the knowledge of $x_2^0[0, \cdots, 3]$ and $\Delta x_2[0, \cdots, 3]$;
   (c) Compute $\Delta y_4[0, 4, 8, 12]$ with the knowledge of $\Delta z_4[0]$;
   (d) Compute Δy_3 with the knowledge of $\Delta y_4[0, 4, 8, 12]$ and $x_4[0, 4, 8, 12]$;
   (e) With the knowledge of Δy_3 and Δx_3 and using the super-box matches, we can determine the possible values of x_3^0 satisfying $\gamma(x_3^0) \oplus \gamma(x_3^0 \oplus \Delta x_3) = \Delta y_3$. According to Property 1, x_3^0 has only one possible value on average.
2. Now that we have obtained the value of the 25 nibbles namely $x_1^0[0], x_2^0[0, \cdots, 3], x_3^0, x_4^0[0, 4, 8, 12]$. For all the 15 possible values of $\Delta^t x_1^0[0], t \in [1, 15]$, we can compute the t-th element of B_0^0, $\Delta^t x_5[0]$, by executing the following substeps:
   (a) Compute $\Delta^t x_2[0, \cdots, 3]$ with the knowledge of $\Delta^t x_0[0]$ and $x_1^0[0]$;
   (b) Compute $\Delta^t x_3$ with the knowledge of $\Delta^t x_2[0, \cdots, 3]$ and $x_2^0[0, \cdots, 3]$;
   (c) Compute $\Delta^t x_4[0, 4, 8, 12]$ with the knowledge of $\Delta^t x_3$ and x_3^0;
   (d) Compute $\Delta^t x_5[0]$ with the knowledge of $\Delta^t x_4[0, 4, 8, 12]$ and $x_4^0[0, 4, 8, 12]$. And $\Delta^t x_5$ is the t-th element of B_0^0
3. Store all the 2^{44} B_0^0s in a hash table T_s.

Online Phase: In the online phase of this attack, we first find the right pairs satisfying the truncated differential characteristic. Then, for each member of the pairs, we construct its σ-set A_0 and deduce the corresponding ordered sequence B_0^0 through partial encryptions&decryptions. Finally, we check whether the obtained B_0^0 exist in the precomputed lookup table T_s. The detailed procedure is as follows.

1. Encrypt 2^{41} structures of 2^{16} plaintexts such that $p[0, 4, 8, 12]$ takes all values and other nibbles are constants. There are about 2^{31} pairs (p, p') in each structure, so there are 2^{72} pairs in total.
2. Within each structure, select the pairs whose Δc only have difference at positions 0,4,8,12. Since this is a 48-bit filter, approximately 2^{24} of the 2^{72} message pairs will remain after this step.
3. For each remaining pair, assuming that it satisfies the differential path in Figure 4, we do the following substeps.
   (a) Guess the difference value $\Delta x_1[0]$ and linearly deduce $\Delta y_0[0, 4, 8, 12]$.
   (b) For each guess, deduce the possible subkey nibbles $k_0[0, 4, 8, 12]$ with the knowledge of $\Delta y_0[0, 4, 8, 12]$ and $\Delta x_0[0, 4, 8, 12] = \Delta p[0, 4, 8, 12]$. According to Property 1, one $k_0[0, 4, 8, 12]$ value can be acquired on average.
   (c) Guess the difference value $\Delta y_5[0]$ and deduce $\Delta x_6[0, \cdots, 3]$.
   (d) For each guess, deduce the possible $u_7[0, \cdots, 3]$ with the knowledge of $\Delta x_6[0, \cdots, 3]$ and $\Delta y_6[0, 4, 8, 12]$ ($\Delta y_6 = \Delta \tau(c)$). According to Property 1, one $u_7[0, \cdots, 3]$ can be acquired on average.
4. For each deduced subkey $k_0[0, 4, 8, 12] \| u_7[0, \cdots, 3]$, we obtain at least one ordered sequence B_0^0 with the following substeps.
   (a) Select one message of the right pair, denoted by p^0, and deduce its $x_1^0[0]$ through partial encryption.
   (b) Then, let t traverse through $[1, 15]$ so that we can compute $\Delta^t x_1[0] = x_1^0[0] \oplus t$ and deduce plaintexts p^t with the knowledge of $\Delta^t x_1[0]$ and p^0 through partial decryption. At this point, we have acquired the plaintexts $(p^0, \cdots, p^{15})$ corresponding to the σ-set $(x_1^0, \cdots, x_1^{15})$.

(c) Query the ciphertexts of $(p^0, \cdots, p^{15})$ and deduce the sequence

$$(x_6^0[0, \cdots, 3], \cdots, x_6^{15}[0, \cdots, 3]) \tag{4}$$

through partial decryption with the knowledge of $u_7[0, \cdots, 3]$.

(d) Guess subkey $u_6[0]$ and deduce the ordered sequence $B_0^0 = (\Delta^1 x_5[0], \cdots, \Delta^{15} x_5[0])$ from (4) through partial decryption.

5. Identify the right subkeys $k_0[0, 4, 8, 12] \| u_6[0] \| u_7[0, \cdots, 3]$ by verifying whether the sequence B_0^0 exists in the precomputed lookup table T_s. If $B_0^0 \in T_s$, the key guesses are correct with high probability. The error rate is $2^{44-60} = 2^{-16}$ to be precise.

Complexity analysis. In the pre-computation phase, T_s contains 2^{44} sequences and each sequence occupies 60 bits of space. So the memory complexity of this attack is dominated by T_s' 2^{44} 64-bit blocks. Since each sequence has 15 nibbles, it requires $2^{44} \times 15 \approx 2^{48}$ encryptions to construct the lookup table. The time complexity of the online phase is dominated by step 1 which involves encrypting $2^{41} \times 2^{16} = 2^{57}$ plaintexts. So the time and data complexity of our attack are both 2^{57}.

5 Extend the Basic Attack to 8 Rounds for mCrypton-128

Using the key bridging technique, we can further attack 8-round mCrypton-128. According to the key schedule of mCrypton-128, the knowledge of k_8 can deduce some bits in k_0 (Proposition 3) with which we can lower the time complexity of the online phase.

Proposition 3. *By the key schedule of mCrypton-128, knowledge of the entire 16-nibble k_8 allows deduce $k_0[6]$, 3 bits of $k_0[2]$ and 1 bit of $k_0[14]$.*

Proof. According to the key schedule of mCrypton-128, the relationship of the 8 bits are:

$$k_0[2]|_{4,3,2} = k_8[3]|_{3,2,1} \tag{5}$$
$$k_0[6]|_{4,3,2} = k_8[7]|_{3,2,1} \tag{6}$$
$$k_0[6]|_{1} = k_8[4]|_{4} \tag{7}$$
$$k_0[14]|_{1} = k_8[12]|_{4} \tag{8}$$

The readers may refer to [1] for the detailed key schedule of mCrypton-128. □

In order to make full use of Proposition 3, we deliberately change the form of the truncated differential characteristic of this attack to the one in Figure 2. The σ-set of this attack has an active nibble at position 8 and is denoted by A_8. The corresponding ordered sequence is composed of nibble 0 of x_5 and is denoted by B_8^0. The complete differential characteristic of this attack can be seen in Figure 5 in Appendix A.

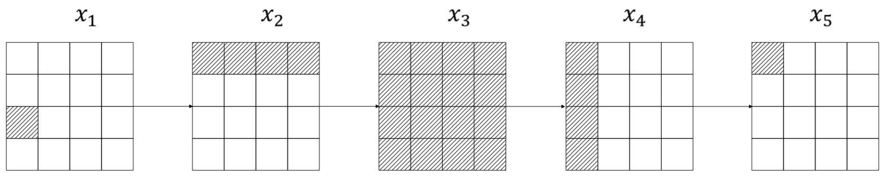

Figure 2. The differential characteristic for 8-round mCrypton-128.

The precomputation phase of this 8-round attack is identical to that of the basic attack. In this phase, we construct a lookup table T_s containing 2^{44} possible values of B_8^0 determined by 11 intermediate variable nibbles namely

$$x_1^0[8], \Delta x_1[12], x_2^0[0, 1, 2, 3], x_4^0[0, 4, 8, 12], \Delta z_4[0].$$

In the online phase, additional subkey has to be deduced so that we can extend the basic 7-round attack to 8 rounds. The procedure of the online phase is as follow.

1. Encrypt 2^{41} structures of 2^{16} plaintexts with active nibbles at positions 2,6,10,14. There are about 2^{31} pairs (p, p') in each structure, so there are 2^{72} pairs in total.
2. For each pair, do the following substeps.
   (a) Guess the difference $\Delta y_6[0, \cdots, 3]$ and compute the subkey k_8 using the super-box matches. Since there are 2^{16} possible values of $\Delta y_6[0, \cdots, 3]$ and each can deduce one k_8 one average (Proposition 1), 2^{16} k_8 values are obtained in this step.
   (b) Deduce $k_0[6]$ from k_8 using Proposition 3 and obtain $\Delta y_0[6]$ through partial encryption. Discard the keys if $\Delta y_0[6]|_4 \neq 0$ because
$$\Delta y_0[6] = m_3 \bullet \Delta z_0[2], \quad m_3 = 0111_2. \tag{9}$$
   This is a one-bit filter so there are 2^{15} subkey guesses left.
   (c) Now that we have acquired $z_0[2]|_{3,2,1}$ from (9), we guess $z_0[2]|_4$ and compute $\Delta y_0[2, 6, 10, 14]$ with which we can deduce $k_0[2, 6, 10, 14]$. Discard the guesses violating equations (6) and (8) of Proposition 3. This step involves a 1-bit guess and a 4-bit filter, so there are about $2^{15+1-4} = 2^{12}$ subkey guesses remain.
   (d) Guess $\Delta y_5[0]$ and deduce the 2^4 possible values of $u_7[0, \cdots, 3]$. At this point, we have acquired 2^{16} key guesses of $k_0[2, 6, 10, 14]\|u_7[0, \cdots, 3]\|k_8$.
3. For each subkey, select a member of the right pair, guess $u_6[0]$ and construct the sequence B_8^0 through partial encryptions & decryptions.
4. Identify the right guess by checking whether B_8^0 exist in T_s.

Complexity analysis. Similar to the basic attack, the data complexity of this attack is 2^{57} and the memory complexity is still dominated by the pre-computed lookup table, which is 2^{44} 64-bit blocks to be precise. The time complexity of this attack is dominated by the 3rd step of the online phase which involves a 4-bit guess and 16 encryptions/decryptions to construct B_8^0. Since this step has to be executed on each of the 2^{20} subkey guesses within each of the 2^{72} pairs, the time complexity of this attack is $2^{72} \times 2^{20} \times 2^4 \times 16 = 2^{100}$. The 8-round attack recovers 100 subkey bits namely $k_0[2, 6, 10, 14]\|u_6[0]\|u_7[0, \cdots, 3]\|k_8$.

6 9-Round Attack on mCrypton-128

The 15-nibble sequences used in the previous attacks can not provide enough information to identify the correct subkey guesses in the 9-round attack on mCrypton-128. In the 9-round attack, we consider the σ-set with 2 active nibbles containing 256 64-bit blocks and its corresponding ordered sequence consists of 255 nibbles occupying 1020 bits of space. The key bridging technique used in this attack is interpreted as Proposition 4.

Proposition 4. *By the key schedule of mCrypton-128, the knowledge of the entire 16-nibble k_9 allows to deduce $k_0[0, 3]$*

Proof. According to the key schedule, we have
$$k_0[3]|_{4,3,2} = k_9[12]|_{3,2,1}$$
$$k_0[3]|_1 = k_9[13]|_4$$
$$k_0[0] = S_0((k_9[8]|_{2,1}\|k_9[9]|_{4,3})) \oplus (k_9[13]|_{3,2,1}\|k_9[14]|_4) \oplus 1$$
These relationships can be deduced easily from the key schedule. □

To fully utilize the key bridging technique, we select the σ-set with active nibbles at positions 0 and 12, denoted by
$$A_{0,12} = \{x_1^0, x_1^1, ..., x_1^{255}\}.$$
As is described in Subsection 2.1, the differential brunch of the π operation in mCrypton is 4. So, we deliberately assign that the x_1^0 of $A_{0,12}$ belongs to a pair satisfying the 5-round truncated differential characteristic

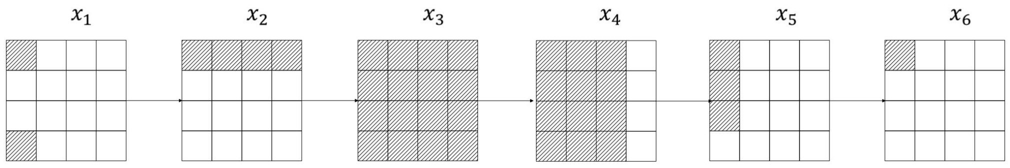

Figure 3. The 5-round differential characteristic for attacking 9-round mCrypton-128.

shown in Figure 3. The ordered sequence corresponding to $A_{0,12}$ consists of $\Delta^t x_6[0] (1 \leq t \leq 255)$ and is denoted by

$$B^0_{0,12} = (\Delta^1 x_6[0], \cdots, \Delta^{255} x_6[0]).$$

Similar to Proposition 1 and 2, the sequence $B^0_{0,12}$ is determined by 42 intermediate variable nibbles namely:

$$x^0_1[0, 12], x^0_2[0, \cdots, 3], x^0_3, x^0_4, x^0_5[0, 4, 8, 12]$$

and the number of desired nibbles can be lowered to 29 with the help of the truncated differential characteristic in Figure 3 by using the differential enumeration method. The 29 decisive nibbles for $B^0_{0,12}$ are namely

$$x^0_1[0, 12], \Delta x_1[0, 12], x^0_2[0, 1, 2, 3], x^0_3, x^0_5[0, 4, 8, 12], \Delta z_5[0]. \tag{10}$$

However, the differential character can only be satisfied when $\Delta z_5[0] = 0x8 = 1000_2$, which means $\Delta z_5[0]$ can only take 1 rather than 2^4 values. Since $\Delta z_5[0]$ can only take 1 value, $\Delta x_5[0, 4, 8]$ can only take $2^{3\times3} = 2^9$ values, which is also true for $x^0_5[0, 4, 8]$ according to Property 1. So the total number of the targeted ordered sequence is $2^{4\times25+9} = 2^{109}$. This 9-round attack is in high accordance with the 8-round one introduced in Section 5, so we only briefly summarize the whole procedure as follow.

Precompuation phase. Construct the lookup table T_s containing 2^{109} values of $B^0_{0,12}$ determined by 29 nibble parameters listed in (10).

Online phase.

1. Encrypt 2^{25} structures of 2^{32} plaintexts such that $p[\lambda]$, where

$$\lambda = (0, 3, 4, 7, 8, 11, 12, 15),$$

   takes all values and other nibbles are constants. There are 2^{88} pairs in total.
2. For each pair, do the following steps.
   (a) Guess $\Delta y_7[0, \cdots, 3]$ and deduce k_9 using super-box matches. Since there are 2^{16} possible values of $\Delta y_7[0, \cdots, 3]$ and each can deduce averaging one k_9 (Proposition 1), 2^{16} k_9 values are obtained in this step.
   (b) Deduce $k_0[0, 3]$ with the knowledge of k_9 using Proposition 4. Then, compute $\Delta y_0[0, 3]$ and deduce $\Delta y_0[\lambda]$ with the relation between Δy_0 and Δz_0. With the knowledge of $\Delta y_0[\lambda]$, we can further retrieve $k_0[\lambda]$.
   (c) Guess $\Delta y_6[0]$ and deduce 2^3 possible values of $u_8[0, \cdots, 3] \| u_7[0]$, where $u_7[0]$ is deduced from the knowledge of $\Delta x_6[0] = 1000_2$.
3. For each subkey, select a member of the right pair and deduce its $B^0_{0,12}$ through partial decryptions.
4. Refer to the T_s and identify the right guess.

Complexity analysis. The data complexity of the 9-round attack is $2^{25+32} = 2^{57}$. The memory complexity is dominated by the lookup table set up in the precomputation phase, which is $2^{109} \times 1020/64 \approx 2^{113}$ 64-bit blocks to be precise. The time complexity is dominated by the step 3 of the online phase which involves $2^{88} \times 2^{16} \times 2^3 \times 256 = 2^{115}$ encryptions/decryptions. The 9-round attack recovers 116 subkey bits namely $k_0[\lambda] \parallel u_7[0] \parallel u_8[0, \cdots, 3] \parallel k_9$, where $\lambda = (0, 3, 4, 7, 8, 11, 12, 15)$.

7 Conclusion

In this paper, we analyze the lightweight SPN block cipher mCrypton using the meet-in-the-middle (MITM) attack under the single-key model. We use the differential enumeration technique to lower the memory complexity, which used to be the bottleneck of the MITM method. We set up a 4-round distinguisher and manage to launch a basic MITM attack on 7-round mCrypton-64/96/128. Adding some key bridging techniques, we extend the basic attack to 8 rounds for mCrypton-128. We construct a 5-round distinguisher and further mount to 9 rounds for mCrypton-128. The 9-round attack retrieves 116 subkey bits with memory complexity 2^{113} and time complexity 2^{115}.

Acknowledgement. This work has been supported by the National Natural Science Foundation of China (Grant No. 61133013) and by 973 Program (Grant No. 2013CB834205).

References

1. Lim, C.H., Korkishko, T.: mcrypton–a lightweight block cipher for security of low-cost rfid tags and sensors. In: Information Security Applications. Springer (2006) 243–258
2. Lim, C.H.: Crypton: A new 128-bit block cipher. NIsT AEs Proposal (1998)
3. Park, J.H.: Security analysis of mcrypton proper to low-cost ubiquitous computing devices and applications. International Journal of Communication Systems **22**(8) (2009) 959–969
4. Mala, H., Dakhilalian, M., Shakiba, M.: Cryptanalysis of mcryptona lightweight block cipher for security of rfid tags and sensors. International Journal of Communication Systems **25**(4) (2012) 415–426
5. Shakiba, M., Dakhilalian, M., Mala, H.: Non-isomorphic biclique cryptanalysis and its application to full-round mcrypton. IACR Cryptology ePrint Archive **2013** (2013) 141
6. Jeong, K., Kang, H., Lee, C., Sung, J., Hong, S., Lim, J.I.: Weakness of lightweight block ciphers mcrypton and led against biclique cryptanalysis. Peer-to-Peer Networking and Applications (2013) 1–17
7. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique cryptanalysis of the full aes. In: Advances in Cryptology–ASIACRYPT 2011. Springer (2011) 344–371
8. Diffie, W.: Exhaustive crypianalysis of the nbs daia encrypiion siandard. (1977)
9. Dunkelman, O., Sekar, G., Preneel, B.: Improved meet-in-the-middle attacks on reduced-round des. In: Progress in Cryptology–INDOCRYPT 2007. Springer (2007) 86–100
10. Jia, K., Yu, H., Wang, X.: A meet-in-the-middle attack on the full kasumi. IACR Cryptology ePrint Archive **2011** (2011) 466
11. Demirci, H., Selçuk, A.A., Türe, E.: A new meet-in-the-middle attack on the idea block cipher. In: Selected Areas in Cryptography, Springer (2004) 117–129
12. Sekar, G., Mouha, N., Velichkov, V., Preneel, B.: Meet-in-the-middle attacks on reduced-round xtea. In: Topics in Cryptology–CT-RSA 2011. Springer (2011) 250–267
13. Bogdanov, A., Rechberger, C.: A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher ktantan. In: Selected Areas in Cryptography, Springer (2011) 229–240
14. Lu, J., Wei, Y., Pasalic, E., Fouque, P.A.: Meet-in-the-middle attack on reduced versions of the camellia block cipher. In: Advances in Information and Computer Security. Springer (2012) 197–215
15. Chen, J., Li, L.: Low data complexity attack on reduced camellia-256. In: Information Security and Privacy, Springer (2012) 101–114
16. Aoki, K., Sasaki, Y.: Meet-in-the-middle preimage attacks against reduced sha-0 and sha-1. In: Advances in Cryptology-CRYPTO 2009. Springer (2009) 70–89
17. Sasaki, Y.: Meet-in-the-middle preimage attacks on aes hashing modes and an application to whirlpool. In: Fast Software Encryption, Springer (2011) 378–396
18. Sasaki, Y., Aoki, K.: Meet-in-the-middle preimage attacks on double-branch hash functions: Application to ripemd and others. In: Information Security and Privacy, Springer (2009) 214–231
19. Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In: Advances in Cryptology-CRYPTO 2007. Springer (2007) 150–169
20. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round aes. In: Fast Software Encryption, Springer (2008) 116–126
21. Demirci, H., Taşkın, İ., Çoban, M., Baysal, A.: Improved meet-in-the-middle attacks on aes. In: Progress in Cryptology-INDOCRYPT 2009. Springer (2009) 144–156

22. Wei, Y., Lu, J., Hu, Y.: Meet-in-the-middle attack on 8 rounds of the aes block cipher under 192 key bits. In: Information Security Practice and Experience. Springer (2011) 222–232
23. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round aes-192 and aes-256. In: Advances in Cryptology-ASIACRYPT 2010. Springer (2010) 158–176
24. Derbez, P., Fouque, P.A., Jean, J.: Improved key recovery attacks on reduced-round aes in the single-key setting. In: Advances in Cryptology–EUROCRYPT 2013. Springer (2013) 371–387
25. Daemen, J., Rijmen, V.: Aes proposal: Rijndael. (1999)
26. Gilbert, H., Peyrin, T.: Super-sbox cryptanalysis: improved attacks for aes-like permutations. In: Fast Software Encryption, Springer (2010) 365–383

Appendix

A The Complete Differential Characteristics Used in This Article

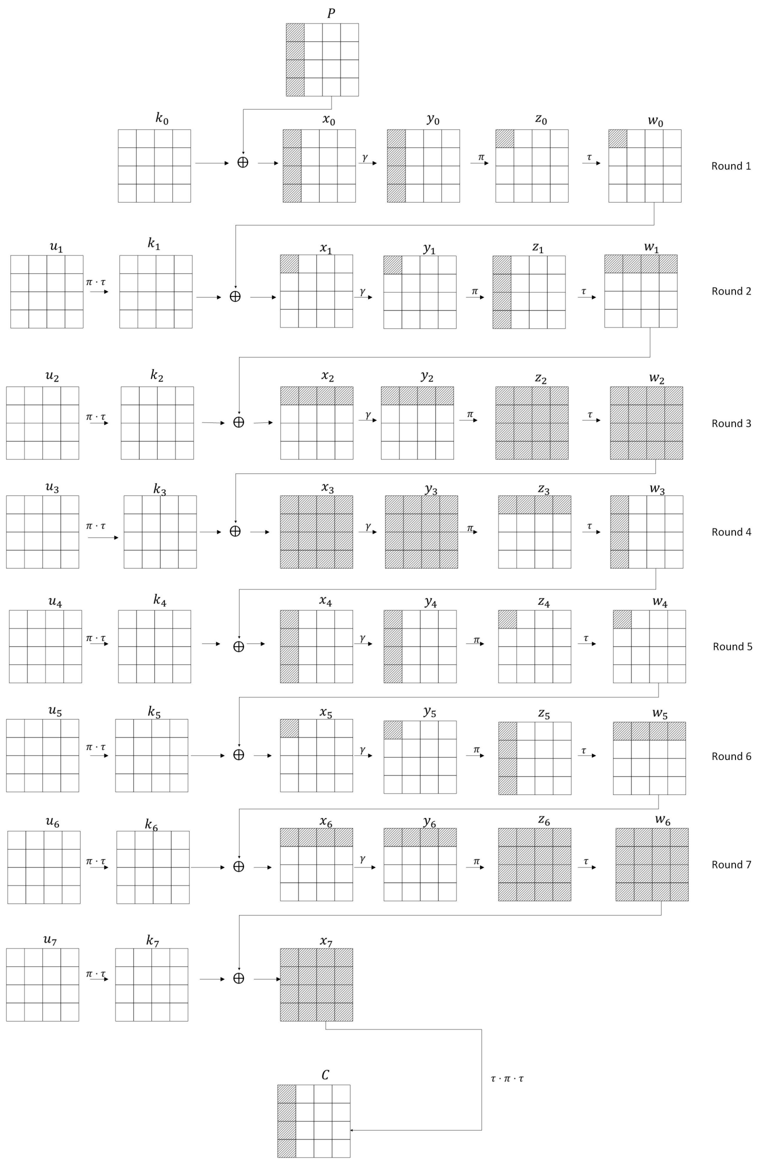

Figure 4. Complete 7-round differential characteristic used in Section 4

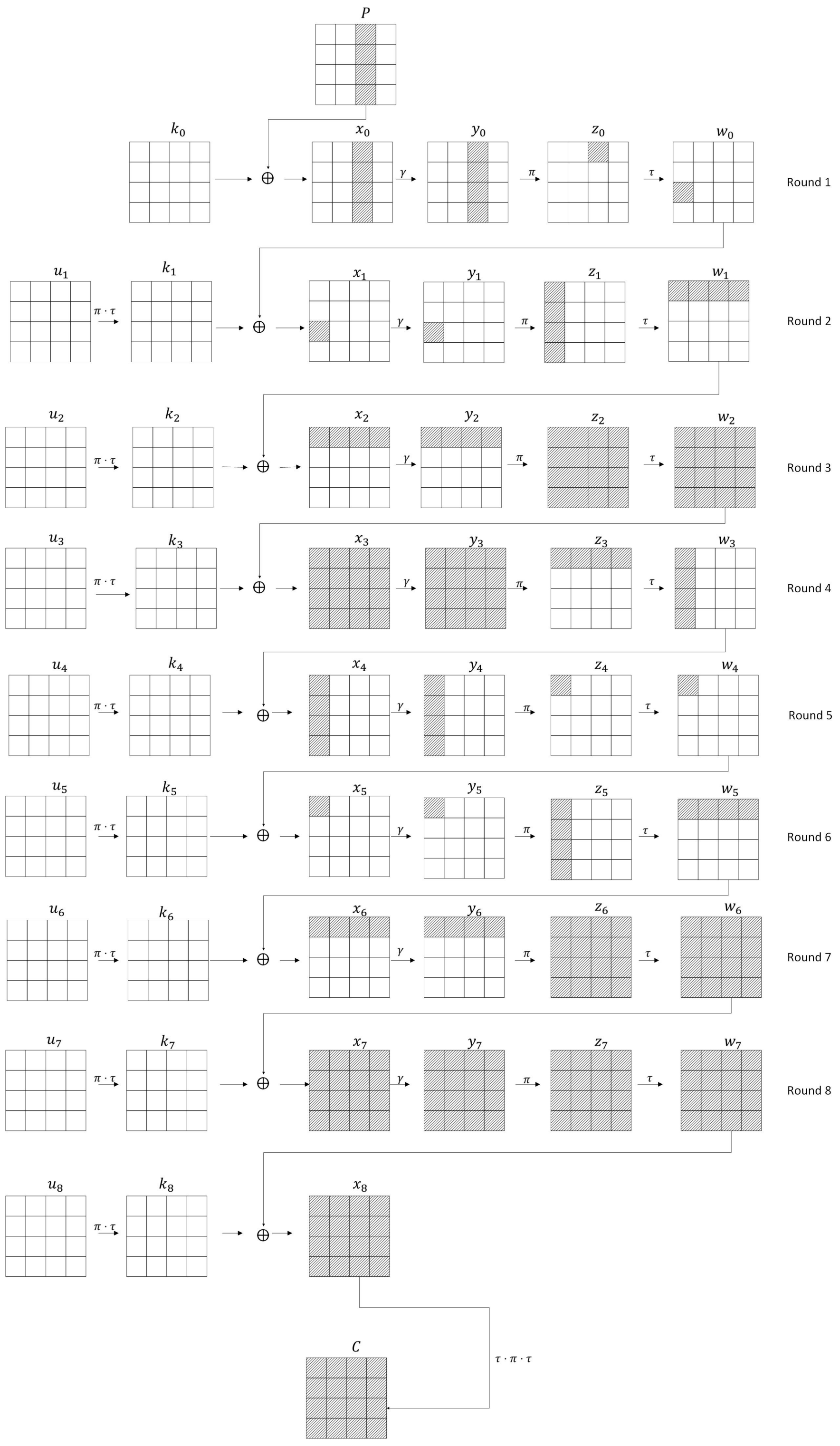
P
k_0
x_0
y_0
z_0
w_0
γ
π
τ
Round 1
u_1 k_1 x_1 y_1 z_1 w_1
$\pi \cdot \tau$
Round 2
u_2 k_2 x_2 y_2 z_2 w_2
Round 3
u_3 k_3 x_3 y_3 z_3 w_3
Round 4
u_4 k_4 x_4 y_4 z_4 w_4
Round 5
u_5 k_5 x_5 y_5 z_5 w_5
Round 6
u_6 k_6 x_6 y_6 z_6 w_6
Round 7
u_7 k_7 x_7 y_7 z_7 w_7
Round 8
u_8 k_8 x_8
$\tau \cdot \pi \cdot \tau$
C

Figure 5. Complete 8-round differential characteristic used in Section 5

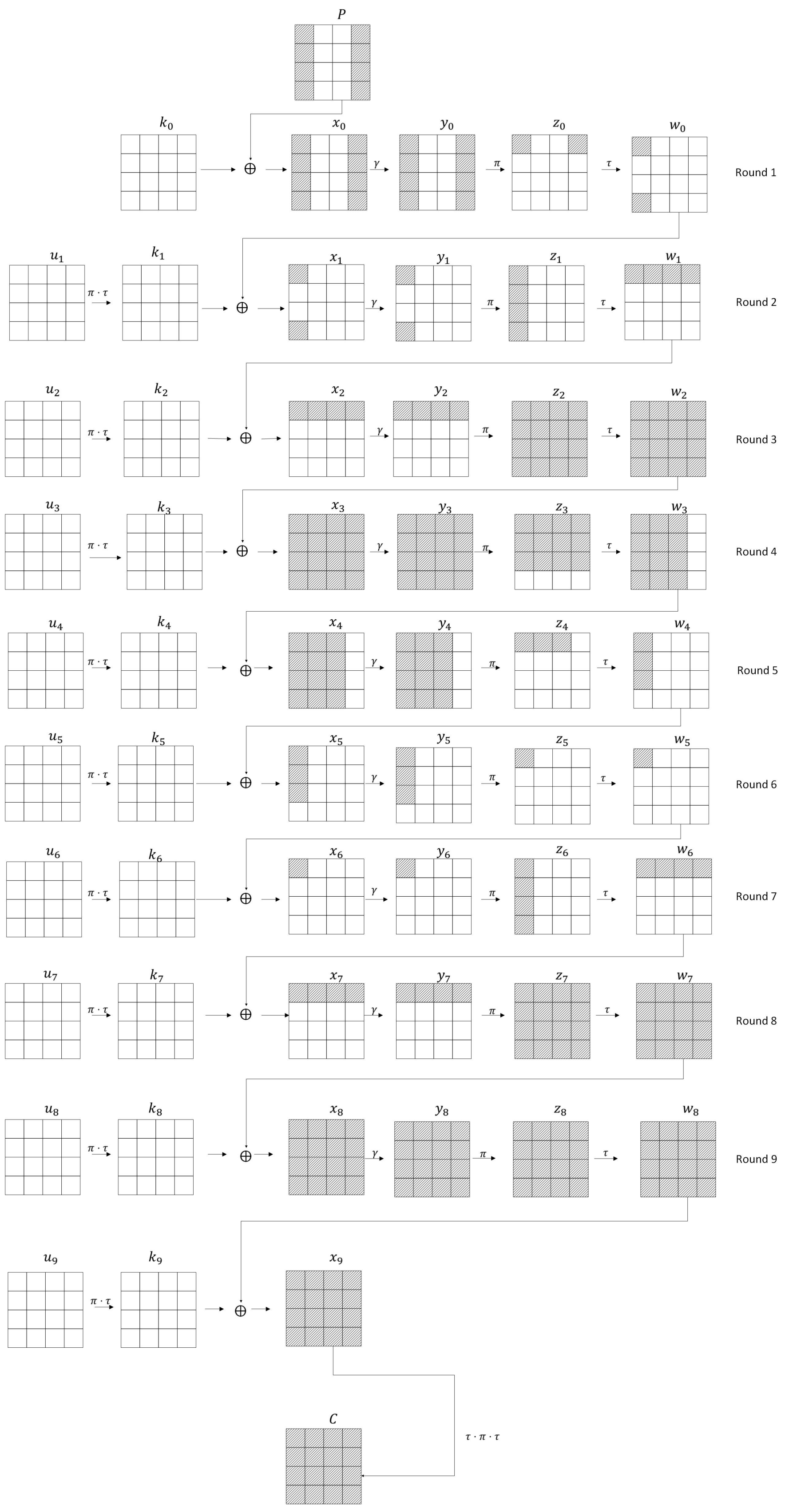

Figure 6. Complete 9-round differential characteristic used in Section 6