JAIST Repository
https://dspace.jaist.ac.jp/

Title	Generalized Analysis on Key Collisions of Stream Cipher RC4
Author(s)	Chen, Jiageng; Atsuko Miyaji
Citation	IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E94-A(11): 2194-2206
Issue Date	2011-11-01
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/10522
Rights	Copyright (C)2011 IEICE. Jiageng Chen and Atsuko Miyaji, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E94-A(11), 2011, 2194-2206. http://www.ieice.org/jpn/trans_online/
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

PAPER Special Section on Information Theory and Its Applications

Generalized Analysis on Key Collisions of Stream Cipher RC4**

Jiageng CHEN[†a)], *Nonmember and* Atsuko MIYAJI[†*], *Member*

SUMMARY The fact that the stream cipher RC4 can generate colliding key pairs with hamming distance one was first discovered by Matsui in FSE 2010. This kind of weakness demonstrates that two different secret keys have the same effect on the cipher's encryption and the corresponding decryption procedure. In this paper, we further investigate the property of RC4 key collisions and achieved the following results:
1. We show that RC4 can generate colliding key pairs with various hamming distances, which cannot be generated by Matsui's pattern. We also give concrete examples of colliding key pairs with hamming distances greater than one.
2. We formalize RC4 colliding key pairs into two large patterns, namely, Transitional pattern and Self-Absorbing pattern. All the currently known colliding key pairs can be categorized into either two patterns.
3. We analyze both patterns and clarified the relations among the probability of key collision, key length and hamming distances which yield the colliding key pairs.
4. We demonstrate the vulnerability of key collisions by showing collisions of RC4-Hash function proposed in INDOCRYPT 2006. Some concrete experimental results of RC4-Hash collision are also given in this paper.
key words: *RC4, key collisions, KSA, hamming distance, RC4-Hash*

1. Introduction

The stream cipher RC4 is one of the most famous ciphers widely used in real world applications such as Microsoft Office, Secure Socket Layer (SSL), Wired Equivalent Privacy (WEP), etc. Due to its popularity and simplicity, RC4 has become a hot cryptanalysis target since its specification was made public on the Internet in 1994 [4]. Various general weaknesses of RC4 have been discovered in some previous works including [5]–[7], etc. Another popular cryptanalysis direction of RC4 is in the WEP environment. Such works include [8]–[11], etc.

This paper specifically investigates the weakness of RC4 key collisions, namely, the existence of secret key pairs that generate the same initial states after the key scheduling algorithm. Needless to say, any secure cipher designs should not have such properties since two different keys will have the same effect on the encryption and the corresponding decryption, and also the key space is reduced which can help improve the brute force attack. The study of "colliding keys" of RC4 can be dated back to 2000. Grosul and Wallach [1] first pointed out that RC4 can generate near collisions when the key size is close to the full 256 bytes. In [2] first colliding key pairs with hamming distance one were discovered, where hamming distance one means that the two keys differ from each other at one position.

We further investigate this area, and pointed out that RC4 can generate far more colliding key pairs with different properties other than the ones given in [2]. We clarify that all the currently known RC4 colliding key pairs can be organized into two patterns, according to the behavior during the KSA. We analyze these two generalized patterns and formalize the RC4 key collisions. Collision probability is also estimated, and we point out that it is mainly affected by key length and hamming distances between the two keys. We prove this fact in the theorems, and also some concrete results are shown.

The collision patterns demonstrate that the direct use of KSA as a compression function to build the hash function is not secure by showing concrete examples of RC4-Hash [14].

Structure of the paper. In Sect. 2, we briefly describe the RC4 algorithm, followed by some previous works on RC4 key collisions. Sect. 3 shows the formalized RC4 colliding key patterns and how they work. The probability evaluations and the simulations are given in Sect. 4 followed by the vulnerability of key collisions in Sect. 5 with experimental collision results on RC4-Hash. Also concrete RC4 colliding key pairs and a fast searching technique for searching collisions in Self-Absorbing pattern are given in Appendix B and Appendix D.

2. Preparation

2.1 Description of RC4

The internal state of RC4 consists of a permutation S of the numbers $0, \cdots, N-1$ and two indices $i, j \in \{0, \cdots, N-1\}$. The index i is determined and known to the public, while j and permutation S remain secret. RC4 consists of two algorithms: The Key Scheduling Algorithm (KSA) and the Pseudo Random Generator Algorithm (PRGA). The KSA generates an initial state from a random key K of k bytes as described in Algorithm 1. It starts with an array $\{0, 1, \cdots, N-1\}$ where $N = 256$ by default. At the end, we obtain the initial state S_{N-1}. Once the initial state is created, it is used by PRGA. The purpose of PRGA is to generate a keystream of bytes which will be XORed with the

Manuscript received February 18, 2011.
Manuscript revised June 10, 2011.
[†]The authors are with Japan Advanced Institute of Science and Technology, Nomi-shi, 923-1292 Japan.
[*]This study is partly supported by Grant-in-Aid for Scientific Research (B), 20300032.
[**]The preliminary versions were presented at ISPEC 2010 and SCN 2010, respectively.
a) E-mail: jg-chen@jaist.ac.jp
DOI: 10.1587/transfun.E94.A.2194

Copyright © 2011 The Institute of Electronics, Information and Communication Engineers

plaintext to generate the ciphertext. PRGA is described in Algorithm 2.

Algorithm 1. KSA

```
for i = 0 to N − 1 do
    S[i] ← i
end for
j ← 0
for i = 0 to N − 1 do
    j ← j + S[i] + K[i mod k]
    swap(S[i], S[j])
end for
```

Algorithm 2. PRGA

```
i ← 0
j ← 0
loop
    i ← i + 1
    j ← j + S[i]
    swap(S[i], S[j])
keystream byte z_i = S[S[i] + S[j]]
end loop
```

In this paper, we mainly focus on KSA algorithm.

2.2 Previous Research on RC4 Key Collisions

Two important previous studies on RC4 key collisions are [1], [2]. In [1], the authors pointed out that it's possible for two secret keys with length close to 256 bytes to generate similar internal state after KSA, and thus they will generate similar hundred byte output during PRGA. The reason for this is that for two keys K_1, K_2, if we assume $K_1[i] = K_2[i]$ except when $i = t$, then when t is close to 255, the two internal states will be substantially similar. However, this idea cannot generate strict key collisions, and this result only works for key lengths close to 256.

In [2], RC4 key collision was first discovered. The key pattern is almost the same as in [1], namely, two keys differ at only one byte position ($K_1[i] = K_2[i]$ except $i = t$) and the value difference is $1(K_1[t] = K_2[t]-1)$. The intuition behind the collision is that from the first time i touches the different position t, the pattern ensures that there are always only two differences in the internal state as the key scheduling process continues. The difference is absorbed when i touches t for the last time. Please refer to [2] for the detailed description.

3. Generalized RC4 Colliding Key Pairs

We show that RC4 can generate many other colliding key pairs with different key relations, which cannot generated by using the techniques in [2]. We formalize all the currently known colliding key pairs into two patterns. We describe them in the following section by first giving the key relations, and then explaining how the two keys with these relations can achieve collisions.

3.1 Notation

- K_1, K_2: a secret key pair with some differences between them.
- $S_{1,i}, S_{2,i}$: S-Boxes corresponding to the secret key pair at time i before the swap operation.
- $i, j_{1,i}, j_{2,i}$: internal states of RC4. When $j_{1,i} = j_{2,i}$, we use j_i to denote.
- k: the lengths (bytes) of the secret keys.
- h: hamming distances between the two keys (number of different positions where two keys differ from each other).
- d: the first index of the key differences.
- Γ: the set of indices at which two keys differ from each other, $|\Gamma| = h$, $\Gamma = \{\gamma_0, \cdots, \gamma_{h-1}\}$ and $d = \gamma_0$.
- n_i: the number of times the key difference γ_i appears during the KSA. $n_i = \lfloor \frac{256+k-1-\gamma_i}{k} \rfloor$ for $i = 0, \cdots, h-1$.

3.2 Transitional Pattern

Key relations in Transitional pattern:

$$K_2[i] = K_1[i] + 1, i \in \Gamma$$

Namely, two keys differ from each other at h places, and the value differences at these positions all equal 1.

Transitional pattern has the property that after the first internal state differences are generated, which is due to the key difference, the internal state differences are transferred to the later indices of the S-Box, and these differences exist before the last key difference comes into play during the KSA.

Figure 1 illustrates the case in which the secret keys are short, so they will appear several times during the KSA. When i first touches the key difference, j difference and two S-Box differences are generated. Notice that the Transitional pattern requires that one j equal i. Thus the two S-Box differences generated at the beginning are located next to each other, and meanwhile, we require that S-Box value differences also be one. The dotted line area in the figure shows the three internal state differences generated by the first key difference. The next two j return to the same value,

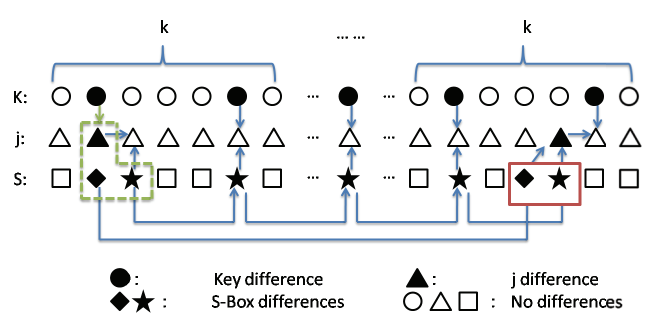


Fig. 1 Transitional pattern.

Table 1 Transitional pattern, $h = 3, n = 2 (k = 128)$.

			Internal State																		Difference
i	$K_1[i]/K_2[i]$	$j_{1,i}/j_{2,i}$	0	1	2	3	4	5	6	7	8	$\cdots$	129	130	131	132	133	134	135	136	
0	$K_1[0]$	*		1	2																Same
	$K_2[0] = K_1[0]$	*		1	2																
1	$K_1[1]$	1		1	2																j, S-Box
	$K_2[1] = K_1[1] + 1$	2		2	1																
2	$K_1[2]$	4		1			2														S-Box
	$K_2[2] = K_1[2]$	4		2			1														
4	$K_1[4]$	8		1							2										S-Box
	$K_2[4] = K_1[4] + 1$	8		2							1										
8	$K_1[8]$	129		1									2								S-BoX
	$K_2[8] = K_1[8] + 1$	129		2									1								
129	$K_1[1]$	132		1												2					S-Box
	$K_2[1] = K_1[1] + 1$	132		2												1					
132	$K_1[4]$	135		1															2		S-Box
	$K_2[4] = K_1[4] + 1$	135		2															1		
134	$K_1[6]$	1																1	2		S-Box
	$K_2[6] = K_1[6]$	1																2	1		
135	$K_1[7]$	135																1	2		j
	$K_2[7] = K_1[7]$	134																1	2		
136	$K_1[8]$	*																1	2		Same
	$K_2[8] = K_1[8] + 1$	*																1	2		

due to the effects of previous j difference (▲) and one S-Box difference (★). Meanwhile, the S-Box difference (★) is transferred to the next key difference index, and this transfer will repeat each time when i touches the next key difference index. The situation for the last appearance of the key is a little bit different. In order to achieve a collision, we require that the two S-Box differences ♦, ★ be in consecutive positions just before the last key difference index. The two S-Box differences are absorbed by each other and generate a j difference (▲). Finally, the last key difference is there to absorb the previous j difference and the internal states become the same.

The colliding key pairs found in [2] demonstrate a special case of this pattern, where the hamming distance between two keys can only be one ($|\Gamma| = h = 1$). In our generalized Transitional pattern, two keys can have various hamming distances as the probability allows. Table 1 shows an example with concrete numbers on how a 128-byte colliding key pair with hamming distance three can achieve a collision. Two keys differ from each other at indices 1, 4 and 8. The S-Box in Table 1 denotes the state after the swap operation at each step. Notice that when $i = 134$, one of the S-Box differences should be swapped to the index 134, but not necessarily from index 1, as shown in the example. The first S-Box difference can be touched by j before 134, to be swapped to other positions. As long as this S-Box difference appears in index 134 when $i = 134$, the pattern works.

3.3 Self-Absorbing Pattern

In addition to the above Transitional pattern, we investigate that some of the other RC4 colliding key pairs have the following properties: the internal state differences are generated and absorbed within one key appearance, namely, the differences will not be transferred to the later parts of the S-Box.

In [12], the authors pointed out that by setting the two keys as

$$K_2[d] = K_1[d] + 1, K_2[d+1] = K_1[d+1] - 1$$

collisions might be achieved. [12] assumes that $j_{1,d} = d$, and $j_{1,d+1} = d + 1$, then after $i = d + 1$, the two S-Box becomes the same. However, at step $i = d + 1$ after the swap, $j_{1,d+1} \neq j_{2,d+1}$, and two S-Box will differ from each other again due to the j difference, thus a collision can not be achieved as they expected. It seems that by adding another key differential $K_2[d+2] = K_1[d+2]+1$, the j difference will be absorbed and a collision can be achieved. However this is only the case for long keys whose key differential indices don't repeat. For short keys which appear more than once during KSA, collisions can not be achieved.

To explain, let's see the conditions we need to satisfy for the first and second key appearances. According to the key pattern, we need $S_{1,d}[d+1] = S_{1,d}[d] + 1 (S_{2,d}[d+1] = S_{2,d}[d]+1)$ and $S_{1,d+k}[d+k+1] = S_{1,d+k}[d+k]+1 (S_{2,d+k}[d+k+1] = S_{2,d+k}[d+k]+1)$. Let's assume $S_{1,d}[d] = S_{2,d}[d] = a$, and $S_{1,d+k}[d+k] = S_{2,d+k}[d+k] = b$. Then in order to achieve a collision according to [12], during the first key appearance, we need $j_{1,d} = d$, $j_{1,d+1} = d + 1$ and $j_{2,d} = d + 1$, $j_{2,d+1} = d$, thus we have:

$$j_{1,d+1} = j_{1,d} + a + 1 + K_1[d+1] \Rightarrow K_1[d+1] = 256 - a$$
$$j_{2,d+1} = j_{2,d} + a + K_2[d+1] \Rightarrow K_2[d+1] = 255 - a$$

For the second key appearance, we will need $j_{1,d+k} = d + k$, $j_{1,d+k+1} = d + k + 1$ and $j_{2,d+k} = d + k + 1$, $j_{2,d+k+1} = d + k$, thus we have:

$$j_{1,d+k+1} = j_{1,d+k} + b + 1 + K_1[d+1] \Rightarrow K_1[d+1] = 256 - b$$
$$j_{2,d+k+1} = j_{2,d+k} + b + K_2[d+1] \Rightarrow K_2[d+1] = 255 - b$$

Now we can observe a contradiction since it is impossible

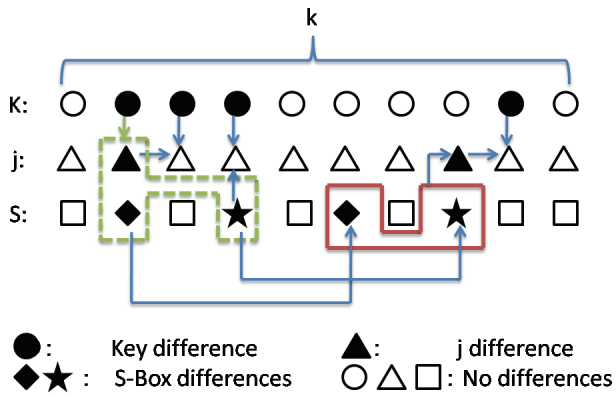


Fig. 2 Self absorbing pattern 1.

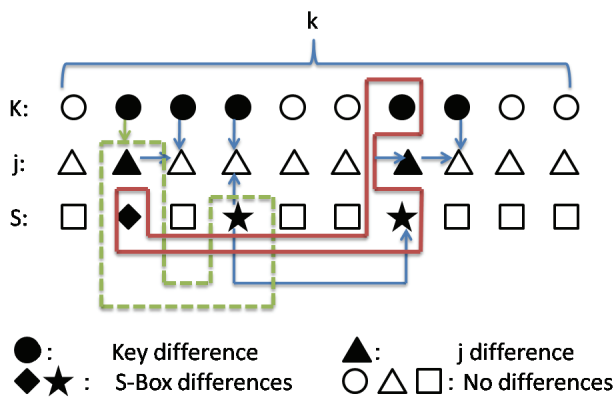


Fig. 3 Self absorbing pattern 2.

for value a to be at index $d + k$ when $i = d + k$, namely, $a \neq b$.

As we will show later that in order to achieve collisions in Self-Absorbing pattern, the second key differential has to be used to absorb the j difference instead of absorbing the S-Box difference in [12]. We point out that Self-Absorbing pattern can be further divided into two sub-patterns, which are shown in Figs. 2 and 3. Due to the self absorbing property, only one key appearance needs to be illustrated, since the others are the same.

Key relations in Self-Absorbing pattern 1:

$$K_2[d] = K_1[d] + t$$
$$K_2[d+1] = K_1[d+1] - t$$
$$K_2[i] = K_1[i] + t, i \in \Gamma \setminus \{\gamma_0, \gamma_1\}.$$

The value difference $t \geq 2$ is the same for all h different positions.

Figure 2 illustrates the case of hamming distance 4 ($h = 4$) and $t = 2$. The first key difference generates three internal differences (dotted line area). In this illustration, the key value difference is $t = 2$, so the interval between two S-Box differences is also required to be t. The second key difference is there to absorb the previous j difference. The third key difference makes the S-Box difference (★) jump to the index just before the last key difference within this key appearance. S-Box difference ♦ should be swapped to the index two intervals from ★ when i touches that index. Then when i touches the S-Box difference ★, two S-Box differences absorb each other and generate a j difference (solid line area). Finally the last key difference is there to absorb the previous j difference, so that the internal states become the same. Table 2 demonstrates one example of 128-byte colliding key pair with hamming distance 4. Two keys differ from each other at indices 1, 2, 3 and 8.

Key relations in Self-Absorbing pattern 2:

$$K_2[d] = K_1[d] + t$$
$$K_2[d+1] = K_1[d+1] - t$$
$$K_2[i] = K_1[i] + t, i \in \Gamma \setminus \{\gamma_0, \gamma_1, \gamma_{h-2}, \gamma_{h-1}\}$$
$$K_2[\gamma_{h-2}] = K_1[\gamma_{h-2}] + t', t' = \gamma_2 - \gamma_{h-2}$$
$$K_2[\gamma_{h-1}] = K_1[\gamma_{h-1}] + t'', t'' = \gamma_{h-2} - \gamma_0$$

For the previous $h-2$ different positions, the value difference $t \geq 2$ is the same. The last two value differences t' and t'', which are determined by the specific Γ values, can be different values other than t.

Self-Absorbing pattern 2 is almost the same as Self-Absorbing pattern 1, except that in addition to using S-Box differences themselves, it also depends on key differences to absorb the S-Box differences (shown in solid line area in Fig. 3) at the final stage. This will allow a more flexible way on how the key value difference can vary, namely, the value difference can choose different values instead of a fixed value, as in the Transitional pattern and Self-Absorbing pattern 1. The relation between t', t'' and Γ can be easily derived from Fig. 3. At step $i = \gamma_{h-2}$, we have $j_{1,\gamma_{h-2}} = j_{1,\gamma_{h-2}-1} + \gamma_2 + K_1[\gamma_{h-2}]$ and $j_{2,\gamma_{h-2}} = j_{2,\gamma_{h-2}-1} + \gamma_0 + K_2[\gamma_{h-2}]$. And a collision requires that $j_{1,\gamma_{h-2}-1} = j_{2,\gamma_{h-2}-1}$ and $j_{1,\gamma_{h-2}} = \gamma_{h-2}$, $j_{2,\gamma_{h-2}} = \gamma_0$, thus $K_2[\gamma_{h-2}] = K_1[\gamma_{h-2}] + (\gamma_2 - \gamma_{h-2})$. For step $i = \gamma_{h-1}$, we need the key difference to absorb the previous j difference. We have $j_{1,\gamma_{h-1}} = j_{1,\gamma_{h-2}} + S_1[\gamma_{h-1}] + K_1[\gamma_{h-1}]$, and $j_{2,\gamma_{h-1}} = j_{2,\gamma_{h-2}} + S_2[\gamma_{h-1}] + K_2[\gamma_{h-1}]$, and the collision requires $S_1[\gamma_{h-1}] = S_2[\gamma_{h-1}]$, $j_{1,\gamma_{h-1}} = j_{2,\gamma_{h-1}}$, $j_{1,\gamma_{h-2}} = \gamma_{h-2}$ and $j_{2,\gamma_{h-2}} = \gamma_0$, we derive the relation $K_2[\gamma_{h-1}] = K_1[\gamma_{h-1}] + (\gamma_{h-2} - \gamma_0)$.

Table 3 shows a 128-byte colliding key pair example with hamming distance 5. The two keys differ from each other at indices 1, 2, 3, 5 and 6. The relation between the t', t'' and Γ can be verified by $K_2[5] = K_1[5] + (3 - 5) = K_1[5] - 2$, and $K_2[6] = K_1[6] + (5 - 1) = K_1[6] + 4$, which matches the example in Table 3.

Three concrete colliding key pairs with key length 128 and hamming distances 3, 4 and 5 in Transitional pattern, Self-Absorbing pattern 1 and 2 are given in the Appendix B.

4. Probability Evaluation

In this section, we evaluate the existence probabilities of RC4 colliding key pairs, and give approximate statistics on the scale and distribution of these keys.

Table 2 Self-absorbing pattern 1, $h = 4, n = 2 (k = 128)$.

			Internal State																		Difference
i	$K_1[i]/K_2[i]$	$j_{1,i}/j_{2,i}$	0	1	2	3	4	5	6	7	8	$\cdots$	129	130	131	132	133	134	135	136	
0	$K_1[0]$	*		1		3															Same
	$K_2[0] = K_1[0]$	*		1		3															
1	$K_1[1]$	1		1		3															j, S-Box
	$K_2[1] = K_1[1] + 2$	3		3		1															
2	$K_1[2]$	*		1		3															S-Box
	$K_2[2] = K_1[2] - 2$	*		3		1															
3	$K_1[3]$	7		1						3											S-Box
	$K_2[3] = K_1[3] + 2$	7		3						1											
5	$K_1[5]$	1						1		3											S-Box
	$K_2[5] = K_1[5]$	1						3		1											
7	$K_1[7]$	7						1		3											j
	$K_2[7] = K_1[7]$	5						1		3											
8	$K_1[8]$	*						1		3											Same
	$K_2[8] = K_1[8] + 2$	*						1		3											
129	$K_1[1]$	129											129		131						j, S-Box
	$K_2[1] = K_1[1] + 2$	131											131		129						
130	$K_1[2]$	*											129		131						S-Box
	$K_2[2] = K_1[2] - 2$	*											131		129						
131	$K_1[3]$	135											129						131		S-Box
	$K_2[3] = K_1[3] + 2$	135											131						129		
133	$K_1[5]$	129															129		131		S-Box
	$K_2[5] = K_1[5]$	129															131		129		
135	$K_1[7]$	135															129		131		j
	$K_2[7] = K_1[7]$	133															129		131		
136	$K_1[8]$	*															129		131		Same
	$K_2[8] = K_1[8] + 2$	*															129		131		

Table 3 Self-absorbing pattern 2, $h = 5, n = 2 (k = 128)$.

			Internal State															Difference
i	$K_1[i]/K_2[i]$	$j_{1,i}/j_{2,i}$	0	1	2	3	4	5	6	$\cdots$	129	130	131	132	133	134	
0	$K_1[0]$	*		1		3											Same
	$K_2[0] = K_1[0]$	*		1		3											
1	$K_1[1]$	1		1		3											j, S-Box
	$K_2[1] = K_1[1] + 2$	3		3		1											
2	$K_1[2]$	*		1		3											S-Box
	$K_2[2] = K_1[2] - 2$	*		3		1											
3	$K_1[3]$	5		1				3									S-Box
	$K_2[3] = K_1[3] + 2$	5		3				1									
5	$K_1[5]$	5		1				3									j
	$K_2[5] = K_1[5] - 2$	1		1				3									
6	$K_1[6]$	*		1				3									Same
	$K_2[6] = K_1[6] + 4$	*		1				3									
129	$K_1[1]$	129									129		131				j, S-Box
	$K_2[1] = K_1[1] + 2$	131									131		131				
130	$K_1[2]$	*									129		131				S-Box
	$K_2[2] = K_1[2] - 2$	*									131		129				
131	$K_1[3]$	133									129				131		S-Box
	$K_2[3] = K_1[3] + 2$	133									131				129		
133	$K_1[5]$	133									129				131		j
	$K_2[5] = K_1[5] - 2$	129									129				131		
134	$K_1[6]$	*									129				131		Same
	$K_2[6] = K_1[6] + 4$	*									129				131		

4.1 Transitional Pattern

For most of the key lengths, they cannot divide 256 except lengths $256, 128, 64, 32, 16$ (ignore the rest smaller ones which are not used for practical reasons). Thus part of the key will repeat $\lfloor \frac{256}{k} \rfloor$ times while the rest part will repeat $\lfloor \frac{256}{k} \rfloor + 1$ times. Due to this phenomenon, the probability will vary depending on the exact locations of the hamming distances, which brings in a lot of unnecessary parameters. For the simplicity of the description without loosing the rigorous evaluation, we compute the upper bound and lower bound of the colliding key pairs with only two input parameters, namely, the key length k and the hamming distance h. The upper bound probability can be computed by chopping the tail part $256 - k \times \lfloor \frac{256}{k} \rfloor$ of the S-Box and treat the

key as it repeats only $\lfloor \frac{256}{k} \rfloor$ times during KSA. The lower bound, on the other hand, can be computed by adding length $k - 256 + k \times \lfloor \frac{256}{k} \rfloor$ to the end of the S-Box and treat the key as it repeats $\lfloor \frac{256}{k} \rfloor + 1$ times during KSA. Then for a given hamming distance h, we can compute the collision probability which is not related to the locations of the hamming distance anymore, and finally we sum up the probability for all cases of the hamming distances. Note that for the key lengths dividing 256, the exact evaluation is given (the upper bound is equal to the lower bound).

From the previous analysis, we know that colliding key pairs have the property that the key value difference is fixed at one, and the hamming distance can vary. We divide the whole process into three phases as shown in Fig. 1, namely, the starting phase (first appearance of the key), the ending phase (last appearance of the key) and the repeating phase (middle repeating parts).

Starting Phase. First, before i touches γ_0, j can not touch γ_0 or $\gamma_0 + 1$ with probability $(\frac{254}{256})^{\gamma_0}$. When i touches γ_0, we need $j_1 = \gamma_0$ and $j_2 = \gamma_0 + 1$ with probability $\frac{1}{256}$. When $i = \gamma_0 + 1$, the S-Box difference at index i needs to be swapped to the next key difference index γ_1, namely, $j_{1,\gamma_0+1} = j_{2,\gamma_0+1} = \gamma_1$ with probability $\frac{1}{256}$.

For each of the other key difference indices $\gamma_n \in \Gamma$, we will pay the probability $\frac{1}{256}$ each to let the S-Box difference at $\gamma_{n \bmod h}$ swap to $\gamma_{n+1 \bmod h}$ when $i = \gamma_n$. This gives us a total probability $(\frac{1}{256})^{h-1}$. When i is between two consecutive key difference indices, the pattern requires that j does not touch the later key difference index, otherwise i will never be able to touch the later S-Box difference again. This will add $(\frac{255}{256})^{\gamma_1-\gamma_0-2}(\frac{255}{256})^{\gamma_2-\gamma_1-1}\cdots(\frac{255}{256})^{\gamma_{h-1}-\gamma_{h-2}-1}(\frac{255}{256})^{k+\gamma_0-\gamma_{h-1}-1} = (\frac{255}{256})^{k-h-1}$ to the total probability cost. Thus, the total probability in the starting phase is $(\frac{1}{256})^{h+1}(\frac{254}{256})^{\gamma_0}(\frac{255}{256})^{k-h-1}$.

Repeating Phase. For upper bound, key will appear $\lfloor \frac{256}{k} \rfloor - 2$ times, and for lower bound, it will appear $\lfloor \frac{256}{k} \rfloor - 1$ times during the repeating phase. For each key appearance, the procedure is as follows. When i touches one key difference index, $\frac{1}{256}$ probability will be paid, $(\frac{1}{256})^h$ in total. When i is between two difference indices, it is not allowed to touch the later one (same as starting phase), this will add probability $(\frac{255}{256})^{k-h}$ in the repeating phase. Thus, the probability that one key appearance must pay is $(\frac{1}{256})^h(\frac{255}{256})^{k-h}$. The total probability is $((\frac{1}{256})^h(\frac{255}{256})^{(k-h)})^{\lfloor \frac{256}{k} \rfloor - 1}$ for the lower bound, and $((\frac{1}{256})^h(\frac{255}{256})^{(k-h)})^{\lfloor \frac{256}{k} \rfloor - 2}$ for the upper bound.

Ending Phase. When i touches $\{\gamma_0 + k \times (\lfloor \frac{256}{k} \rfloor - 1), \cdots, \gamma_{h-3} + k \times (\lfloor \frac{256}{k} \rfloor - 1)\}$ (upper bound), or $\{\gamma_0 + k \times (\lfloor \frac{256}{k} \rfloor), \cdots, \gamma_{h-3} + k \times (\lfloor \frac{256}{k} \rfloor)\}$ (lower bound), with probability $\frac{1}{256}$ each, j will touch the next key difference. When $i = \gamma_{h-2} + k \times (\lfloor \frac{256}{k} \rfloor - 1)$ (upper bound), or $i = \gamma_{h-2} + k \times (\lfloor \frac{256}{k} \rfloor)$ (lower bound), with $\frac{1}{256}$, j should touch the index $\gamma_{h-1} - 1 + k \times (\lfloor \frac{256}{k} \rfloor - 1)$ (upper bound), or $\gamma_{h-1} - 1 + k \times (\lfloor \frac{256}{k} \rfloor)$ (lower bound). When $i = \gamma_{h-1} - 2 + k \times (\lfloor \frac{256}{k} \rfloor - 1)$ (upper bound), or $i = \gamma_{h-1} - 2 + k \times (\lfloor \frac{256}{k} \rfloor)$ (lower bound), another S-Box difference should be swapped to here with probability $\frac{1}{256}$. And when $i = \gamma_{h-1} - 1 + k \times (\lfloor \frac{256}{k} \rfloor - 1)$ (upper bound), or $i = \gamma_{h-1} - 1 + k \times (\lfloor \frac{256}{k} \rfloor)$ (lower bound), it is required that $j = \gamma_{h-1} - 1 + k \times (\lfloor \frac{256}{k} \rfloor - 1)$ (upper bound), or $j = \gamma_{h-1} - 1 + k \times (\lfloor \frac{256}{k} \rfloor)$ (lower bound). Up to now, this gives us the successful probability $(\frac{1}{256})^{h+1}$. And as before, when i is between two key differences, j should not touch the latter key difference index. This will add $(\frac{255}{256})^{\gamma_1-\gamma_0-1} \cdot (\frac{255}{256})^{\gamma_2-\gamma_1-1} \cdots (\frac{255}{256})^{\gamma_{h-1}-\gamma_{h-2}-2} = (\frac{255}{256})^{\gamma_{h-1}-\gamma_0-h}$ to the probability. Thus the probability for the Ending Phase is $(\frac{1}{256})^{h+1}(\frac{255}{256})^{\gamma_{h-1}-\gamma_0-h}$.

By multiplying them together, we have the following theorem.

Theorem 1: Given key length k and hamming distance $h \geq 1$, the probability of two keys with relations in Transitional pattern forming a colliding key pair, $P_T(k, h, \gamma_{h-1})$, can be approximated as follows:

$$P_T(k,h,\gamma_{h-1}) \in [(\tfrac{1}{256})^{h\cdot(\lfloor \frac{256}{k} \rfloor+1)+2} \times (\tfrac{255}{256})^{(k-h)\cdot\lfloor \frac{256}{k} \rfloor+\gamma_{h-1}-2}, (\tfrac{1}{256})^{h\cdot\lfloor \frac{256}{k} \rfloor+2} \times (\tfrac{255}{256})^{(k-h)\cdot(\lfloor \frac{256}{k} \rfloor-1)+\gamma_{h-1}-2}]$$

In order to give the total number of the colliding key pairs, we need to sum up the probability for all the possible different hamming distance locations, and multiply the total number of the keys ($2^{8\times k}$). Transitional Pattern requires that $\gamma_1 - \gamma_0 \geq 2$ and $\gamma_{h-1} - \gamma_{h-2} \geq 3$ when $h \geq 2$. Then for a given key length k and a hamming distance h, the legal number of all the possible different hamming distance combinations is $\binom{\gamma_{h-1}-3}{h-1}$. Then we can derive the number of colliding key pairs given k and h, which is given in Theorem 2.

Theorem 2: The number of colliding key pairs with key length k and hamming distance $h \geq 1$ can be computed by the following formula:

$$Pairs_T(k,h) = \begin{cases} 2^{8k} \times \sum_{\gamma_{h-1}=k-1}^{h+2} P_T(k,h,\gamma_{h-1}) \times \binom{\gamma_{h-1}-3}{h-1}, & h \geq 2 \\ 2^{8k} \times \sum_{\gamma_{h-1}=k-1}^{0} P_T(k,h,\gamma_{h-1}), & h = 1 \end{cases}$$

Theorem 2 covers all the general cases of the colliding key pairs in Transitional Pattern. For $h = 1$, it is the special case shown in [2], and our evaluation matches the data provided in [2] very well.

4.2 Self-Absorbing Pattern 1

Compared with the Transitional Pattern, Self-Absorbing patterns have even more parameters, namely, value differences. In order to exploit the relation in a clear way, we again restrict the input parameters to only k and h as in the Transitional Pattern by giving the upper and lower bound of t.

We only need to evaluate the probability of one key appearance, because the other parts just repeat the first key appearance procedure. Different from Transitional Pattern,

Self-Absorbing Patterns require all the key differences to repeat the same time during the KSA, otherwise collision cannot be achieved. For any given key length k, indices between $[0, 256-k\times\lfloor\frac{256}{k}\rfloor-1]$ will repeat $\lfloor\frac{256}{k}\rfloor+1$ times, and indices between $[256-k\times\lfloor\frac{256}{k}\rfloor, k-1]$ will repeat themselves $\lfloor\frac{256}{k}\rfloor$ times. Thus all the key differences can only exist in either of these two internals at the same time.

First we compute the probability for the first key appearance. Before i touches index d, we need $S_d[d]+t = S_d[d+t]$ with probability $\frac{255}{256}\times(\frac{254}{256})^{d-1}+\frac{1}{256}$ (Refer to Lemma 1 in the Appendix A for the proof). When i touches d, we require $j_d = d$ with probability $\frac{1}{256}$. We need one of the differences (★ in Figure 2) to appear at indices $\{\Gamma\bigcup\{\gamma_{h-1}-1\}\}\setminus\{\gamma_0,\gamma_1,\gamma_{h-1}\}$ when i touches them, and j cannot touch the later key difference position when i is between the consecutive two of them. The probability can be calculated in the same way as in the repeating phase in Transitional pattern, namely, $(\frac{1}{256})^{h-3}(\frac{255}{256})^{\gamma_{h-1}-\gamma_0-h+1}$. Also we need the S-Box difference (♦) to be at position $\gamma_{h-1}-t-1$ and it cannot be touched when i is between $\gamma_{h-1}-t-1$ and $\gamma_{h-1}-1$. So this will give us probability $\frac{1}{256}(\frac{255}{256})^{t-1}$. Finally, when i touches index $\gamma_{h-1}-1$, we need $j_{\gamma_{h-1}-1} = \gamma_{h-1}-1$ with probability $\frac{1}{256}$. By multiplying them together, we can derive the probability of one key appearance $(\frac{1}{256})^h(\frac{255}{256})^{\gamma_{h-1}-\gamma_0-h+t}(\frac{255}{256}(\frac{254}{256})^{\gamma_0-1}+\frac{1}{256})$, which can be further approximated as $(\frac{1}{256})^h\times(\frac{255}{256})^{\gamma_{h-1}-h+t}$. γ_{h-1} will disappear in the final evaluation formula since we will sum up the probabilities for all the locations. However, t is still not fixed. Since $(\frac{1}{256})^h$ is the dominant part, one way to approximate is just to eliminate the t parameter. Here in order to be more accurate, we give the upper and lower bound of t. According to the pattern, $t\geq 2$. Given γ_{h-1}, h, the max t can be denoted as $t\leq\gamma_{h-1}-h+2$.

Theorem 3: Given hamming distance $h\geq 3$, γ_{h-1} and t, the probability of two keys with relations in Self-Absorbing pattern 1 to achieve collision for one key appearance, $P_{S1}(h,\gamma_{h-1},t)$, can be approximated as follows:

$$P_{S1}(h,\gamma_{h-1},t) = \left(\frac{1}{256}\right)^h\times\left(\frac{255}{256}\right)^{\gamma_{h-1}-h+t}$$

where $t\in[2,\gamma_{h-1}-h+2]$.

Next is to compute the combinations of the hamming distances. The rule here is that due to the value t, we have $t-1$ positions that key differences cannot be placed, and $\gamma_1-\gamma_0=1$. For the first interval $[0, 256-k\times\lfloor\frac{256}{k}\rfloor-1]$, $\gamma_{h-1}\in[3, 256-k\times\lfloor\frac{256}{k}\rfloor-1]$ (make sure that all the hamming distances are inside the interval), we have $\binom{\gamma_{h-1}-t}{h-2}$ different ways to place the $h-2$ key differences (exclude γ_{h-1} and γ_0 and γ_1 can be treated as one). For the latter interval, $\gamma_{h-1}\in[256-k\times\lfloor\frac{256}{k}\rfloor+3, k-1]$, and we have $\binom{\gamma_{h-1}-t-(256-k\times\lfloor\frac{256}{k}\rfloor)}{h-2}$ different ways. Now we are ready to have the following theorem.

Theorem 4: For Self-Absorbing pattern 1, the number of colliding key pairs with key length k and hamming distance $h\geq 3$ can be computed by the following formula:

$$Pairs_{S1}(k,h) = 2^{8k}\times\Big(\sum_{\gamma_{h-1}=h-1}^{256-k\times\lfloor\frac{256}{k}\rfloor-1}(P_{S1}(h,\gamma_{h-1},t))^{\lfloor\frac{256}{k}\rfloor+1}\times\binom{\gamma_{h-1}-t}{h-2}+\sum_{\gamma_{h-1}=256-k\times\lfloor\frac{256}{k}\rfloor+h-1}^{k-1}(P_{S1}(h,\gamma_{h-1},t))^{\lfloor\frac{256}{k}\rfloor}\times\binom{\gamma_{h-1}-t-(256-k\times\lfloor\frac{256}{k}\rfloor)}{h-2}\Big)$$

for $t\in[2,\gamma_{h-1}-h+2]$.

4.3 Self-Absorbing Pattern 2

The way we evaluate Self-Absorbing Pattern 2 is very similar to Self-Absorbing pattern 1 due to the similarity between the two patterns. We point out the differences here. First, we don't need the S-Box difference (♦) to be at position $\gamma_{h-1}-t-1$ any more, since the key difference at index γ_{h-2} will adjust to absorb the S-Box differences. This will cut the probability $\frac{1}{256}(\frac{255}{256})^{t-1}$. Also, the pattern requires one of the S-Box difference at index γ_0 remain unchanged until it is swapped to index γ_{h-2} at this step. This makes us to add probability $(\frac{255}{256})^{\gamma_{h-1}-\gamma_0-2}$. To sum up, $P_{S2}(h,\gamma_{h-1},\gamma_0)$ can be approximated as

$$P_{S2}(h,\gamma_{h-1},\gamma_0) = (\frac{1}{256})^{h-2}\times(\frac{255}{256})^{2\gamma_{h-1}-\gamma_0-h-3}$$

Notice that compared with the Self-Absorbing pattern 1, the dominant part of the probability increases to $(\frac{1}{256})^{h-2}$, while we only save one $\frac{1}{256}$ due to the unnecessary S-Box swap. This is because Self-Absorbing pattern 2 can be seen as a Self-Absorbing pattern 1 with one extra key differential added to the index $\gamma_{h-1}-1$, and this new added key differential does not contribute to the probability cost. The number of colliding key pairs is shown in Theorem 5.

Theorem 5: For Self-Absorbing pattern 2, the number of colliding key pairs with key length k and hamming distance $h\geq 5$ can be computed by the following formula:

$$Pairs_{S2}(k,h) = 2^{8k}\times\Big(\sum_{\gamma_{h-1}=h-1}^{256-k\times\lfloor\frac{256}{k}\rfloor-1}(P_{S2}(h,\gamma_{h-1},\gamma_0))^{\lfloor\frac{256}{k}\rfloor+1}\times\binom{\gamma_{h-1}-\gamma_0-1}{h-2}+\sum_{\gamma_{h-1}=256-k\times\lfloor\frac{256}{k}\rfloor+h-1}^{k-1}(P_{S2}(h,\gamma_{h-1},\gamma_0))^{\lfloor\frac{256}{k}\rfloor}\times\binom{\gamma_{h-1}-\gamma_0-1-(256-k\times\lfloor\frac{256}{k}\rfloor)}{h-2}\Big)$$

for $\gamma_0\in[0,\gamma_{h-1}-h+1]$.

One thing to notice is that from the theoretical evaluation, we can observe that the collision probability (complexity) of Self-Absorbing pattern 1 with hamming distance h is almost equal to the one of Self-Absorbing pattern 2 with hamming distance $h+2$ given the same k.

We can conclude that the probabilities of both Transitional pattern and Self-Absorbing patterns are mainly affected by hamming distance h and length of the secret key k. The probability decreases as the hamming distance h becomes larger or the key length k becomes shorter (n becomes larger). Note that to achieve a collision in Transitional pattern, Self-Absorbing pattern 1 and 2, the corresponding hamming distances are required to be $h\geq 1$,

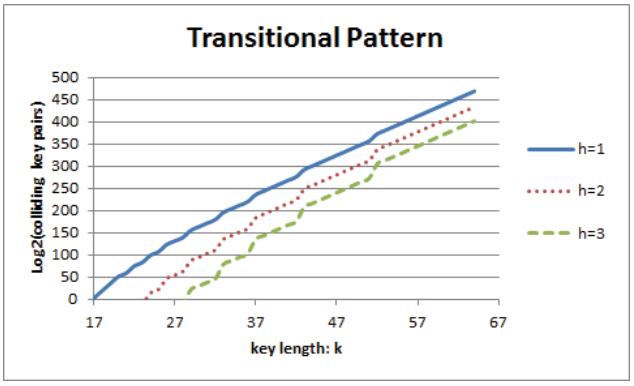


Fig. 4 Colliding key pairs for transitional pattern.

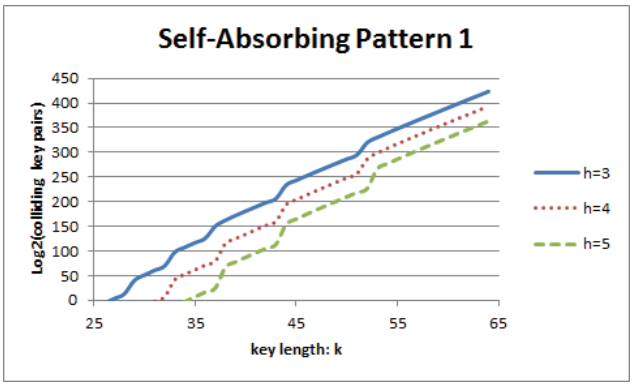


Fig. 5 Colliding key pairs for self-absorbing pattern 1.

$h \geq 3$ and $h \geq 5$, respectively. This can be easily seen from Fig. 1 and Fig. 2 for Transitional pattern and Self-Absorbing pattern 1. For Self-Absorbing pattern 2, it seems that h could start from 4. In this case, according to the pattern, we have $j_{1,\gamma_{h-2}} = j_{1,\gamma_{h-2}-1} + \gamma_{h-2} + K_1[\gamma_{h-2}]$, $j_{2,\gamma_{h-2}} = j_{1,\gamma_{h-2}-1} + \gamma_0 + K_2[\gamma_{h-2}]$ and $j_{1,\gamma_{h-2}-1} = j_{2,\gamma_{h-2}-1}$. And also $j_{1,\gamma_{h-2}} = \gamma_{h-2}$, $j_{2,\gamma_{h-2}} = \gamma_0$. As a result, $K_1[\gamma_{h-2}] = K_2[\gamma_{h-2}]$, namely the key differential at index $\gamma_{h-1} - 1$ disappears and it becomes the Self-Absorbing pattern 1 with $h = 3$. That's why for Self-Absorbing pattern 2, we require $h \geq 5$.

Figures 4, 5 and 6 give the number of colliding key pairs for different key lengths and hamming distances according to the previous theorems. The data shown in the figures are the averages of the upper and lower bound of the estimation.

4.4 Experimental Evaluation

To confirm that our previous theoretical analysis is correct, we give the experimental evaluation here. Our goal is to evaluate the average collision probability for each of the three patterns given some k and h. For short key size with large h value, the collision probability is so small that it is impossible to carry out the experiment. For example, for Transitional pattern with h as small as 2, even the key is repeated only twice during KSA such as $k = 128$, the estimated probability is about 2^{-48} which is too small to evaluate the average probability (we can hardly collect sufficient data). Thus we target full length key with $k = 256$, and select h such that the estimated probability is around 2^{-32} which falls into the practical implementation scope. The theoretical collision probability can be easily computed by taking the average value of $P_T(k, h, \gamma_{h-1})$, $P_{S1}(h, \gamma_{h-1}, t)$ and $P_{S2}(h, \gamma_{h-1}, \gamma_0)$ for each of the three patterns. And once we confirm the correctness of these three collision probabilities, the number of colliding key pairs $Pairs_T(k, h)$, $Pairs_{S1}(k, h)$ and $Pairs_{S2}(k, h)$ can be trivially confirmed since they are derived directly from the collision probabilities. The corresponding theoretical probabilities are shown as follows.

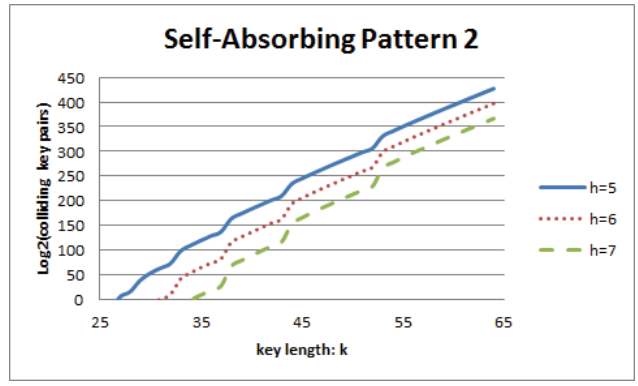


Fig. 6 Colliding key pairs for self-absorbing pattern 2.

Transitional Pattern ($k = 256$, $h = 2$). During the KSA, all the hamming distance will appear once, and the probability is equivalent to the upper bound of the $P_T(k, h, \gamma_{h-1})$. Thus the theoretical collision probability can be computed by taking the average value of $P_T(k, h, \gamma_{h-1})$ for each $\gamma_{h-1} \in [4, 255]$, which can be denoted as $Ave(P_T(k, h, \gamma_{h-1}))$ as follows:

$$Ave(P_T(k, h, \gamma_{h-1})) \approx \frac{\sum_{\gamma_{h-1}=4}^{255} (\frac{1}{256})^4 \times (\frac{255}{256})^{\gamma_{h-1}-2}}{252} = 2^{-32.6619}$$

Self-Absorbing Pattern 1 ($k = 256$, $h = 4$). For $k = 256$, all the key differentials locate within the same interval, thus the theoretical probability can be computed by taking the average value of $P_{S1}(h, \gamma_{h-1}, t)$ for each $(t, \gamma_{h-1}) \in [2, \gamma_{h-1} - 2] \times [3, 255]$, which can be denoted as $Ave(P_{S1}(h, \gamma_{h-1}, t))$ as follows:

$$Ave(P_{S1}(h, \gamma_{h-1}, t)) \approx \frac{\sum_{\gamma_{h-1}=3}^{255} \sum_{t=2}^{\gamma_{h-1}-2} (\frac{1}{256})^4 \times (\frac{255}{256})^{\gamma_{h-1}-4+t}}{31878} = 2^{-33.3234}$$

Self-Absorbing Pattern 2 ($k = 256$, $h = 6$). Same as Self-Absorbing pattern 1, all the key differentials locate within the same interval, thus the theoretical probability can be computed by taking the average value of $P_{S2}(h, \gamma_{h-1}, \gamma_0)$ for each $(\gamma_0, \gamma_{h-1}) \in [0, \gamma_{h-1}] \times [5, 255]$, which can be denoted as $Ave(P_{S2}(h, \gamma_{h-1}, \gamma_0))$ as follows:

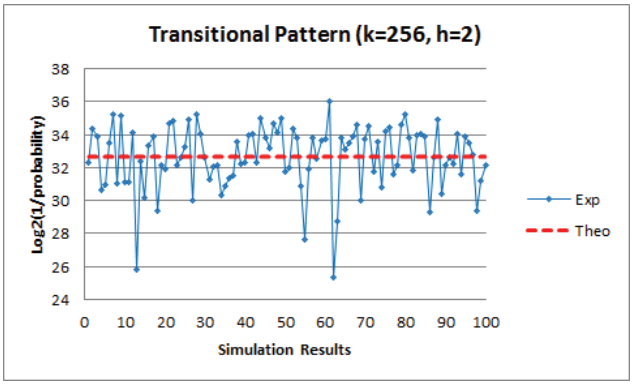


Fig. 7 Experimental result for transitional pattern.

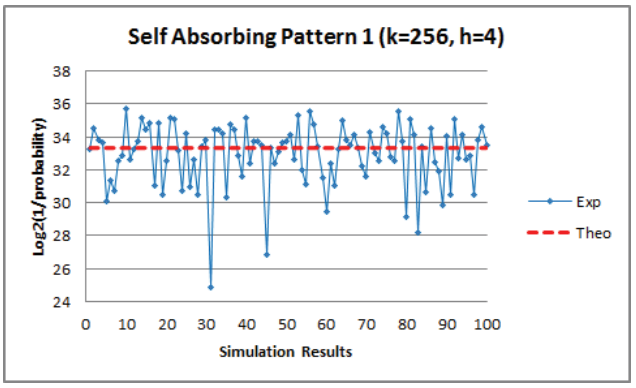


Fig. 8 Experimental result for self-absorbing pattern 1.

Table 4 Collision probability comparison.

Pattern	k	h	Theoretical Probability	Average Experimental Probability
Tran	256	2	$2^{-32.6619}$	$2^{-32.6253}$
Self1	256	4	$2^{-33.3234}$	$2^{-32.9144}$
Self2	256	6	$2^{-33.3017}$	$2^{-33.3571}$

$$Ave(P_{S2}(h,\gamma_{h-1},\gamma_0)) \approx \frac{\sum_{\gamma_{h-1}=5}^{255}\sum_{\gamma_0=0}^{\gamma_{h-1}-5}(\frac{1}{256})^4\times(\frac{255}{256})^{2\gamma_{h-1}-\gamma_0-9}}{31626} = 2^{-33.3017}$$

For the experiment, we fix the k and h according to the ones we choose previously. Then we randomly generate related key pairs and hamming locations Γ according to the three patterns until a collision is found. The experimental collision probability is one over the number of generated related key pairs. 100 experimental results are collected for each of the three patterns on parallel computer SGI Altix4700 with 20 cores used (Dual Core Intel Itanium Series 9000 1.67 GHz), which takes around one week time. The simulation results are shown in Figs. 7, 8 and 9, and the comparison between the theoretical and experimental probability is summarized in Table 4, which confirms the correctness of our evaluation.

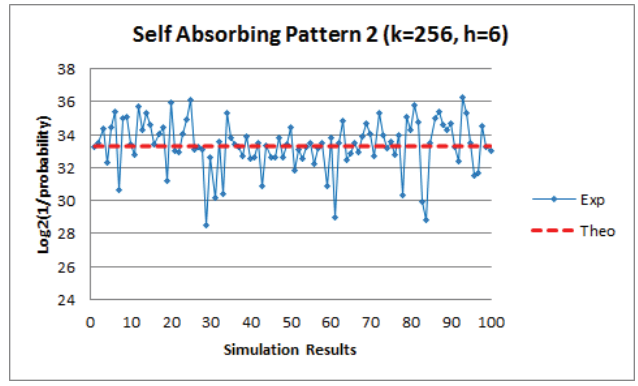


Fig. 9 Experimental result for self-absorbing pattern 2.

5. Vulnerability Inducing by RC4 Key Collision

In this section, we demonstrate how the key collision can influence the security of a hash function using RC4 proposed in INDOCRYPT 2006. The "RC4-Hash" followed the "wide pipe" hash function design principle proposed by Lucks [13] and was claimed to be as efficient as some widely-used hash functions, such as SHA-family and MD-family, while also ruling out all possible generic attacks against those famous hash functions. First hash collision was found in [16] by exploiting the idea of Finney States [15]. We first briefly describe the RC4-Hash algorithm, and then we give the collision analysis based on RC4 key collision. For a more detailed description of the hash function, please refer to [14].

5.1 RC4-Hash

$\{0,1\}^{<2^{64}}$ denotes the set of all messages whose length is at most $2^{64}-1$. l is the output length of the RC4 hash function, $16 \le l \le 64$. RC4-Hash function can be described as $\{0,1\}^{<2^{64}} \to \{0,1\}^{8l}$.

Padding Rule: $\mathrm{pad}(M) = \mathrm{bin}_8(l)\|M\|1\|0^k\|\mathrm{bin}_{64}(|M|) = M_1\|\cdots\|M_t$, where M_t is the last 512-bit block. $\mathrm{bin}_{64}(|M|)$ is the 64-bit binary representation of the number of bits of M. k is the least non-negative integer such that $8+|M|+1+k+64 \equiv 0 \bmod (512)$ and $|M_i| = 512$.

Iteration Phase: Let $(S_0, j_0) = (S^{IV}, 0)$ be an initial value. The compression function C is invoked iteratively as follows:

$$(S_0, j_0) \xrightarrow{M_1} (S_1, j_1) \xrightarrow{M_2} \cdots (S_{t-1}, j_{t-1}) \xrightarrow{M_t} (S_t, j_t)$$

where $(S, j) \xrightarrow{X} (S^*, j^*)$ denotes $C((S, j), X) = (S^*, j^*)$.

Post-Processing: Let (S_t, j_t) be the internal state after the classical iteration. Compute $S_{t+1} = S_0 \circ S_t$ and $j_{t+1} = j_t$. Then compute $HBG_l(OWT(S_{t+1}, j_{t+1}))$.

```
C((S, j), X)
for i = 0 to 255
    j ← j + S[i] + X[r(i)]
    Swap(S[i],S[j]);
Return (S,j)
```

```
OWT((S, j))
Temp1 = S
for i = 0 to 511
    j ← j + S[i]
    Swap(S[i], S[j])
Temp2 = S
S = Temp1 ∘ Temp2 ∘ Temp1
Return (S, j)
```

```
HBG_l((S, j))
for i = 1 to l
    j ← j + S[i]
Swap(S[i], S[j])
Out = S[S[i] + S[j]]
```

$\circ$ denotes the composition of the permutations. Function $r : [256] \rightarrow [64]$ reorders the 64-byte message block. There are four r mapping functions(r_0, r_1, r_2, r_3) corresponding to the four iteration processes for each message block. In other words, each message block is reordered three times (r_0 is the identity permutation) during one iteration process. Refer to appendices for S^{IV} and r_i.

5.2 Collisions for RC4-Hash Function

Let's look at the iteration phase carefully. After message is padded, it is cut into 64-byte blocks, and each block is processed by the compression function C four times. The compression function C is actually the KSA in RC4, and the input message block can be seen as a 64-byte secret key, except for two differences. First, the message block is reordered by using r_i functions three times (instead of using the same 64-byte key which appears 4 times during KSA) and second, instead of the identity permutation used at the beginning of KSA, a shuffled S-Box S^{IV} is used as the initial S-Box. The similarities between the compression function and the KSA give us the intuition that we can make use of the RC4 key collision to find collisions for RC4-Hash. Now let's take a look at how these two differences can affect the collision search. In both Transitional pattern and Self-Absorbing pattern, when i touches the first different position, we need $S_d[d] + t = S_d[d + t]$. This is very easy to achieve when the initial S-Box is an identity permutation (j does not touch index d or index $d + t$ before i touches index d). But still we can make this happen with S^{IV} (Several candidates are available by checking S^{IV} carefully, and we use one of them in the following example). For the transitional pattern, the reordering of the message will not have much effect on finding collisions, because even though the different positions between the two messages change three times, we do not have to pay extra probabilities because there are no restrictions among these different message positions. Thus it works just the same as finding key collisions in the Transitional pattern. However, there are strict relations between the different positions in Self-Absorbing pattern (Self-Absorbing pattern 1: $K_2[d] = K_1[d] + t$, $K_2[d + 1] = K_1[d + 1] - t$ and $K_2[i] = K_1[i] + t$ for $i \in \Gamma \setminus \{\gamma_1, \gamma_2\}$), and the reordering of the message breaks those relations at the later rounds in the compression function, thus making it difficult to find a collision by using this pattern.

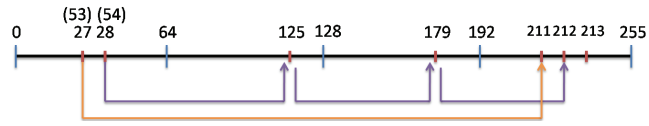


Fig. 10 RC4-Hash collision using transitional pattern.

Here we give a concrete collision example by making use of the Transitional pattern. Since the initial S-Box S^{IV} is not an identity permutation, we need to first make two consecutive indices have the value difference one. There are several candidates we can use, by examining the S^{IV} carefully, we choose to let $S^{IV}[24] = 53$ appear in index 27 when i touches it, and $S^{IV}[28] = 54$ should not be touched by j before. Then we have two values, 53 and 54, next to each other at indices 27 and 28 when i touches index 27. The four iterations of the 64-byte message block during the compression function C can be seen as a KSA procedure with a 64-byte key. Since the message will be reordered three times, we need to check the mapping function r_i to identify the different positions. According to the Transitional pattern, in order to achieve a collision, two messages should differ from each other at index 27, and the value difference should be one. According to the r_1, r_2 and r_3 in the Appendix C, the differences between two messages will appear at indices 125, 179 and 213. After i touches 213, the two internal states become the same. Figure 10 describes the above collision by using Transitional pattern during one compression function C (63-byte message plus one padded byte).

Several message modifications will help to reduce the collision complexity, which are described as follows, assuming $M_2[27] = M_1[27] + 1$, and $M_2[i] = M_1[i]$ for $i \neq 27$. In the first round, by modifying message $M_1[24] = 27 - j_{23} - S_{23}[24]$, $M_1[27] = 230 - j_{26}$ and $M[28] = 44$, all the first round conditions can be satisfied. We also assume that after two S-Box differences are introduced by the message difference at index 27, one of the S-Box difference at index 27 is not touched by j until $i = 211$. This means $j_{211} = 27$, and pattern requires $j_{212} = 212 = j_{211} + 54 + M_1[44]$, thus we can also set $M_1[44] = 131$ in the first round, which will guarantee in the last round that $j_{212} = 212$ is satisfied with probability 1 as long as $j_{211} = 27$ is passed. The rest of the conditions $j_{125} = 179$, $j_{179} = 212$ and $j_{211} = 27$ are determined in the probabilistic way (2^8). This gives us an approximate complexity at around 2^{24}. Both our result and [16] (with complexity 2^9) can give a practical time collision.

Here suppose we have an piece of input message, which after padding, will result in more than one message block. Collision is achieved for the first message block, and we give the first two input message blocks after padding, and the output intermediate states S_1, j_1. We use $l = 16$ and the collision is found on an Intel Core Duo CPU notebook PC within less than one minute.

Message1(Message2): 10 3C 6F 3A 67 55 2A 60 81 10 73 F0 5E D1 04 F4 C7 77 57 22 88 6B F4 3E 7D 28 9F **0C(0D)** 2C E3 12 F2 83 CF 5E CB A4 55 F2 A4 94 3B A2 FB 83 F8 83 E4 91 BD 6E 2B 7A A5 44 48 CF 43 A3 68 24 22 0E 7C
S_1 : F3 E1 CB 22 3C 1A F7 A2 A6 07 7C A7 BD 4D 0C 02 5D 86 04 38 30 D2 53 03 FB 21 D1 24 A8 DF 83 68 F9 8F 43 D3 B3 C4 B6 D9 F0 39 78 0B DD 26 23 AE CC E8 4E 3E 8A E7 18 E5 FF 7B A9 4C A4 88 41 92 D5 14 82 E0 8C 98 61 C5 65 AB 06 46 6F EA 4F E9 BB A0 ED 00 97 49 D8 F2 63 E6 D7 89 45 48 2D B8 C9 01 3F 59 0A C1 C7 E3 EC 62 96 9F AC 52 C3 16 72 FD 7E B4 8E 25 5E 27 67 A3 B7 CA 09 37 33 99 57 DC 2E 42 69 B 2 CF EF F5 70 A5 1F BA 94 58 B9 56 B5 5F 2C 11 EB 3A 2B 9E C0 32 28 7D FE 5C F4 08 34 5B F1 64 0F D6 4A 2A 40 C6 BC 71 2F 50 10 B0 91 A1 DE D4 FA 90 12 84 5A 3B AD 7A 73 AF 05 9D 87 8B 79 BF C8 D0 FC C2 51 E4 1C 66 74 7F 44 31 55 0E 35 36 6 A 9B 77 95 6B 85 54 DA 81 76 13 47 75 8D 15 B1 CD 19 CE DB 20 4B 1E 93 17 1D 80 1B 9C 60 F8 AA 6D 0D EE 3D F6 9A E2 6C BE 29 6E
j_1 : 8E

5.3 Design Principle Discussion

From the above analysis, we can see that the design of the compression function even by modifying the KSA using S^{IV} and mapping function r_i cannot eliminate the KSA collision property. Here we propose one method to mitigate the attack that caused by the RC4 key collision property. The repair only requires us to reduce the length of the message block from 512-bit to 128-bit, than the collision can be completed eliminated. This is because after reducing to 128-bit (16-byte), each message block will be scrambled 16 times, and this has been proved by Theorem 2 that no collision is possible any more under this setting. In other words, we provide, to some degree, a provable-collision-resistance repair to the problem. However, we must point out that the efficiency suffers from this modification and thus leaves space for designing efficient and secure RC4-based compression function as future work.

6. Conclusion

In this paper, we have shown that RC4 has a vulnerability that generates many colliding key pairs with various hamming distances. We analyze the behavior of these colliding key pairs and formalized them into two patterns, which include the newly discovered colliding key pairs we found, and also the ones found in previous research. We further estimate the numbers for all the RC4 colliding key pairs, and clarify the relations among the number of colliding key pairs, key length and hamming distances. Finally, we show how the RC4 key collision patterns can be used to find hash collisions for RC4-Hash which was proposed at INDOCRYPT 2006.

References

[1] A.L. Grosul and D.S. Wallach, A Related-Key Cryptanalysis of RC4. Technical Report TR-00-358, Department of Computer Science, Rice University, 2000, http://cohesion.rice.edu/engineering/computerscience/tr/TR Download.cfm?SDID=126
[2] M. Matsui, Key Collisions of the RC4 Stream Cipher. Dunkelman, O., Preneel, B. (eds.) FSE 2009. LNCS, vol.5665, pp.1–24, Springer, Heidelberg, 2009.
[3] A. Miyaji and M. Sukegawa, "New analysis based on correlations of RC4 PRGA with nonzero-bit differences," IEICE Trans. Fundamentals, vol.E93-A, no.6, pp.1066–1077, June 2010.
[4] Anonymous: "RC4 source code." CypherPunks mailing list (September 9, 1994), http://cypherpunks.venona.com/date/1994/09/msg00304.html, http://groups.google.com/group/sci.crypt/msg/10a300c9d21afca0
[5] A. Roos, "A class of weak keys in the RC4 stream cipher 1995," http://marcel.wanda.ch/Archive/WeakKeys
[6] I. Mantin and A. Shamir, "A practical attack on broadcast RC4," M. Matsui, (ed.) FSE 2001. LNCS, vol.2355, pp.152–164, Springer, Heidelberg, 2001.
[7] S. Paul and B. Preneel, "A new weakness in the RC4 keystream generator and an approach to improve security of the cipher," B. Roy, W. Meier, (eds.) FSE 2004. LNCS, vol.3017, pp.245–259, Springer, Heidelberg, 2004.
[8] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," S. Vaudenay, A.M. Youssef, (eds.) SAC 2001. LNCS, vol.2259, pp.1–24, Springer, Heidelberg, 2001.
[9] A. Klein, "Attacks on the RC4 stream cipher," Designs, Codes and Cryptography, vol.48, no.3, pp.269–286, 2008.
[10] E. Tews, R.P. Weinmann, and A. Pyshkin, "Breaking 104 Bit WEP in less than 60 seconds," S. Kim, M. Yung, H.-W. Lee, (eds.) WISA 2007. LNCS, vol.4867, pp.188–202, Springer, Heidelberg, 2007.
[11] S. Vaudenay and M. Vuagnoux, "Passive-only key recovery attacks on RC4," C. Adams, A. Miri, M. Wiener, (eds.) SAC 2007. LNCS, vol.4876, pp.344–359, Springer, Heidelberg, 2007.
[12] E. Biham and O. Dunkelman, "Differential cryptanalysis in stream ciphers," Cryptology ePrint Archive, Report 2007/218, IACR, 2007, http://eprint.iacr.org/2007/218
[13] S. Lucks, "A failure-friendly design principle for hash functions," Advances in Cryptology — ASIACRYPT 2005, LNCS, vol.3788, pp.19–35, Springer, Heidelberg, 2005.
[14] D. Chang, K.C. Gupta, and M. Nandi, "RC4-Hash: A new hash function based on RC4," Progress in Cryptology — INDOCRYPT 2006, LNCS, vol.4329, pp.80–94, Springer, Heidelberg, 2006.
[15] H. Finney, An RC4 cycle that can't happen, Newsgroup post in sci.crypt, Sept. 1994.
[16] S. Indesteege and B. Preneel, "Collision for RC4-Hash," 11th International Conference on Information Security. LNCS, vol.5222, pp.355–366, Springer, Heidelberg, 2008.

Appendix A: Probability Proof

Lemma 1: Event A: $S_{d+pk}[d + pk] + t = S_{d+pk}[d + pk + t]$ for $p = 0, 1, \cdots, n - 1$
The probability of Event A is

$$P(A_{d,k,p}) = \frac{255}{256}\times(\frac{254}{256})^{d+pk-1}+\frac{1}{256} \text{ for } p = 0, \cdots, n-1$$

Proof 1: There are two cases that could lead to $S_{d+pk}[d + pk] + t = S_{d+pk}[d + pk + t]$.
Case 1(Event B): $S_{d+pk}[d+pk]$ and $S_{d+pk}[d+pk+t]$ have not been swapped before. The probability for this case(Event B) is $(\frac{254}{256})^{d+pk-1}$.
Case 2(Event C): $S_{d+pk}[d + pk]$ and $S_{d+pk}[d + pk + t]$ have been touched before i touches $d + pk$. The probability of Event C is the complement of Event B, namely, $1 - (\frac{254}{256})^{d+pk-1}$.
According to the law of total probability, we have

$$\begin{aligned} P(A) &= P(A|B)P(B) + P(A|C)P(C) \\ &= 1 \times (\tfrac{254}{256})^{d+pk-1} + \tfrac{1}{256} \times (1 - (\tfrac{254}{256})^{d+pk-1}) \\ &= \tfrac{255}{256} \times (\tfrac{254}{256})^{d+pk-1} + \tfrac{1}{256} \end{aligned}$$

Appendix B: RC4 Colliding Key Pairs

Transitional Pattern, h=3, k=128. $K_1(K_2)$

47 **B9(BA)** 01 3F **C0(C1)** CE A1 84 **72(73)** 0C 45 13 A0 7D 2C 4E 1A 77 3B 12 C8 DD 82 D7 9D D0 CD D2 A5 60 63 2C 44 11 92 A1 E3 BC 7B DA AC 9A 64 63 5C CD EB 4E B3 08 05 01 8E 73 1F F5 97 AA 8C 8C 68 C6 80 BD 91 A3 2A B2 71 DF 87 15 F3 EC 5A 8D 46 4E 60 08 C8 08 A1 7B 70 39 BE E0 B3 C4 29 57 18 69 E7 29 54 0C 8B 6B 52 E4 82 17 94 26 C4 03 EE A4 02 E9 16 29 B6 82 C9 5F D3 8C 0B F8 BD 06 6D 1B 5C 01

Self-Absorbing Pattern 1, h=4, k=128. $K_1(K_2)$

29 **D7(D9) 3C(3A) C5(C7)** 4E A3 5E 9F **FD(FF)** 4C 54 E4 AE 9F D6 56 34 92 18 EB 82 62 5B 75 17 2C 9B 37 88 2E B6 4C 37 C8 14 19 AB 3B B8 F0 06 B2 AD 1D 21 7E 31 97 C8 B9 DA DB 3C BC 0E 31 33 D7 7B 3A 1A DE 1A 60 B1 0E 0D AF 09 5A 6A B3 39 B7 67 B7 37 33 28 A3 C1 5D BB 97 D1 91 2A 0A 46 A6 B3 88 A6 CE 99 15 64 F1 E2 78 A5 4A 9F 7D 12 0E 4D 97 4F 81 C9 13 17 6D 4B 0E 1D 60 76 57 4B E1 1F 1C F8 7E A1 94

Self-Absorbing Pattern 2, h=5, k=128. $K_1(K_2)$

DE **22(24) 62(60) 9D(9F)** AE **4B(49) E7(EB)** 09 DD 9A 87 D7 AF A6 1B 3A 5B E2 FC E1 07 A4 7C C6 41 84 DE 84 CD B8 C4 15 56 29 7C 79 73 8A 6C 02 1A 89 37 E0 2E 5C 6D 3F 0F 9C 68 90 65 03 29 E0 62 0F B9 C6 98 E2 94 6F 02 88 23 45 9F D3 FA 2F 82 28 C8 13 61 CD FA E2 22 F3 2D 78 56 AF 34 9D 91 D6 8A 6B B6 32 F7 14 79 14 90 28 AC EC 96 4D C4 C8 9E C6 2C CE 49 5A A9 40 98 01 52 A3 C0 EB F6 18 79 B9 EA 9E 30 C8

Appendix C: RC4-Hash (r_i Functions and S^{IV})

r1: 00 37 2E 25 1C 13 0A 01 38 2F 26 1D 14 0B 02 39 30 27 1E 15 0C 03 3A 31 28 1F 16 0D 04 3B 32 29 20 17 0E 05 3C 33 2A 21 18 0F 06 3D 34 2B 22 19 10 07 3E 35 2C 23 1A 11 08 3F 36 2D 24 **1B** 12 09

r2: 00 39 32 2B 24 1D 16 0F 08 01 3A 33 2C 25 1E 17 10 09 02 3B 34 2D 26 1F 18 11 0A 03 3C 35 2E 27 20 19 12 0B 04 3D 36 2F 28 21 1A 13 0C 05 3E 37 30 29 22 **1B** 14 0D 06 3F 38 31 2A 23 1C 15 0E 07

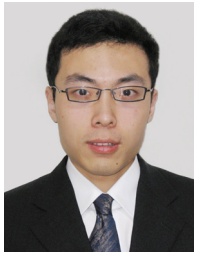
r3: 00 2F 1E 0D 3C 2B 1A 09 38 27 16 05 34 23 12 01 30 1F 0E 3D 2C **1B** 0A 39 28 17 06 35 24 13 02 31 20 0F 3E 2D 1C 0B 3A 29 18 07 36 25 14 03 32 21 10 3F 2E 1D 0C 3B 2A 19 08 37 26 15 04 33 22 11

S^{IV}:
91 39 85 21 41 31 53 3D 71 AB 3F 9B 4A 32 84 F8 EC DA C0 D9 17 24 4F 48 **35** D2 26 3B **36** D0 B9 0C E9 BD 9F A9 F0 9C B8 C8 D1 AD 14 FC 60 D3 8F 65 2C DF 76 01 E8 23 EF 09 72 6D A1 B7 58 42 DB 4E 9D AE BB C1 C7 63 34 78 59 A6 12 4C F1 0D E1 06 92 97 CF B1 67 2D 94 20 1D EA 07 10 13 5B 6C BA 74 3E CB 9E B4 95 43 69 F7 03 80 D7 79 7F B3 AF FB 68 F6 62 8C 0B 86 DD 18 45 BE 9A FD A8 44 E6 3A 99 BC E0 64 81 7C A2 0F 75 E7 96 ED 40 16 98 A5 EB E3 8B C9 54 D5 4D 50 C5 FA 7E CA 27 00 5E 2A F3 E4 57 52 1B 8D 3C A0 2E 7D 70 B5 F2 A7 5C C6 AC AA 37 73 1E 6B 11 38 1F 87 E5 28 6F 25 DE B6 19 2B 77 F4 BF 7A 66 15 5D 61 83 A4 0A 82 2F B0 EE D4 90 29 0E F9 DC 22 88 47 30 8E 49 7B CC CE 04 D8 C4 D6 89 FF C3 1A 08 33 B2 02 8A FE 5A C2 51 F5 6A 5F 4B 56 A3 CD 46 E2 1C 93 55 05 6E

Appendix D: Fast Searching Technique for Self-Absorbing Pattern

The special property of Self-Absorbing pattern allows us to make an efficient colliding key search under this pattern. In order to explain in a simply way, let's assume we want to search for a colliding key pair with parameters d, t and $h = 3$ (it can be applied to general case). In the first round, we have to meet $j_d = d, j_{d+t} = d + t$ which gives us the equation $t = \sum_{i=d+1}^{d+t} K_1[i] + \sum_{i=d+1}^{d+t} S_i[i]$. And also we have to satisfy $S_d[d + t] - S_d[d] = t$. Due to the property of Self-Absorbing pattern, this works exactly the same way for the rest of the rounds. In other words, $\sum_{i=d+1}^{d+t} S_i[i] = \cdots = \sum_{i=d+1+nk}^{d+t+nk} S_i[i] = t - \sum_{i=d+1}^{d+t} K_1[i]$, and $S_d[d+t]-S_d[d] = \cdots = S_{d+nk}[d+t+nk]-S_{d+nk}[d+nk] = t$. Then we can satisfy these conditions in the first round by swapping the right values to the corresponding locations, and hope they will not be touched by j before i touches them. By using this technique, a 39-byte colliding key pair with $h = 3, d = 22, t = 2$ is found within 5 seconds on an Intel Core Duo CPU notebook PC.

$K_1(K_2)$: C2 30 B3 54 07 D8 A5 D4 DF 25 C7 5B 1B 59 27 2F C9 75 77 B8 C5 5E **4F(51) C2(C0)** 11 **0C(0E)** 0D C0 0B 08 09 BC 07 04 D2 EB E1 C8 D1



Jiageng Chen received the B.Sc. in computer science from Huazhong University of Science and Technology (HUST), Wuhan, China in 2004, and the M.Sc. in information science from Japan Advanced Institute of Science and Technology (JAIST), Nomi, Japan in 2007. He is now a Ph.D. candidate of School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He is supported by the Graduate Research Program (GRP) within the School of Information Science at JAIST. His research areas mainly include cryptanalysis on symmetric key cryptography, especially on stream ciphers.

Atsuko Miyaji received the B.Sc., the M.Sc., and the Dr. Sci. degrees in mathematics from Osaka University, Osaka, Japan in 1988, 1990, and 1997 respectively. She joined Panasonic Co., LTD from 1990 to 1998 and engaged in research and development for secure communication. She was an associate professor at the Japan Advanced Institute of Science and Technology (JAIST) in 1998. She has joined the computer science department of the University of California, Davis since 2002. She has been a professor at the Japan Advanced Institute of Science and Technology (JAIST) since 2007 and the director of Library of JAIST since 2008. Her research interests include the application of number theory into cryptography and information security. She received Young Paper Award of SCIS'93 in 1993, Notable Invention Award of the Science and Technology Agency in 1997, the IPSJ Sakai Special Researcher Award in 2002, the Standardization Contribution Award in 2003, Engineering Sciences Society: Certificate of Appreciation in 2005, the AWARD for the contribution to CULTURE of SECURITY in 2007, IPSJ/ITSCJ Project Editor Award in 2007, 2008, 2009, and 2010, the Director-General of Industrial Science and Technology Policy and Environment Bureau Award in 2007, Editorial Committee of Engineering Sciences Society: Certificate of Appreciation in 2007, DoCoMo Mobile Science Awards in 2008, Advanced Data Mining and Applications (ADMA 2010) Best Paper Award, and The chief of air staff: Letter of Appreciation Award. She is a member of the International Association for Cryptologic Research, the Institute of Electronics, Information and Communication Engineers, the Information Processing Society of Japan, and the Mathematical Society of Japan.