Near-Collision Attack and Collision-Attack on Double Block Length Compression Functions based on the Block Cipher IDEA

Donghoon Chang

Center for Information Security Technologies(CIST),
Korea University, Korea
dhchang@cist.korea.ac.kr

Abstract. IDEA is a block cipher designed by Xuejia Lai and James L. Massey and was first described in 1991. IDEA does not vary the constant in its key schedule. In [1], Donghoon Chang and Moti Yung showed that there may be a weakness of hash function based on block cipher whose key schedule does not use various constants. Based on their result, we investigate the security of double block length compression functions based on the block cipher IDEA such that the key size of IDEA is 128 bits and its block length is 64 bits. We use the double block length hash functions proposed by Shoichi Hirose in the second hash workshop in 2006 [2]. Then, we can easily find a near-collision by hand. And also, for a constant c of DBL hash functions, we can find a collision by hand. This means that the constant c may be used as a trapdoor to make the attacker find collision easily.

Keywords : Hash Function, Double Block Length Compression Function, Near-Collision Attack, Collision Attack, IDEA.

1 Introduction.

At the second hash workshop in 2006, Shoichi Hirose [2] proposed several double block length (DBL) compression functions which are collision resistant in the ideal cipher model. But, when we instantiate a block cipher, we can not say the security of the DBL compression functions. Mario Lamberger, Norbert Pramstaller and Vincent Rijmen [3] showed that when we use DESX which is a block cipher based on DES, some DBL hash functions based on DESX is not secure against the second preimage attack. Their result shows that we have to be careful to design the block cipher based on a block cipher. their attack does not use a weakness of DES but the weakness of structure of DESX based on DES. On the other hand, we focus on the underlying block cipher. By analyzing the block cipher IDEA, we show that the DBL compression function based on IDEA has a weakness. some DBL compression functions based on IDEA is not secure against the near-collision attack. This means that when we design secure hash function based on block cipher, we have to vary the constants in the key schedule of the block cipher. This results support the result of [1]. And also, for a constant c of DBL hash functions, we can find a collision by hand. This means that the constant c may be used as a trapdoor to make the attacker find collision easily.

2 IDEA and DBL Compression Functions

2.1 the Block Cipher IDEA

Let the block be 16-bit. $X_1||X_2||X_3||X_4$ is a 64-bit plaintext such that each X_i is 16-bit. IDEA consists of 8 rounds. From the 128-bit master key K, we get (7×128)-bit S as follows.

$$S = K||K^{\lll 25}||K^{\lll 50}||K^{\lll 75}||K^{\lll 100}||K^{\lll 125}||K^{\lll 22}||K^{\lll 47}$$

The first 52 blocks of S are used as subkeys. i-th round subkeys are $Z^i = Z_1^{(i)}||Z_2^{(i)}||\cdots||Z_6^{(i)}$. For example, the first round subkeys are the first 6 blocks of S. $\boxplus$ means the addition modulo 2^{16}. $\odot$ means the multiplication modulo $2^{16}+1$ such that the zero is treated as 2^{16}.

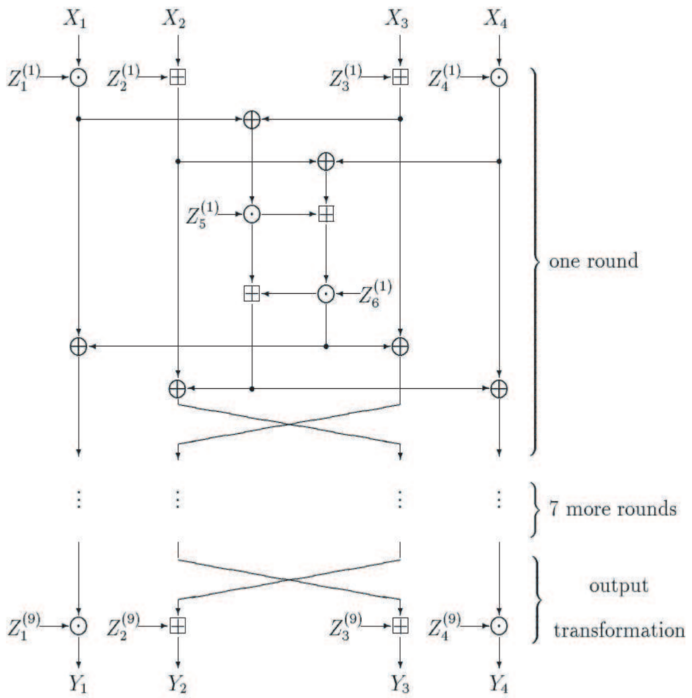

Fig. 1. the Block Cipher IDEA

2.2 DBL Compression Functions

Here, we describe only two DBL compressin functions which are used in our security analysis. e is a block cipher with n-bit block and $2n$-bit key. g_{i-1}, h_{i-1} and m_i are n-bit. c is a non-zero n-bit constant. Shoichi Hirose [2] showed that the the compression functions in Fig. 2 and 3 are collision resistant in the ideal cipher model. In case of IDEA, $n = 64$.

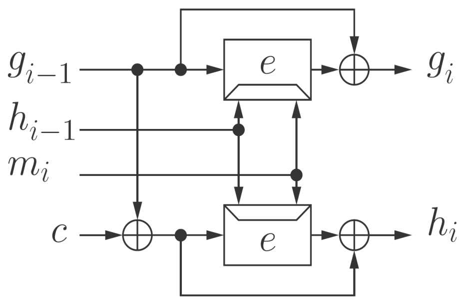

Fig. 2. Double Block Length Compression Function 1

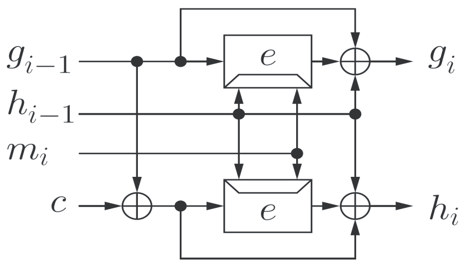

Fig. 3. Double Block Length Compression Function 2

3 Near-Collision Attack on DBL Compression Functions based on IDEA

The attack strategy is similar to that of [1].

Analysis of Key Schedule of IDEA In the key schedule of IDEA, the left rotation is used. We want to choose the 128-bit master key satisfying $Z^1 = Z^2 =$

$\cdots = Z^8$. The candidates of the master key are all zero bits and all one bits. Since IDEA uses $\boxplus$ and $\odot$, we choose all zero bits, i.e. $K = 0^{128}$. Then all subkeys are all zero bits.

Analysis of Round Function of IDEA Let $\mathsf{Round}_i(Z^i||X^i)$ be the i-th round function of IDEA such that Z^i is the 6-block round key and X^i is the 4-block input of round i. Since the master key is 0^{128}, $\mathsf{Round}_i(Z^i||X^i)$=$\mathsf{Round}_i(0^{96}||X^i)$. Then, we want to find X such that $\mathsf{Round}_i(0^{96}||X) = X$. Let $X = a||b||c||d$ such that a, b, c and d are 1-block. In order to find X such that $\mathsf{Round}_i(0^{96}||X) = X$, X have to satisfy conditions (1), (2) and (3) in Fig. 4.

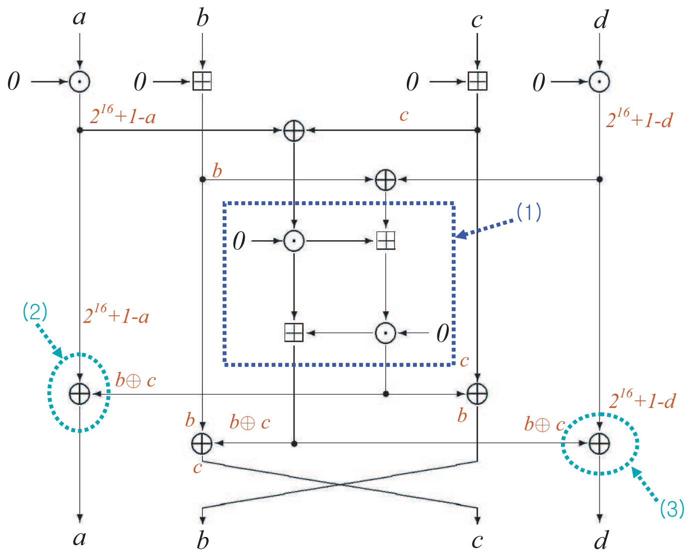

Fig. 4. Analysis of Round Function of IDEA

Condition (2) is that $b \oplus c = (2^{16} + 1 - a) \oplus a$. Condition (3) is $b \oplus c = (2^{16} + 1 - d) \oplus d$. In order to know the condition (1), we look closely like Fig. 5. Since $2^{16} \odot 2^{16} = 1$, $(2^{16})^{-1} = 2^{16}$. In Fig. 5, we have two conditions (1.1) and (1.2). Condition (1.1) is that $2^{16} + 1 - ((2^{16} + 1 - a) \oplus c) + ((2^{16} + 1 - d) \oplus b) = 2^{16} + 1 - (b \oplus c)$. Condition (1.2) is that $2^{16} + 1 - ((2^{16} + 1 - a) \oplus c) = 0$.

Therefore, we want to obtain a, b, c and d satisfying conditions (1.1), (1.2), (2) and (3).

(1.1) $2^{16} + 1 - ((2^{16} + 1 - a) \oplus c) + ((2^{16} + 1 - d) \oplus b) = 2^{16} + 1 - (b \oplus c)$.
(1.2) $2^{16} + 1 - ((2^{16} + 1 - a) \oplus c) = 0$.
(2) $b \oplus c = (2^{16} + 1 - a) \oplus a$.

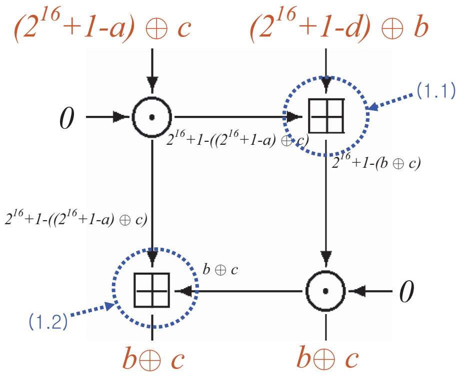

Fig. 5. Analysis of Conditioin (1) Of the Round Function of IDEA

(3) $b \oplus c = (2^{16} + 1 - d) \oplus d$.

It is easy to find a solution that $a = 1, b = 0, c = 1, d = 1$. And also we can know that $\mathsf{Round}_i(0^{96}||T) = T'$ and $\mathsf{Round}_i(0^{96}||T') = T$ where $T = 1||0||1||0$ and $T' = 0||1||0||1$. Since IDEA has 8 rounds, we can know that $e_{0^{128}}(1||0||1||1) = (0||0||1||0)$ and $e_{0^{128}}(1||0||1||0) = (0||0||1||1)$ like the Fig. 6.

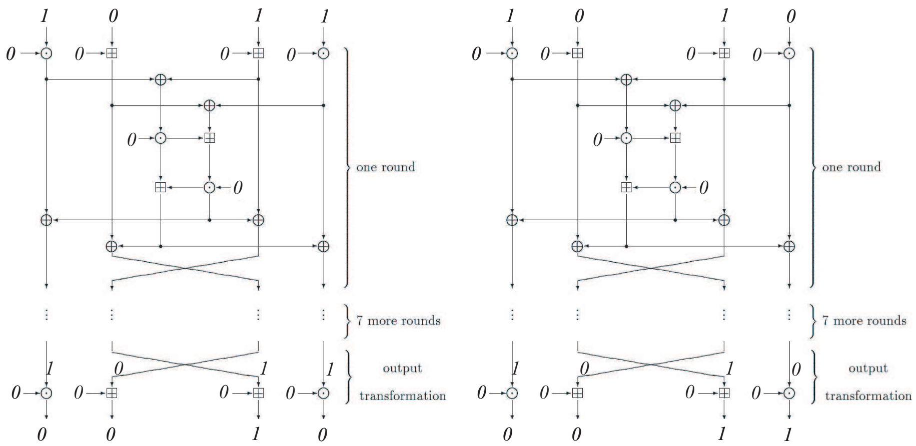

Fig. 6. Two inputs and Two outputs of IDEA

Near-Collision Attack and Collision Attack on DBL Hash Functions based on IDEA We use the previous result such that $e_{0^{128}}(1||0||1||1) = (0||0||1||0)$

and $e_{0^{128}}(1||0||1||0) = (0||0||1||1)$. In Fig. 2 and 3, if $g_{i-1} = (1||0||1||1)$ and $g'_{i-1} = (1||0||1||0)$ and $m_i = m'_i = h_{i-1} = h'_{i-1} = 0^{64}$, then $g_i = g'_i$. If $c = g_{i-1} \oplus g'_{i-1} = (0||0||0||1)$, then $g_i = g'_i$ and $h_i = h'_i$, i.e., we get a collision of DBL compression function based on IDEA. Therefore, c should be chosen carefully so that there is no trapdoor in c.

4 Comment on the result of [1]

We considered two DBL compression functions in Fig. 2 and 3. In [1], they suggested two kinds of modifications of block cipher based hash function. One solution is to change the XOR operation into the addition operation. But, in case of DBL compression function based on IDEA, even though we change the XOR operation into the addition operation in Fig. 2 and 3, the previous results can be applied in the same way. Therefore, based on this result, we can know that to vary the constants of the key schedule of the block cipher is required to guarantee the security.

5 Conclusion

In this paper, based on the result of [1], we showed the near-collision attack and the collision attack on DBL compression functions based on IDEA. In order to instantiate a block cipher in hash function based on the block cipher, we recommend to vary the constants in the key schedule of the block cipher.

References

1. D. Chang and M. Yung, *Do We Need to Vary the Constants? (Methodological Investigation of Block-Cipher Based Hash Functions)*, Cryptology ePrint Archive, Report 2006/467.
2. S. Hirose, *How to Construct Double-Block-Length Hash Functions*, In second Hash Workshop, 2006.
3. M. Lamberger, N. Pramstaller and V. Rijmen, *Second Preimages for Iterated Hash Functions Based on a b-Block Bypass*, Cryptology ePrint Archive, Report 2006/116.