Related-Key Impossible Differential Attacks on Crypton

Yuechuan Wei
School of Computer,
National University of Defense Technology, China
wych004@163.com

Chao Li, Bing Sun
Science College,
National University of Defense Technology, China
lichao_nudt@sina.com, happy_come@163.com

Abstract

Crypton is a 12-round block cipher proposed as an AES candidate and Crtpton v1.0 is the revised version. In this paper, we present two related-key impossible differential attacks to reduced-round Crypton and Crypton v1.0. By carefully choosing the relations of keys, constructing some 6-round related-key differential trials and using some observations on the cipher, we first break 9-round Crypton v1.0 and Crypton with 256 bit keys. This fact reflects some weaknesses of the key schedule algorithms of the two versions of Crypton when using 256 bits user keys.

1. Introduction

The cipher Crypton[1] was proposed as a candidate algorithm for the AES. Since the competition, Crypton have attracted much attention for its several favorable features. For example, the encryption and decryption processes are strictly identical, the structure is highly parallelizable and flexible. Moreover, Crypton provides some provable security against differential and linear cryptanalysis. However, due to its too simple round key computations, some minor weaknesses were found by Serge Vaudenay etc. To fix these weaknesses in the key schedule and enhance the security, the designers introduced a modified Crypton[2] with a new key schedule and new S-boxes. This new version denoted by Crypton v1.0. The new key schedule introduces bit and word rotations and round constants for each round key. Besides, the new key schedule runs much faster than one-block encryption.

Impossible differential cryptanalysis is one of the most powerful tool used for block cipher cryptanalysis. Proposed by Biham and Knudsen respectively, this method was first applied to the cipher DEAL[3] and later to Skipjack[4]. The main idea is to specify a differential of probability zero over some rounds of the cipher. Then one can derive the right key by discarding the keys which lead to the impossible differential. Related-key attacks[5] allow the cryptanalyst to obtain plaintext-ciphertext pairs by using unknown but related keys. By observing the possible weaknesses of the encryption and key schedule algorithms, the attackers choose appropriate relation between keys and then predict the encryptions under these keys. The combination of the above two attacks is called related-key impossible differential attack.

The main cryptanalytic results obtained on Crypton so far are as follows. In FSE'99, H'Halluin et al. proposed a modified square attack for 6-round Crypton[6]. In Asiacrypt'99, Seki and Kaneko found that 4 rounds of Crypton has impossible differential, using this fact they gave an attack to 5-round Crypton[7], and later this result was improved to 6-round by Cheon et al. in ICISC 2001[8]. A stochastic attack presented by Minier and Gilbert in FSE 2000 can work on 8-round Crypton, however, Crypton v1.0 can resist this attack very well. In 2010, Mala et al. described two new impossible differential attacks[10] on 7-round Crypton by using a 4-round impossible differential.

In this paper, we study related-key impossible differential attacks on both Crypton and Crypton v1.0 which are distinguished by key schedules and S-boxes. The attacks exploit the effect of the difference of a pair of plaintexts under two related keys with a certain key differential. Due to the special structure of the key schedules, we can exploit some 6-round related-key impossible differentials of Crypton with 256 bit user keys. By constructing 6-round related-key impossible differentials from the inner of the first round and starting the attacks from the very beginning, using some observations of diffusion layer to accelerate fil-

tration of pairs, we mount the attack on 9-round Crypton and Crypton v1.0 with 256 bit user key. The first proposed attack requires $2^{124.5}$ chosen plaintexts and $2^{176.3}$ 9-round Crypton encryptions. The second proposed attack is a data-time trade off of the first one. It requires 2^{105} chosen plaintexts and $2^{243.8}$ 9-round Crypton encryptions. Both the attacks can retrieve the whole of the 9th round subkey.

The paper is organized as follows: Section 2 briefly introduces some notations and the description of Cryptons. In section 3, we describe some 6-round related-key impossible differentials. Then attacks on Crypton and Crypton v1.0 are discussed in section 4. Section 5 gives a second attack scenario. Section 6 concludes the paper and summarizes our results.

2 Background

2.1 Outline of Crypton

Crypton is a 128-bit block cipher supports key sizes up to 256 bits. The standard number of rounds is 12. Each round employs a SPN (Substitution-Permutation Network) structure and processes 16 bytes block. Let us represent the 128-bit block A as a 4×4 matrix of bytes, left 4×4 matrix is double index and the right one is single index.

$$A = \begin{pmatrix} a_{0,3} & a_{0,2} & a_{0,1} & a_{0,0} \\ a_{1,3} & a_{1,2} & a_{1,1} & a_{1,0} \\ a_{2,3} & a_{2,2} & a_{2,1} & a_{2,0} \\ a_{3,3} & a_{3,2} & a_{3,1} & a_{3,0} \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 & 0 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 15 & 14 & 13 & 12 \end{pmatrix}$$

Crypton uses 6 elementary transformations.

- γ_o and γ_e are byte-wise non-linear substitutions which are applied to odd rounds and even rounds, respectively.
- π_o and π_e are linear transformations that act on odd rounds and even rounds, respectively. The two bit permutations mix each byte column of 4×4 byte array using four masking bytes m_i's given by

  $$m_0 = 0xfc, m_1 = 0xf3, m_2 = 0xcf, m_3 = 0x3f.$$

  We denote "."and "$\oplus$"bit-wise logical operations for AND and XOR, respectively. π_o is given as follows:

  $$B_{i,j} = \oplus_{k=0}^{3}(A_{k,j} \cdot m_{(i+j+k) \mod 4}),$$

  π_e is given as show below.

  $$B_{i,j} = \oplus_{k=0}^{3}(A_{k,j} \cdot m_{(i+j+k+2) \mod 4}).$$

  Both the branch number of π_o and π_e are 4. Note that $\pi_o^{-1} = \pi_o$, $\pi_e^{-1} = \pi_e$.
- τ is a byte transposition, it simply moves the byte at (i, j) position to (j, i) position, i.e.,$B = \tau(A) \Leftrightarrow b_{i,j} = a_{j,i}$.
- σ_K is a bit-wise key XOR with key K.

Let K_i be the i-th encryption round key derived from a user key K using the key schedule. The block cipher Crypton can be described as $\phi_e \circ \rho_e K_{12} \circ \rho_o K_{11} \circ \cdots \circ \rho_e K_2 \circ \rho_o K_1 \circ \sigma_{K_0}$, where odd round function $\rho_o K$ and even round function $\rho_e K$ are defined by $\rho_o K = \sigma_K \circ \tau \circ \pi_o \circ \gamma_o$ and $\rho_e K = \sigma_K \circ \tau \circ \pi_e \circ \gamma_e$. Linear transformation $\phi_e = \tau \circ \pi_e \circ \tau$ is used after the last round. In the same way, define $\phi_o = \tau \circ \pi_o \circ \tau$.

Modified Crypton (i.e. Crypton v1.0) features two changes which we state as follows.

1. The nonlinear transformations γ_o and γ_e use two S-boxes instead of only one. This doesn't influence our attack since we only use the fact that a S-box is a bijective map.

2. The key schedule is changed. The generation of the round keys is more complex then Crypton. This influences our attack since attacks in this paper have a close relation with the key schedule.

We outline the key schedules of both Crypton and Crypton v1.0 in Appendix A. More detail we refer [1] and [2]. In some cases, we don't distinguish Crypton and Crypton v1.0 when describing common features.

2.2 Notations

In the rest of this paper, we will use the following notations:

Let P denotes plaintext and C denotes ciphertext. x_i^{γ}, x_i^{π}, x_i^{τ} and x_i^{σ} denote the intermediate values after the application of $\gamma(\gamma_o$ or $\gamma_e)$, $\pi(\pi_o$ or $\pi_e)$, τ and σ operations of round i, respectively. K_i denotes the subkey of round i, and the initial whitening subkey is K_0.

In some cases, for reducing the attack complexity, the order of the operations in the same round is changed. We can rewrite the round function $\sigma_K \circ \tau \circ \pi_s \circ \gamma_s$ by $\tau \circ \pi_s \circ \sigma_{K^{eq}} \gamma_s$, $s \in \{o, e\}$, which is done by replacing the subkey K with an equivalent subkey K^{eq}, where $K^{eq} = \pi^{-1} \circ \tau^{-1}(K)$.

We denote the lth column of x_i by $x_{i,col(l)}$, denote columns m and n of x_i by $x_{i,col(m,n)}$. In the same way, we can denote the row(s) of x_i. For example, $x_{i,row(1)}$ includes bytes 4, 5, 6 and 7, $x_{i,col(1)}$ includes bytes 1, 5, 9 and 13.

For a 4-byte word $a = (a_3, a_2, a_1, a_0)$, we call a_0 the least byte of a and a_3 the most byte of a. $a^{\ll n}$ denotes left rotation of a by n bits positions, and $a^{\ll b^n}$ denotes left rotation of each byte in a 32-bit word a by n bits positions.

Table 1. n_{ij} **values[10]**

i	j				
	0	1	2	3	4
0	1	0	0	0	0
1	0	0	0	48	972
2	0	0	108	5760	384,282
3	0	48	5760	1,024,800	65,294,892
4	0	972	384,283	65,294,892	4,162,570,479

2.3 Two observations on Crypton

In [10], two observations on diffusion layer of Crypton are given. We list them in following since they are also important to our attacks.

Observation 1. Let n_{ij} be the number of 4-byte words with i non-zero bytes that after the application of π are converted to 4-byte words with j non-zero bytes. Values n_{ij} obtained by computer experiment are given in Table 1. The probability that π transformation transfers a 4-byte word with i non-zero bytes in fixed positions into a word with j non-zero bytes in fixed positions is equal to $p_{ij} = \frac{n_{ij}/C_4^i C_4^j}{(2^8-1)^i}$.

Observation 2. The linear transformation $\pi_e \circ \pi_o$ is equivalent to a byte permutation. Let $C = \pi_e \circ \pi_o$, then $C_{row(i)} = A_{row(i+2) \mod 4}$.

From observation 2, we can easily deduce that $\pi_e \circ \pi_o$ is equal to $\pi_o \circ \pi_e$ since $\pi_e \circ \pi_o = \pi_e^{-1} \circ \pi_o^{-1} = (\pi_o \circ \pi_e)^{-1}$ and $(\pi_o \circ \pi_e)^{-1} = \pi_o \circ \pi_e$. Therefore, when the equivalent round is used in the last round of Crypton reduced to r rounds, the intermediate value x_r^σ is a byte permutation of the ciphertext.

3 6-round related-key impossible differentials of Crypton

In this section, we introduce some 6-round related-key impossible differentials of Crypton-256.

In the key schedule of Crypton, the 256-bit keys are splitted into two 128-bit words, then round transformations are applied to the two words. Since the round transformations are permutations of the 128-bit words, one can easily obtain the inputs given the outputs. Therefore, we can trace the key relations after the round transformations if 256-bit user key is used. However, when user key with other length is used, some zeros are padded to make K to 256 bits, which will confine the relations of the keys. This property makes Crypton-256 more susceptible to related-key attacks than Crypton with other length key. In this paper, we only study the security of 256-bit key version of Crypton against the related-key impossible differential cryptanalysis.

3.1 6-round impossible differentials of Crypton v1.0-256

For Crypton v1.0, we choose two related keys with difference of U and V as follows:

$$\Delta U' = \begin{pmatrix} a & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ a & 0 & 0 & 0 \end{pmatrix}, \Delta V' = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Table 2. Round key differences of Crypton v1.0

Round	$\Delta k_{i,row(0)}$	$\Delta k_{i,row(1)}$	$\Delta k_{i,row(2)}$	$\Delta k_{i,row(3)}$
0	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
1	(a,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
2	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
3	(0,0,0,0)	(b,0,0,0)	(0,0,0,0)	(0,0,0,0)
4	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
5	(0,0,0,0)	(0,0,0,0)	(0,0,0,b)	(0,0,0,0)
6	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
7	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,b,0,0)
8	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
9	(0,c,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)

a,b and c are nonzero values, and $b = a^{\ll 2}$, $c = a^{\ll 4}$.

Hence, the round key differences for the first 9 rounds are presented in Table 2, which will be used in our attacks later.

Using the above key relation, a 6-round related-key impossible differential can be built as Fig. 1. Firstly, a 4-round related-key impossible differential differential can be built with probability 1 in the forward direction, then a 2-round related-key differential with probability 1 in the reverse direction, where the intermediate differences contradict each other. We still use the notations introduced by [10], i.e. the boxes with a black circle refer to bytes with non-zero difference and white boxes with "?"refer to bytes with unknown difference and white boxes refer to bytes with zero difference. Notice that after applying transformation π, the position of each "?"is not fixed.

3.2 6-round impossible differentials of Crypton -256

For Crypton, we choose two related keys with the following difference of V_i.

$$(\Delta V_e[3], \ldots, \Delta V_e[0])^T = \begin{pmatrix} a & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ a & 0 & 0 & 0 \end{pmatrix},$$

$$(\Delta V_e[7], \ldots, \Delta V_e[4])^T = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The round key differences of Crypton-256 for the first 9 rounds are presented in Table 3.

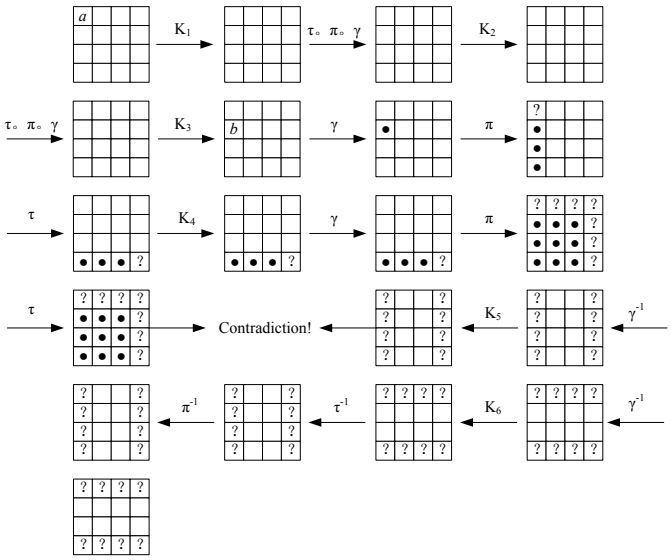


Figure 1. 6-round related-key impossible differential of Crypton v1.0

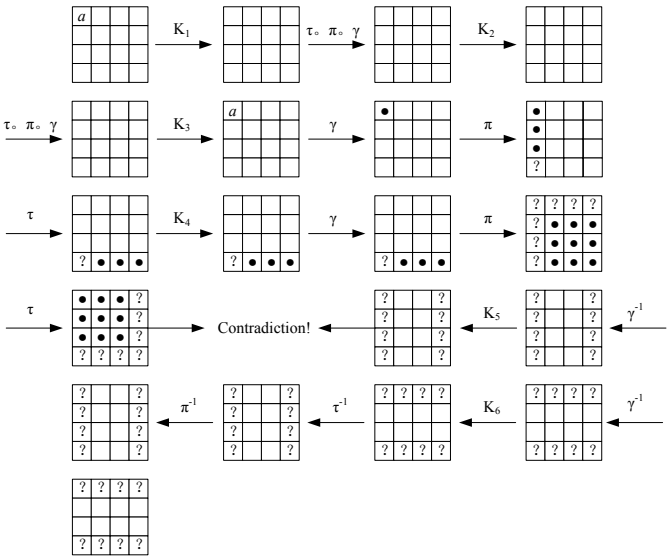


Figure 2. 6-round related-key impossible differential of Crypton

Table 3. Round key differences of Crypton

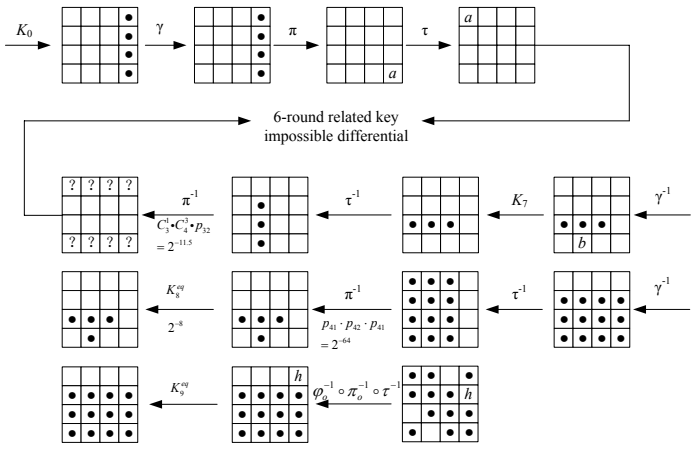
Round	$\Delta k_{i,row(0)}$	$\Delta k_{i,row(1)}$	$\Delta k_{i,row(2)}$	$\Delta k_{i,row(3)}$
0	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
1	(a,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
2	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
3	(a,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
4	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
5	(0,0,0,a)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
6	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
7	(0,0,0,a)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
8	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
9	(0,a,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)

a is a nonzero value.

In the same way, we can obtain the related-key impossible differentials one of which is shown in Fig. 2.

4 A 9-round related-key impossible differential attack

In this Section, we describe attacks of two versions of Crypton reduced to 9 rounds. The attacks are based on the above 6-round related-key impossible differentials with additional one round at the beginning and two rounds at the end. In the 8th round and 9th round, we use their equivalent round functions. The attacks on Crypton v1.0 and Crypton are depicted in Fig.3 and Fig.4 respectively. We only present the attack procedure of Crypton v1.0. The attack on Crypton is quite similar.



a and b are nonzero values, $b = a^{\ll 2}$, h is the most byte of $\pi_o(0,0,a^{\ll 4},0)^T$

Figure 3. 9-round related-key impossible differential attack on Crypton v1.0

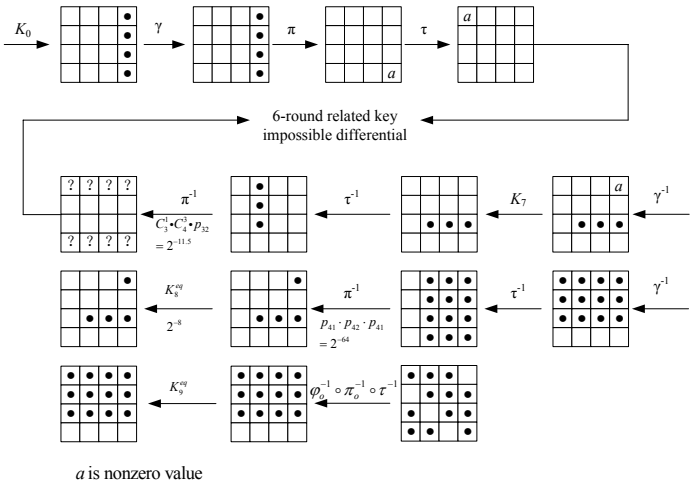


Figure 4. 9-round related-key impossible differential attack on Crypton

4.1 Attack procedure

Precomputation: For all the 2^{32} possible pairs of values of $(x^{\tau}_{1,col(3)}, x'^{\tau}_{1,col(3)})$ with difference $(a,0,0,0)$, compute 4 bytes values in byte 0,4,8,12 of plaintexts. Store the pairs of 4-byte values in a hash table S indexed by the XOR difference in these bytes.

The attack algorithm is as follows:

1. Take 2^n structures of plaintexts such that in each structure, plaintexts have fixed values in all bytes but column 0, thus we get 2^{n+32} plaintexts and 2^{n+63} plaintext pairs. Choose plaintext pairs $(P, P^{'})$ whose corresponding ciphertext pairs $(C, C^{'})$ have zero difference at the three bytes (1,11,14) and have difference of h in byte 4, where h is the first byte of $\pi_o(0,0,a^{\lll 4},0)^T$. From observation 2, we can obtain $(x^{\sigma}_9, x'^{\sigma}_9)$ from (C, C'). The expected number of such pairs is $2^{n+63} \times 2^{-8\times 4} = 2^{n+31}$.

2. For all pairs $(x^{\sigma}_9, x'^{\sigma}_9)$, compute $x^{*\sigma}_9 = x'^{\sigma}_9 \oplus \pi_o(0,0,a^{\lll 4},0)^T$ to obtain new pairs $(x^{\sigma}_9, x^{*\sigma}_9)$.

3. Guess the 32-bit value for $K^{eq}_{9,row(1)}$, and for each guess, partially decrypt the pairs $(x^{\sigma}_{9,row(1)}, x^{*\sigma}_{9,row(1)})$ to obtain $(x^{\sigma}_{8,col(1)}, x'^{\sigma}_{8,col(1)})$. Choose pairs whose difference $\Delta x^{\sigma}_{8,col(1)}$ is nonzero at byte 9 and zero at bytes 1,5,13. From Observation 1, the probability of such a difference is equal to $p_{41} \simeq 2^{-24}$, thus the expected number of the remaining pairs is $2^{n+31} \times p_{41} = 2^{n+7}$.

4. Guess the 32-bit value for $K^{eq}_{9,row(2)}$, and for each guess, partially decrypt the remaining pairs through one round to obtain $(x^{\sigma}_{8,col(2)}, x'^{\sigma}_{8,col(2)})$. Choose pairs whose difference $\Delta x^{\sigma}_{8,col(2)}$ is nonzero at bytes 10, 14 and zero at bytes 2, 6. The probability of such a difference is equal to $p_{42} \simeq 2^{-16}$, thus the expected number of the remaining pairs is $2^{n+7} \times p_{42} = 2^{n-9}$.

5. Guess the 32-bit value for $K^{eq}_{9,row(3)}$, and for each guess, partially decrypt the remaining pairs to obtain $(x^{\sigma}_{8,col(3)}, x'^{\sigma}_{8,col(3)})$. Choose pairs whose difference $\Delta x^{\sigma}_{8,col(3)}$ is nonzero at byte 11 and zero at bytes 3,7,15. The probability of such a difference is also equal to $p_{41} \simeq 2^{-24}$, thus the expected number of the remaining pairs is $2^{n-9} \times p_{41} = 2^{n-33}$.

6. Guess the 8-bit value for K^{eq}_8 in byte 14, and for each guess, partially decrypt the remaining pairs to obtain $(x^{\sigma}_7, x'^{\sigma}_7)$ in byte 14. Choose pairs whose difference Δx^{σ}_7 in byte 14 is b, where $b = a^{\lll 2}$. The probability of such a difference is equal to 2^{-8}, thus the expected number of the remaining pairs is $2^{n-33} \times 2^{-8} = 2^{n-41}$.

7. Guess the 24-bit value for K^{eq}_8 in bytes 9,10,11, and for each guess, partially decrypt the remaining pairs to obtain $\Delta x^{\gamma}_{6,col(2)}$. Choose pairs whose difference $\Delta x^{\gamma}_{6,col(2)}$ has two active bytes, one of them is in byte 2, and the location of the other difference is not important. According to Observation 1, the probability of such a difference is $C^1_3 \cdot p_{23}$. Besides, the location of three nonzero differences in $\Delta x^{\pi}_{6,col(2)}$ is also arbitrary (if choose different 3 nonzero positions, the other steps must be changed accordingly), therefore the probability is equal to $C^1_3 \cdot C^3_4 \cdot p_{23} \simeq 2^{-11.5}$. Thus the expected number of the remaining pairs is $2^{n-41} \times 2^{-11.5} = 2^{n-52.5}$.

8. Initialize a list A of the 2^{32} possible values of the bytes $K_{0,col(0)}$.

9. For each of the $2^{n-52.5}$ remaining plaintext pairs, compute $\Delta P = P \oplus P^{'}$. If the bin ΔP in S is nonempty, access this bin. For each pair (x, y) in this bin, remove from the list A the value $P_{col(0)} \oplus x$. The probability that a subkey $P_{col(0)} \oplus x$ be removed by a remaining pair is about 2^{-32}. We expect each pair deletes one subkey candidate on average.

10. If A is not empty, output the values in A along with the guess of $K^{eq}_{9,row(1,2,3)}$ and $K^{eq}_{8,(9,10,11,14)}$.

4.2 Analysis of the attack complexity

In this attack, 16 bytes subkey $K^{eq}_{9,row(1,2,3)}$ and $K^{eq}_{8,(9,10,11,14)}$ should be guessed. After the filtering in step 7, there remains about $2^{n-52.5}$ plaintext pairs. After analyzing one of such pairs, the probability that a wrong 16-byte key value survives the elimination process is $1-2^{-32}$. Thus after analyzing all the $2^{n-52.5}$ pairs, only about $2^{8\times 16}(1-2^{-32})^{2^{n-52.5}}$ wrong key guess remain. If $n = 92.5$, the expected number is much smaller than 1, and we can expect that only the right subkey will remain. Therefore, the number of required plaintexts is $2^{n+32} = 2^{124.5}$.

The time complexity can be computed as follows. Step 3 requires about $2\times 2^{32}\times 2^{n+31} = 2^{156.5}$ computations which equivalent to $2^{156.5} \times \frac{4}{16} = 2^{154.5}$ one round encryptions.

Step 4 requires about $2 \times 2^{32+32} \times 2^{n+7} \times \frac{4}{16} = 2^{162.5}$ one round encryptions. Step 5 requires about $2 \times 2^{32+32+32} \times 2^{n-9} \times \frac{4}{16} = 2^{178.5}$ one round encryptions. Step 6 requires about $2 \times 2^{32+32+32+8} \times 2^{n-33} \times \frac{1}{16} = 2^{160.5}$ one round encryptions. Step 7 requires about $2 \times 2^{32+32+32+8+24} \times 2^{n-41} \times \frac{4}{16} = 2^{177.5}$. At last, step 9 requires about $2 \times 2^{128} \times 2^{n-52.5} = 2^{170}$ memory accesses.

For recovering the other four bytes of K_9^{eq}, a scenario similar to the above attack can be performed. As described in step 7, we expect 3 nonzero positions in $\Delta x^{\pi}_{7,col(2)}$ is changed, and we change the other steps accordingly. This attack complexity is the same as the above one. Hence the time complexity is about $2^{178.5} \times 2 = 2^{179.5}$ for obtaining the whole K_9^{eq}.

Consequently, this attack requires about $2^{124.5}$ chosen plaintexts and less than $2^{179.5} \times \frac{1}{9} \simeq 2^{176.3}$ encryptions of 9-round Crypton v1.0.

Both the procedure and the complexity of the attack on Crypton-256 are just the same with the above one. So we omit the details here.

5 A second attack scenario

In the following, we will give another 9-round attack on Crypton v1.0. This attack is a data-time trade off of the first one. We present the attack in Fig. 5.

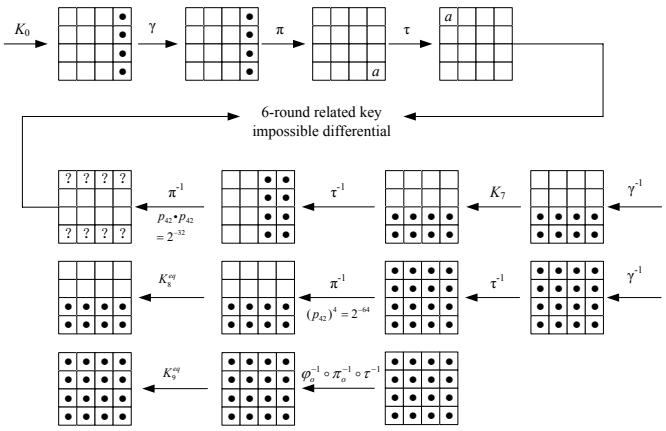


Figure 5. a second 9-round related-key impossible differential attack on Crypton v1.0

In this attack, the precomputation is the same as the first attack. The difference lies in the way of tracing the propagation of ciphertexts. We can use all the 2^{n+63} plaintext pairs in this attack. All the 16 bytes of K_9^{eq} and a half of K_8^{eq} in bytes 8, 9, 10, 11, 12, 13, 14, 15 need to be guessed.

We first guess each row of K_9^{eq} and partially decrypt each row of pairs $(x_9^{\sigma}, x_9^{*\sigma})$, then check whether only the difference in row 2 and row 3 are nonzero. This filtering is done by multiplying four $p_{42} \simeq 2^{-16}$ conditions. Guess the 4 bytes of K_8^{eq} in bytes 8,9,10,11, and partially decrypt the remaining pairs to get $\Delta x^{\gamma}_{6,col(1)}$. Choose pairs whose difference $\Delta x^{\gamma}_{6,col(1)}$ has two active bytes (one with fixed position and the other is flexible), this probability is $3 \times p_{42}$. Guess another four bytes of K_8^{eq} in bytes 12,13,14,15 to calculate $\Delta x^{\sigma}_{7,row(3)}$, then reverse $\Delta x^{\sigma}_{7,row(3)} \oplus \{0, b, 0, 0\}$ to get $\Delta x^{\sigma}_{6,col(0)}$, where $b = a^{\ll 2}$, in the same way, choose pairs whose difference $\Delta x^{\gamma}_{6,col(1)}$ has two active bytes. This probability is p_{42}. After this filtering, the remaining ciphertext pairs can be used to discard wrong subkey guesses.

In this attack, we guess a total of 192 subkey bits, but only a portion of 2^{-96} of the pairs can be used to discard wrong subkey guesses. By choosing 2^n structure, we can get 2^{n+32} plaintexts and 2^{n+63} plaintext pairs. After analyzing all the 2^{n-33} pairs, only about $2^{200}(1 - 2^{-32})^{2^{n+63-96}}$ wrong key guess remain. If $n = 73$, the expected number is much smaller than 1, and we can expect that only the right subkey will remain. Therefore, the number of required plaintexts is $2^{n+32} = 2^{105}$. The time complexity is dominated by the step of guessing K_8^{eq} in bytes 12,13,14,15, in this step, 2^{192} bits subkey should be guessed, the number of remaining pairs is $2^{n+63-80}$. Therefore, the time complexity is $2 \times 2^{192+56} \times \frac{4}{16} = 2^{247}$ one round encryptions which is equivalent to $2^{247} \times \frac{1}{9} = 2^{243.8}$ encryptions of 9-round Crypton. For obtaining the whole K_9^{eq}, the time complexity should be doubled.

6 Summary

This paper considers impossible differential cryptanalysis under related key model firstly. Different from most of the previous results on Crypton with 128-bit user keys, our targets are Crypton and Crypton v1.0 with 256-bit user keys. By choosing proper differences of the related keys, we constructed 6-round related-key impossible differentials of Crypton and Crypton v1.0, and proposed 9-round attacks on the two versions of Crypton-256.

In these attacks, several techniques, including appropriate selection of additional rounds and using hash table, made the attack effective. Besides, the chosen related-key difference made our attack start from the very beginning of the cipher, the property of the diffusion layer made us obtain intermediate values from ciphertexts directly. These attacks retrieve the whole of the 9th round subkey of two versions of Crypton-256.

This work is the extend version of "Related-key impossible differential cryptanalysis on Crypton and Crypton v1.0"which appears in WorldCIS 2011, and it is supported by the Natural Science Foundation of China (No: 60803156, 61070215).

References

[1] C.H. Lim. Crypton: A New 128-bit Block Cipher. The First Advanced Encryption Standard Candidate Conference, NIST, 1998.

[2] C.H. Lim. A Revised Version of Crypton-Crypton v1.0. FSE'99, LNCS 1636, pp. 31-45, 1999.

[3] L. Knudsen. DEAL — A 128-bit Block Cipher. Technical Report 151, Department of Informatics, University of Bergen, Bergen, Norway, Feb. 1998.

[4] E. Biham, A. Biryukov, A. Shamir. Cryptanalysis of Skipjack Reduced to 31 rounds Using Impossible Differentials. EuroCrypt'1999, LNCS 1592, pp. 12–23. Springer–Verlag, 1999.

[5] E.L. Biham. New Types of Cryptanalytic Attacks Using Related Keys. Journal of Cryptology. 4(1), pp. 3-72, 1991.

[6] G. D'Halluin, G. Bijnens, V. Rijmen and B. Preneel. Attack on Six Rounds of Crypton. FSE'99, LNCS 1636, pp. 46-59, Springer–Verlag, 1999.

[7] H. Seki, T. Kaneko. Cryptanalysis of Five Rounds of Crypton Using Impossible Differentials. Advances in Cryptology-Asiacrypt'99. LNCS 1716, pp. 45-51, Springer–Verlag, 1999.

[8] J. Cheon, M. Kim, K. Kim, J. Lee. Improved Impossible Differential Differential Cryptanalysis of Rijndael and Crypton. ICISC 2001, LNCS 2288, pp. 39-49, Springer–Verlag, 2002.

[9] M. Minier, H. Gilbert. Stochastic Cryptanalysis of Crypton. FSE 2000, LNCS 1978, pp. 121-133. Springer–verlag, 2001.

[10] H. Mala, M. Shakiba, M. Dakhilalian. New Impossible Differential Attacks on Reduced-round Crypton. Computer Standards and Interfaces, 32(2010), pp. 222–227, 2010.

A Appendix

A.1 Key Schedule of CRYPTON

The key schedule expands the user key K into 13 32-bit round keys. Firstly prepend as many zeros to K as need to make K to 256 bits. Then convert the resulting user key into 8 32-bit words $U[i](0 \leq i \leq 7)$: $U[i] = k_{4i+3}k_{4i+2}k_{4i+1}k_{4i}$, and perform the following:

$$(V_e[3], V_e[2], V_e[1], V_e[0])^T = (\tau \circ \gamma_o \circ \sigma_P \circ \pi_o)((U[6], U[4], U[2], U[0])^T)$$
$$(V_e[7], V_e[6], V_e[5], V_e[4])^T = (\tau \circ \gamma_o \circ \sigma_P \circ \pi_o)((U[7], U[5], U[3], U[1])^T)$$
$$T_0 = V_e[0] \oplus V_e[1] \oplus V_e[2] \oplus V_e[3]$$
$$T_1 = V_e[4] \oplus V_e[5] \oplus V_e[6] \oplus V_e[7]$$
$$E_e[i] = V_e[i] \oplus T_1 \; for \; i = 0, 1, 2, 3$$
$$E_e[i] = V_e[i] \oplus T_0 \; for \; i = 4, 5, 6, 7$$

Table 4. Round keys of the first 9 rounds of Crypton

$K_e[0] = E_e[0]$	$K_e[4] = E_e[4]$
$K_e[1] = E_e[1]$	$K_e[5] = E_e[5]$
$K_e[2] = E_e[2]$	$K_e[6] = E_e[6]$
$K_e[3] = E_e[3]$	$K_e[7] = E_e[7]$
$K_e[8] = ROL(E_e[0], 8)$	$K_e[12] = E_e[4] \oplus RC_0$
$K_e[9] = E_e[1] \oplus RC_0$	$K_e[13] = ROL(E_e[5], 16)$
$K_e[10] = ROL(E_e[2], 16)$	$K_e[14] = E_e[6] \oplus RC_0$
$K_e[11] = E_e[3] \oplus RC_0$	$K_e[15] = ROL(E_e[7], 24)$
$K_e[16] = ROL(E_e[0], 8) \oplus RC_1$	$K_e[20] = ROL(E_e[4], 8) \oplus RC_0$
$K_e[17] = ROL(E_e[1], 24) \oplus RC_0$	$K_e[21] = ROL(E_e[5], 16) \oplus RC_1$
$K_e[18] = ROL(E_e[2], 16) \oplus RC_1$	$K_e[22] = ROL(E_e[6], 16) \oplus RC_0$
$K_e[19] = ROL(E_e[3], 8) \oplus RC_0$	$K_e[23] = ROL(E_e[7], 24) \oplus RC_1$
$K_e[24] = ROL(E_e[0], 24) \oplus RC_1$	$K_e[28] = ROL(E_e[4], 8) \oplus RC_{02}$
$K_e[25] = ROL(E_e[1], 8) \oplus RC_{02}$	$K_e[29] = ROL(E_e[5], 8) \oplus RC_1$
$K_e[26] = ROL(E_e[2], 8) \oplus RC_1$	$K_e[30] = ROL(E_e[6], 16) \oplus RC_{02}$
$K_e[27] = ROL(E_e[3], 8) \oplus RC_{02}$	$K_e[31] = E_e[7] \oplus RC_1$
$K_e[32] = ROL(E_e[0], 24) \oplus RC_{13}$	$K_e[36] = ROL(E_e[4], 24) \oplus RC_{02}$
$K_e[33] = E_e[1] \oplus RC_{02}$	$K_e[37] = ROL(E_e[5], 8) \oplus RC_{13}$
$K_e[34] = ROL(E_e[2], 8) \oplus RC_{13}$	$K_e[38] = ROL(E_e[6], 8) \oplus RC_{02}$
$K_e[35] = ROL(E_e[3], 24) \oplus RC_{02}$	$K_e[39] = E_e[7] \oplus RC_{13}$

The first 9 round keys with initial key are given in table 4, where P, Q, RC_0, RC_1, RC_{02} and RC_{13} are constants we don't care about.

A.2 Key Schedule of CRYPTON v1.0

256 bit user key $K = k_{31} \ldots k_1 k_0$ are splited into U and V such that $U[i] = k_{8i+6}k_{8i+4}k_{8i+2}k_{8i}$ and $V[i] = k_{8i+7}k_{8i+5}k_{8i+3}k_{8i+1}$ for $i = 0, 1, 2, 3$. Then compute $E_e[i]$ using round transformations with all-zero key as

$$U' = \rho_o(U), \qquad V' = \rho_e(V)$$
$$E_e[i] = U'[i] \oplus T_1, \;\; E_e[i+4] = V'[i] \oplus T_0,$$

where $T_0 = \oplus_{i=0}^{3} U'[i]$ and $T_1 = \oplus_{i=0}^{3} V'[i]$.
1. compute the round keys for the first 2 rounds as

$$K_e[i] \leftarrow E_e[i] \oplus C_e[0] \oplus MC_i,$$
$$K_e[i+4] \leftarrow E_e[i+4] \oplus C_e[1] \oplus MC_i, for \; 0 \leq i \leq 3.$$

2. for rounds r=2,3,...,12, repeat the following two steps alternately:
For even rounds:

$$\{E_e[3], E_e[2], E_e[1], E_e[0]\} \leftarrow \{E_e[0]^{\ll b^6}, E_e[3]^{\ll b^6}, E_e[2]^{\ll 16}, E_e[1]^{\ll 24}\},$$
$$K_e[4r+i] \leftarrow E_e[i] \oplus C_e[r] \oplus MC_i, for \; 0 \leq i \leq 3.$$

For odd rounds:

$$\{E_e[7], E_e[6], E_e[5], E_e[4]\} \leftarrow \{E_e[6]^{\ll 16}, E_e[5]^{\ll 8}, E_e[4]^{\ll b^2}, E_e[7]^{\ll b^2}\},$$
$$K_e[4r+i] \leftarrow E_e[i+4] \oplus C_e[r] \oplus MC_i, for\ 0 \leq i \leq 3.$$

$C_e[k]$ and MC_i are constants we don't care about.