Cryptanalysis of Hummingbird-2

Kai Zhang, *Lin Ding and Jie Guan*

(*Zhengzhou Information Science and Technology Institute*, *Zhengzhou* 450000, *China*)

Abstract: Hummingbird is a lightweight encryption and message authentication primitive published in RISC'09 and WLC'10. In FSE'11, Markku-Juhani O.Saarinen presented a differential divide-and-conquer method which has complexity upper bounded by 2^{64} operations and requires processing of few megabytes of chosen messages under two related nonces (*IV*s). The improved version, Hummingbird-2, was presented in RFIDSec 2011. Based on the idea of differential collision, this paper discovers some weaknesses of the round function *WD*16 combining with key loading algorithm and we propose a related-key chosen-*IV* attack which can recover the full secret key. Under 24 pairs of related keys, the 128 bit initial key can be recovered, with the computational complexity of $O(2^{32.6})$ and data complexity of $O(2^{32.6})$. The result shows that the Hummingbird-2 cipher can't resist related key attack.

Key Words: Cryptanalysis; Hummingbird-2; Related Key Attack; Lightweight Cipher; Hybrid Cipher

1 Introduction

Symmetric encryption algorithms are traditionally categorized into two types of schemes: block ciphers and stream ciphers. Stream ciphers distinguish themselves from block ciphers by the fact that they process plaintext symbols (typically bits) as soon as they arrive by applying a very simple but ever changing invertible transformation, it's based on the idea of "*One Time Pad Assumption*". As for block ciphers, their security are from the complexity of the encryption transformation, it's based on the theory of "*Confusion and Diffusion*". Nowadays, people try to combine the stream cipher and the block cipher together to make safer ciphers, such as *CSA*[4], Hummingbird family ciphers[1,2,3], etc.

Hummingbird-1 is a recent cryptographic algorithm proposal for RFID tags and other constrained devices. It is covered by several pending patents and is being commercially marketed by the Revere Security. Revere has invested into Hummingbird's cryptographic security assurance before its publication by contracting ISSI, a private consultancy employing some ex-NSA staff and members of U.Waterloo CACR. In FSE 2011, Markku-Juhani O. Saarinen proposed a differential divide-and-conquer method which has complexity upper bounded by 2^{64} operations and requires processing of few megabytes of chosen messages under two related nonces (*IV*s). In RFIDSec 2011, the improved version, Hummingbird-2, was presented. It is also an encryption and message authentication primitive that has been designed particularly for resource-constrained devices such as RFID tags, wireless sensors, smart meters and industrial controllers. For Hummingbird-2, Xinxin Fan and Guang Gong proposed a side channel cube attack which can recover the first 48 bit initial key for the data complexity of $O(2^{18})$. There are no other cryptanalytic results on Hummingbird-2 up to now.

Related key cryptanalysis is first introduced by Biham and independently by Knudsen in 1993[7,8], it is a type of chosen-key attacks, in which the relationship between the keys used is

* This research is supported by the Science and Technology on Communication Security Laboratory Foundation.

known. People try to get the information of the initial key by analyzing the ciphertexts under certain related keys. Combined with differential attack, Kelsey proposed Related Key differential cryptanalysis in Ref.[9], and it is also combined with impossible differential attack and high order differential attack.

In the specification of Hummingbird-2, the author referred to a related key differential characteristic, but didn't make an attack. In the present report we show that the published version of Hummingbird-2 is suspectible to a related-key chosen-*IV* attack that under 24 pairs of related keys, the 128 bit initial key can be recovered with the computational complexity of $O(2^{32.6})$ and data complexity of $O(2^{32.6})$.

This paper is structed as follows. In Section 2 we give a description of Hummingbird-2. In Section 3 we present a key observation about the initialization and encryption procedure of algorithm, then we propose an attack that recover the key, furthermore we make an improvement of the attack, followed by conclusions in Section 4.

2 Description of Hummingbird-2

The Hummingbird-2 cipher has a 128-bit secret key K and a 128-bit internal state R which is initialized using a 64-bit Initialization Vector (i.e. IV). The key, registers and IV are denoted as follows:

$$K = (K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8)$$
$$R = (R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8)$$
$$IV = (IV_1, IV_2, IV_3, IV_4)$$

The nonlinear function $f(x)$ and $WD16(x,a,b,c,d)$ are expressed as

$$x=(x_3,x_2,x_1,x_0)$$
$$S(x) = S_1(x_0) \mid S_2(x_1) \mid S_3(x_2) \mid S_4(x_3)$$
$$L(x) = x \oplus (x <<< 6) \oplus (x <<< 10)$$
$$f(x) = L(S(x))$$
$$WD16(x,a,b,c,d) = f(f(f(f(x \oplus a) \oplus b) \oplus c) \oplus d)$$

The S-Boxes S_1,S_2,S_3 and S_4 are given in Table 1 below.

Table 1 The *S*-Boxes of Hummingbird-2

X	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S_1(x)$	7	c	e	9	2	1	5	f	b	6	d	0	4	8	a	3
$S_2(x)$	4	a	1	6	8	f	7	c	3	0	e	d	5	0	b	2
$S_3(x)$	2	f	c	1	5	6	a	d	e	8	3	4	0	b	9	7
$S_4(x)$	f	4	5	8	9	7	2	1	a	3	0	e	6	c	d	b

(1) The Initialization Process

First of all, the initial state of the registers $R^{(0)}$ are filled with IV as follows:

$$R^{(0)}=(R_1^{(0)}, R_2^{(0)}, R_3^{(0)}, R_4^{(0)}, R_5^{(0)}, R_6^{(0)}, R_7^{(0)}, R_8^{(0)}) =(IV_1, IV_2, IV_3, IV_4, IV_1, IV_2, IV_3, IV_4)$$

Then iterate for i=0,1,2,3 as follows:

$$t_1 = WD16(R_1^{(i)} \boxplus <i>, K_1, K_2, K_3, K_4)$$

($<i>$represents the binary expansion of i, "⊞" represents "addition module 2^{16}")

$$t_2 = WD16(R_2^{(i)} \boxplus t_1, K_5, K_6, K_7, K_8)$$
$$t_3 = WD16(R_3^{(i)} \boxplus t_2, K_1, K_2, K_3, K_4)$$

$$t_4 = WD16(R_4^{(i)} \boxplus t_3, K_5, K_6, K_7, K_8)$$
$$R_1^{(i+1)} = (R_1^{(i)} \boxplus t_4) <<< 3$$
$$R_2^{(i+1)} = (R_2^{(i)} \boxplus t_1) >>> 1$$
$$R_3^{(i+1)} = (R_3^{(i)} \boxplus t_2) <<< 8$$
$$R_4^{(i+1)} = (R_4^{(i)} \boxplus t_3) <<< 1$$
$$R_5^{(i+1)} = R_5^{(i)} \oplus R_1^{(i+1)}$$
$$R_6^{(i+1)} = R_6^{(i)} \oplus R_2^{(i+1)}$$
$$R_7^{(i+1)} = R_7^{(i)} \oplus R_3^{(i+1)}$$
$$R_8^{(i+1)} = R_8^{(i)} \oplus R_4^{(i+1)}$$

The initial state of registers for encrypting the first plaintext word is $R^{(4)}$.

(2) The Encryption Process

The encryption of the ith plaintext P_i to C_i need four iteration of $WD16$ as follows:

$$t_1 = WD16(R_1^{(i)} \boxplus P_i, K_1, K_2, K_3, K_4)$$
$$t_2 = WD16(R_2^{(i)} \boxplus t_1, K_5 \oplus R_5^{(i)}, K_6 \oplus R_6^{(i)}, K_7 \oplus R_7^{(i)}, K_8 \oplus R_8^{(i)})$$
$$t_3 = WD16(R_3^{(i)} \boxplus t_2, K_1 \oplus R_5^{(i)}, K_2 \oplus R_6^{(i)}, K_3 \oplus R_7^{(i)}, K_4 \oplus R_8^{(i)})$$
$$C_i = WD16(R_4^{(i)} \boxplus t_3, K_5, K_6, K_7, K_8) \boxplus R_1^{(i)}$$

The registers R_1 to R_8 are refreshed as follows:

$$R_1^{(i+1)} = R_1^{(i)} \boxplus t_3$$
$$R_2^{(i+1)} = R_2^{(i)} \boxplus t_1$$
$$R_3^{(i+1)} = R_3^{(i)} \boxplus t_2$$
$$R_4^{(i+1)} = R_4^{(i)} \boxplus R_1^{(i)} \boxplus t_3 \boxplus t_1$$
$$R_5^{(i+1)} = R_5^{(i)} \oplus (R_1^{(i)} \boxplus t_3)$$
$$R_6^{(i+1)} = R_6^{(i)} \oplus (R_2^{(i)} \boxplus t_1)$$
$$R_7^{(i+1)} = R_7^{(i)} \oplus (R_3^{(i)} \boxplus t_2)$$
$$R_8^{(i+1)} = R_8^{(i)} \oplus (R_4^{(i)} \boxplus R_1^{(i)} \boxplus t_3 \boxplus t_1)$$

3 Cryptanalysis of Hummingbird-2

Our representation obtains a series of differential characteristics based on the thought of related key attack and differential collision through the initialization and the encryption process of the algorithm. First we construct certain partial differentials within the round function $WD16$ by choosing proper related keys, then we detect whether the differential pairs we built has occurred by examining the difference of the ciphertexts. If the differential pairs occurred, we can use the differential cryptanalysis techniques to recover the key.

3.1 Differential Properties of *S*-Boxes on Hummingbird-2

First of all, introduce some concepts of differential cryptanalysis.

Definition 1[14] A differential of a function f: $F_2^n \to F_2^n$ is a pair $(\alpha, \beta) \in F_2^n \times F_2^n$ such that $f(x+\alpha)=f(x)+\beta$ for some $x \in F_2^n$. We call α the input difference and β the output difference. The differential probability $p_f(\alpha \to \beta)$ of a differential (α, β) with respect to $f(x)$ is defined as

$$p_f(\alpha \to \beta) = p\{(x_1, x_2) \in F_2^n \times F_2^n : f(x_1) - f(x_2) = \beta \mid x_1 - x_2 = \alpha\}$$

Through analyzing the four S-Boxes of Hummingbird-2, we study the distribution of the probability of differentials, and get various differential pairs with different differential probability. As for our attack, we only use the highest differential probability which is 1/4 for all of the four S-Boxes, so we only illustrate these differential pairs in Table 2.(In the table 2, $\alpha\rightarrow\beta$ represents the input difference and output difference respectively.)

Table 2 Highest differential pairs of four S-Boxes of Hummingbird-2

S box	Highest probability differential pairs
S_1	$1\rightarrow d,2\rightarrow 6,2\rightarrow e,3\rightarrow 2,3\rightarrow b,5\rightarrow e,6\rightarrow 8,7\rightarrow 8,8\rightarrow 9,$ $8\rightarrow c,9\rightarrow 5,b\rightarrow 1,b\rightarrow b,c\rightarrow 4,e\rightarrow 1,e\rightarrow f,f\rightarrow 4,f\rightarrow 7$
S_2	$1\rightarrow 3,1\rightarrow 7,2\rightarrow d,3\rightarrow 2,3\rightarrow e,4\rightarrow 5,4\rightarrow 6,6\rightarrow 9,7\rightarrow 8,$ $7\rightarrow e,a\rightarrow 2,b\rightarrow 4,b\rightarrow 9,c\rightarrow 1,d\rightarrow d,e\rightarrow 4,e\rightarrow f,f\rightarrow 1$
S_3	$1\rightarrow 7,1\rightarrow d,2\rightarrow c,2\rightarrow e,3\rightarrow 3,4\rightarrow 3,5\rightarrow 4,6\rightarrow 7,6\rightarrow f,$ $7\rightarrow 4,8\rightarrow 5,a\rightarrow 1,b\rightarrow f,c\rightarrow 9,d\rightarrow 8,d\rightarrow e,f\rightarrow 1,f\rightarrow 5$
S_4	$1\rightarrow e,2\rightarrow a,2\rightarrow b,3\rightarrow 1,7\rightarrow 1,7\rightarrow e,8\rightarrow 5,8\rightarrow f,9\rightarrow c,$ $a\rightarrow 4,a\rightarrow f,b\rightarrow 2,c\rightarrow 3,c\rightarrow 8,e\rightarrow 2,e\rightarrow 9,f\rightarrow 7,f\rightarrow 9$

Next, we can recover the key blocks using the high probability differential pairs above with the differential properties of the algorithm in the next Section 3.2.

3.2 Differential Properties of Hummingbird-2

The round function $WD16$ can be expressed in the Figure 1 below:

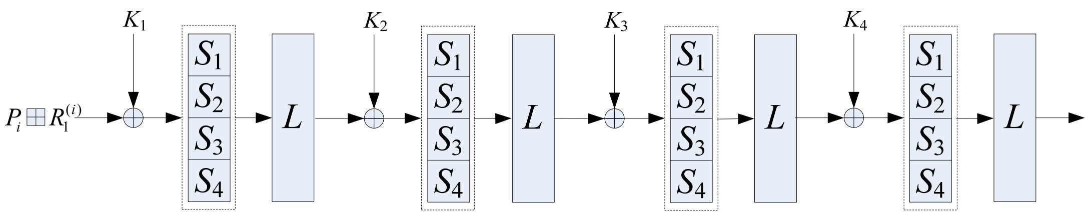

Figure 1 Round function $WD16$

In this chapter, we first deal with the differential characteristic of the $WD16$ function and then step by step we analyze the differential characteristic of the algorithm.

The round function $WD16$ can be viewed as a small "block cipher". To minimize the probability of differential over round function $WD16$, it's number of active S-Boxes must be minimized. As the algorithm consists of 4 round functions, and for each block of subkey it is used twice, on the same location of first round and the third round or the second round and the fourth round. So if we introduce a difference on the subkey of the first round or the second round which causes an active S-Box, at the same position on the third round or the fourth round must emerge an active S-Box. That is to say, the number of the active boxes is gemination, at least 2.

Note the 16 bit input of the 4 S-Boxes is Y=(y_{15},y_{14},y_{13},y_{12},y_{11}…,y_0), y_{15} is the most significant bit and the y_0 is the least significant bit, input of the four S-Boxes S_1,S_2,S_3,S_4 are (y_3,y_2,y_1,y_0), (y_7,y_6, y_5,y_4), (y_{11},y_{10},y_9,y_8),(y_{15},y_{14},y_{13},y_{12}) respectively. Remark the 16 bit subkey K_i as ($K_i[3]$,$K_i[2]$,$K_i[1]$, $K_i[0]$).

We take $\Delta K_1 = K_1 \oplus K_1'$=($\Delta K_1[3]$,0000,0000,0000), ($\Delta K_1[3] \neq 0000$) as an example:

S_4 is the only active S-Box of all the S-Boxes, for S_4 ,$\Delta K_1 \rightarrow \Delta Z$ is one of the highest differential probability pairs with the differential probability of p, if we choose related keys with

$\Delta K_2 = L(\Delta Z)$, $\Delta K_3, \cdots, \Delta K_8$ are all zero, it is obvious that for each round function *WD*16, the probability for input difference and the output difference are both zero is *p*. Furthermore, each encryption process (or initialization process) consists of 4 round function *WD*16, according to the algorithm, at the same position of the third round the differential pair $\Delta K_1 \to \Delta Z$ also exists, so if the difference of the plaintext block is zero, under the related keys above, the difference of the ciphertext is also zero with the probability p^2.

We take ΔK_1 =$(3000)_{16}$ as an example, the initialization and the encryption process of the algorithm have the properties below:

Property 1 Differential characteristic of the initialization for each round: Under two related keys $\Delta K = (\Delta K_1, \Delta K_2, \Delta K_3, \Delta K_4, \Delta K_5, \Delta K_6, \Delta K_7, \Delta K_8) = (3000, 0441, 0000, 0000, 0000, 0000, 0000, 0000)_{16}$, the differential characteristic below pass each round of initialization for the probability of $1/2^4$:

$$\Delta(IV_1, IV_2, IV_3, IV_4) = (0000, 0000, 0000, 0000)$$

$$\downarrow$$

$$\Delta(R1_{-3}, R2_{-3}, R3_{-3}, R4_{-3}, R5_{-3}, R6_{-3}, R7_{-3}, R8_{-3}) = (0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000)$$

If we find some *IV* which make the differential characteristic above occurs, we can use the differential pair $0x3 \rightarrow 0x1$ of the S_4 to recover the input, i.e. $IV_1 \oplus K_1[3]$, as IV_1 is known, then we can recover the subkey block $K_1[3]$ easily.

Property 2 Differential characteristic of the whole initialization process: Under two related keys $\Delta K = (\Delta K_1, \Delta K_2, \Delta K_3, \Delta K_4, \Delta K_5, \Delta K_6, \Delta K_7, \Delta K_8) = (3000, 0441, 0000, 0000, 0000, 0000, 0000, 0000)_{16}$, the differential characteristic below pass the whole initialization process for the probability of $1/2^{16}$:

$$\Delta(IV_1, IV_2, IV_3, IV_4) = (0000, 0000, 0000, 0000)$$

$$\downarrow$$

$$\Delta(R1_0, R2_0, R3_0, R4_0, R5_0, R6_0, R7_0, R8_0) = (0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000)$$

For the initialization process are totally 4 round, so the characteristic in property 1 can hold through the whole initialization process with the probability of $1/2^{16}$.

Property 3 An iterated differential characteristic during the encryption process: Under the related keys in the property 2, the differential characteristic below pass each encryption process for the probability of $1/2^4$:

$$\Delta(P_i, R1_0, R2_0, R3_0, R4_0, R5_0, R6_0, R7_0, R8_0) = (0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000)$$

$$\downarrow$$

$$\Delta(C_i, R1_1, R2_1, R3_1, R4_1, R5_1, R6_1, R7_1, R8_1) = (0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000)$$

The property 3 denote that if the difference of the plaintext is (0000), based on the situation of property 2, the difference of the ciphertext is (0000) for the probability of $1/2^4$.

As for the several properties above, under the conditions of related keys, if the *IV* difference and the plaintext(*P*) difference are both zero, when we change the value of *IV*, we can always find such values which can satisfy the three properties.

3.3 Key Recovery Attack on Hummingbird-2

In this section, we introduce the key recovery attack algorithm on Hummingbird-2. Here is the clue of the attack: Firstly, we construct differentials through related keys, then we use different IVs to run the initialization process and the encryption process of the algorithm until we find a proper IV which satisfy the three properties in the section 3.2, whether a IV satisfy these properties can be shown through the output difference. If we find a proper IV, it means that the differential pair we constructed has occurred and we can get the input of the active S-Box for the first round of the initialization process, then the subkey can be calculated. Subkeys $K_1,\cdots,K_7$ can be recovered through this process gradually and K_8 can be recovered by exhaustive search.

Next, we take the recovery process of the four significant bits of subkey K_1, ie. $K_1[3]$ as an example to introduce the procedure of the key recover.

Algorithm 1 The key recovery algorithm

Phase1. Encrypt using related keys K and $K \oplus \Delta K$, changing IV until we find a IV which make $C_0 = C_0', C_1 = C_1'$;

(Remark: P_0, P_1 can be any value but the difference ΔP_0, ΔP_1 must be zero)

Phase2. As the input difference and the output difference of S_4 is $0x3 \rightarrow 0x1$, searching S-Box distribution of the probability of differentials we can recover $IV_1 \oplus K_1[3]$, then we can get a $K_1[3]$ candidate set because IV_1 is known and the correct $K_1[3]$ must be within;

Phase3. Make the intersection of the candidate sets, if the number of the candidate set is bigger than one, goto Phase1, find new candidate set and make the intersection; Else if the number of the candidate set is equal to zero, clear the candidate set and goto Phase1; Otherwise return the unique $K_1[3]$ and finish the algorithm.

Using the algorithm above we can always get the right value of $K_1[3]$, the rest 12 bits $K_1[0]$, $K_1[1]$ and $K_1[2]$ can be recovered in the same way.

Similarly, through using different related keys and known K_1, we can use the same technique to recover K_2, under the condition of known K_1 and K_2 we can recover K_3, etc. Then we can recover K_2,K_3,K_4,K_5,K_6,K_7 in turn.

The related keys we constructed to recover all of the key blocks are shown in Table 3 below:

Table 3 Related Keys needed to recover different key blocks

The key blocks to be recovered	The high probability differential pairs used	The constructed related key ΔK
$K_1[0]$	$3\rightarrow 2$	$(0003, 2088, 0000, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_1[1]$	$b\rightarrow 4$	$(00b0, 0411, 0000, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_1[2]$	$d\rightarrow 8$	$(0d00, 2082, 0000, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_1[3]$	$3\rightarrow 1$	$(3000, 0441, 0000, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_2[0]$	$3\rightarrow 2$	$(0000, 0003, 2088, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_2[1]$	$b\rightarrow 4$	$(0000, 00b0, 0411, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_2[2]$	$d\rightarrow 8$	$(0000, 0d00, 2082, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_2[3]$	$3\rightarrow 1$	$(0000, 3000, 0441, 0000, 0000, 0000, 0000, 0000)_{16}$
$K_3[0]$	$3\rightarrow 2$	$(0000, 0000, 0003, 2088, 0000, 0000, 0000, 0000)_{16}$
$K_3[1]$	$b\rightarrow 4$	$(0000, 0000, 00b0, 0411, 0000, 0000, 0000, 0000)_{16}$
$K_3[2]$	$d\rightarrow 8$	$(0000, 0000, 0d00, 2082, 0000, 0000, 0000, 0000)_{16}$
$K_3[3]$	$3\rightarrow 1$	$(0000, 0000, 3000, 0441, 0000, 0000, 0000, 0000)_{16}$
$K_4[0]$	$3\rightarrow 2$	$(0000, 0000, 0000, 0003, 2088, 0000, 0000, 0000)_{16}$
$K_4[1]$	$b\rightarrow 4$	$(0000, 0000, 0000, 00b0, 0411, 0000, 0000, 0000)_{16}$

$K_4[2]$	$d \to 8$	$(0000,0000,0000,0d00,2082,0000,0000,0000)_{16}$
$K_4[3]$	$3 \to 1$	$(0000,0000,0000,3000,0441,0000,0000,0000)_{16}$
$K_5[0]$	$3 \to 2$	$(0000,0000,0000,0000,0003,2088,0000,0000)_{16}$
$K_5[1]$	$b \to 4$	$(0000,0000,0000,0000,00b0,0411,0000,0000)_{16}$
$K_5[2]$	$d \to 8$	$(0000,0000,0000,0000,0d00,2082,0000,0000)_{16}$
$K_5[3]$	$3 \to 1$	$(0000,0000,0000,0000,3000,0441,0000,0000)_{16}$
$K_6[0]$	$3 \to 2$	$(0000,0000,0000,0000,0000,0003,2088,0000)_{16}$
$K_6[1]$	$b \to 4$	$(0000,0000,0000,0000,0000,00b0,0411,0000)_{16}$
$K_6[2]$	$d \to 8$	$(0000,0000,0000,0000,0000,0d00,2082,0000)_{16}$
$K_6[3]$	$3 \to 1$	$(0000,0000,0000,0000,0000,3000,0441,0000)_{16}$
$K_7[0]$	$3 \to 2$	$(0000,0000,0000,0000,0000,0000,0003,2088)_{16}$
$K_7[1]$	$b \to 4$	$(0000,0000,0000,0000,0000,0000,00b0,0411)_{16}$
$K_7[2]$	$d \to 8$	$(0000,0000,0000,0000,0000,0000,0d00,2082)_{16}$
$K_7[3]$	$3 \to 1$	$(0000,0000,0000,0000,0000,0000,3000,0441)_{16}$

Now, we have recovered 112 bits($K_1,\cdots,K_7$) of the K, the last 16 bits K_8 can be recovered by exhaustive search.

3.4 Complexities of the Attack

The precondition of the attack is the occurrence of first two ciphertext difference are zero, first of all, let us consider the probability of the occurrence. According to section 3.2, the probability of $C_0 = C_0', C_1 = C_1'$ is $1/2^{32}$, during the process of choosing different *IV*s, the probability of the occurrence increase with the data size, the relation is shown in Table 4 below:

Table 4 The Relationship between the Data Size and the Ciphertext Collision

Data Size	2^{23}	2^{24}	2^{25}	2^{26}	……
Collision Probability	0.39	0.63	0.86	0.98	……

With the increasing of the data size, the collision probability approaches 1 gradually, that is to say, if the data size is sufficient, the collision will occur. According to the Table above, when the data size reaches $O(2^{26})$, the collision can occur with the probability of 98%, we adopt this result when we calculate the data complexity of our attack, the data complexity to recover the first 4 bits of the key is $O(2^{26})$.

Then we consider the probability to get the unique key if the collision above appears.

If randomly distributed, the probability of $C_i = C_i'(i = 0,1)$ is $1/2^{32}$. Under the condition in section 3.2, the probability is $1/2^{24}$, so if the inside of the encryption process is random whereas the ciphertext difference can pass the collision test, the probability is $2^{-32}/(2^{-32}+2^{-24})=1/(2^8+1)\approx$ 0.4%. Through differential techniques we can get four $K_i[j]$ candidates, when we get four such candidate sets, if the intersection of the sets is zero, it is definitely the random case , then we should discard the candidates and choose new *IV*s to start over; If the differential characteristic satisfy the properties we constructed, the probability of acquire a unique key is 98%(More details see Appendix 1). So in the key recover algorithm, we can determine whether we should add more data to reduce the scale of the candidate set or not. Now, we need about 2^{28} pairs of plaintexts to recover the first 4 bits of the key.

In this way, we can recover the first 112 bits of the key in turn with data complexity of $O(2^{32.8})$ and computational complexity of $O(2^{32.8})$, to recover the subkey block K_8 we need one plaintext block, the computational complexity is $O(2^{16})$. So the data complexity and computational complexity of the attack to recover the 128 bits key are both $O(2^{32.8})$. As we need one related key to recover each four bits of subkeys and each related key is different from each other according to table 3, we need 112/4=28 pairs of related keys totally.

3.5 An Improvement of the Attack

As all the subkeys are recovered by blocks in turn, we can add exhaustive scale to reduce the related keys and the complexity. We listed the relations as below:

Table 5 Relationship of Related Keys, Computational Complexity and Data Complexity under different Exhaustive Scale

Exhaustive Scale(*bit*)	20	24	28	32	36
Related Keys needed(*pairs*)	27	26	25	24	23
Computational Complexity	$O(2^{32.8})$	$O(2^{32.7})$	$O(2^{32.6})$	$O(2^{32.6})$	$O(2^{36})$
Data Complexity	$O(2^{32.8})$	$O(2^{32.7})$	$O(2^{32.6})$	$O(2^{32.6})$	$O(2^{36})$

Through analysis, after our improvement of the attack, the computational complexity can be reduced to $O(2^{32.8})$, the data complexity can be reduced to $O(2^{32.8})$, and the related keys needed can be reduced to 24 pairs at the same time.

4 Conclusion

The designers of Hummingbird-2 claimed that the Hummingbird-2 is resistant to all previously known cryptanalytic attacks, including related key attack. However, in this paper, we present a related-key chosen *IV* attack combining with differential techniques on Hummingbird-2. First of all, using related keys to construct partial differential with probability, ensure the collision with sufficient chosen *IV*s, and judge it by the difference of ciphertext, then use differential techniques to recover the initial key. As the key loading algorithm is too simple, though adding the influence of the registers, these effects can be eliminated by differential techniques, which make the attack possible. Under 24 pairs of related keys, we can recover the 128 bit initial key with computational complexity of $O(2^{32.8})$ and data complexity of $O(2^{32.8})$. Compared with the attack proposed by *Markku-Juhani O. Saarinen*, our attack use the inner differential characteristic of round function *WD*16 rather than the outer differential characteristic. Furthermore, we have proved that the Hummingbird-1 can also be analyzed in the same way. The result in this paper shows that Hummingbird-2 cipher can't resist the related-key attack. The ability of Hummingbird family ciphers to resist other cryptanalysis is further to be studied.

References:

[1] Engels, D., Fan, X., Gong, G., Hu, H., Smith, E.M.: Ultra-Lightweight Cryptography for Low-Cost RFID Tags: Hummingbird Algorithm and Protocol. Centre for Applied Cryptographic Research (CACR) Technical Reports, CACR-2009-29. http://www.cacr.math.uwaterloo.ca/techreports/2009/cacr2009-29.pdf

[2] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith. Hummingbird: Ultra-Lightweight

Cryptography for Resource-Constrained Devices. FC 2010 Workshops, RLCPS, WECSR, and WLC 2010, ser. LNCS6054, R. Curtmola et al. (eds.), Berlin, Germany: Springer-Verlag, pp.3-18, 2010.

[3] D. Engels, M.-J. O. Saarinen, and E. M. Smith. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. In the proceedings of The 7th Workshop on RFID Security and Privacy - RFIDSec 2011, Berlin, Germany: Springer-Verlag, 2011.

[4] Ralf-Phillip Weinmann and Kai Wirt. Analysis of the DVB Common Scrambling Algorithm. IFIP International Federation for Information Processing 2005, Volume 175/2005, pp.195-207, 2005.

[5] M.-J. O. Saarinen. Cryptanalysis of Hummingbird-1. The 18th International Workshop on Fast Software Encryption - FSE 2011, ser. LNCS 6733, A. Joux (ed.), Berlin, Germany: Springer-Verlag, pp.328-341, 2011.

[6] Xinxin Fan and Guang Gong. On the Security of Hummingbird-2 against Side Channel Cube Attacks. In the 2011 West European Workshop on Research in Cryptography- WEWoRC 2011, Weimar, Germany: Springer-Verlag, pp.100-104, 2011.

[7] Biham E. New types of cryptanalytic attacks using related keys[C]. EUROCRYT 1993, LNCS 765. Springer-Verlag, 1994:398-409.

[8] Knudsen L. Cryptanalysis of LOKI[C]. ASIACRYPT 1992, LNCS 739. Springer-Verlag, 1993:22-35.

[9] Keysey J, Schneier B, and Wanger D. Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES[C]. CRYPTO 1996, LNCS 1109. Springer-Verlag, 1996:237-251.

[10] Jakimoski G, Desmedt Y. Related-Key differential cryptanalysis of 192-bit key AES Variants [C]. SAC 2003, LNCS 3006. Springer-Verlag, 2004:208-221.

[11] Biham E, Dunkelman O, Keller N. Related-key impossible differential attacks on 8-round AES-192[C]. CT-RSA 2006, LNCS 3860. Springer-Verlag, 2006:21-33.

[12] Zhang W, Wu W, Zhang L, and Feng D. Improved related-key impossible differential attack on reduced-round AES-192[C]. SAC 2006, LNCS 4356. Springer-Verlag, 2007:15-27.

[13] Biham E, Dunkelman E, Keller O. Related-key boomerang and rectangle attacks[C]. EUROCRYPT 2005[C], LNCS 3494. Springer-Verlag, 2005:507-525.

[14] Biham E, Shamir A. Differential Cryptanalysis of DES-like Cryptosystems. In:Menezes A, Vanstone, S.A. (eds.) CRYPTO 1990. LNCS 537. Springer-Verlag, 1990:2-21.

Appendix 1

Set A, B, C, D and E represent the candidates sets eliminating the correct subkey block $K_i[j]$:

$$
\begin{aligned}
P_4 &= P(A \cap B \cap C \cap D = \varnothing) \\
&= P(|A \cap B \cap C \cap D| = 0) \\
&= \sum_{k=0}^{\min\{|A|,|B|,|C|\}} P(|A \cap B \cap C| = k, |A \cap B \cap C \cap D| = 0) \\
&= \sum_{k=0}^{\min\{|A|,|B|,|C|\}} P(|A \cap B \cap C| = k) \times P(|A \cap B \cap C \cap D| = 0 / |A \cap B \cap C| = k) \\
&= \sum_{k=0}^{\min\{|A|,|B|,|C|\}} \left(\sum_{j=k}^{\min\{|A|,|B|\}} P(|A \cap B| = j) \times P(|A \cap B \cap C| = k / |A \cap B| = j) \right) \\
&\quad \cdot P(|A \cap B \cap C \cap D| = 0 / |A \cap B \cap C| = k) \\
&= \sum_{k=0}^{3} \frac{C_{15}^{k} \cdot C_{15-k}^{3}}{C_{15}^{k} \cdot C_{15}^{3}} \sum_{j=k}^{3} \frac{C_{15}^{3} \cdot C_{3}^{j} \cdot C_{15-3}^{3-j}}{(C_{15}^{3})^2} \cdot \frac{C_{15}^{j} \cdot C_{j}^{k} \cdot C_{15-j}^{3-k}}{C_{15}^{j} \cdot C_{15}^{3}} \\
&\approx 98\%
\end{aligned}
$$