Cryptanalysis of Double-Block-Length Hash Mode MJH

Deukjo Hong and Daesung Kwon

Abstract

A double-block-length (DBL) hash mode of block ciphers, MJH has been proved to be collision-resistant in the ideal cipher model upto $2^{2n/3-\log n}$ queries. In this paper we provide first cryptanalytic results for MJH. We show that a collision attack on MJH has the time complexity below the birthday bound. When block ciphers with 128-bit blocks are used, it has time complexity around 2^{124}, which is to be compared to the birthday attack having complexity 2^{128}. We also give a preimage attack on MJH. It has the time complexity of $2^{3n/2+1}$ with n-bit block ciphers, which is to be compared to the brute force attack having complexity 2^{2n}.

Key Words. MJH, Hash Function, Collision, Preimage

1 Introduction

Block ciphers and hash functions are widely used popular cryptographic primitives. Many hash functions are often designed based on block-cipher-like components. Some of them can be regarded as hash modes of block ciphers. PGV modes [10] are representative single-block-length (SBL) hash modes, where the length of the chaining and hash values is the same as the block length of the underlying block cipher. There are several double-block-length (DBL) hash modes such as MDC-2 , MDC-4 [3, 9], Hirose's scheme [2], Abreast-DM, and Tandem-DM schemes [6], where the length of the chaining and hash value is twice as long as the block length of the underlying block cipher.

MJH is a double-block-length (DBL) hash mode of block ciphers, proposed by Lee and Stam [7] at CT-RSA 2011. The compression function of MJH consists of two block cipher encryptions and one key schedule operation. The designers proved that the security bound of MJH for collision

resistance in the ideal cipher model is about $O(2^{2n/3-\log n})$, and claimed that MJH could be a good alternative to MDC-2 [9, 3] because it has efficiency advantage of one key schedule operation per block compared to MDC-2, and better security bound in the sense of collision resistance. Any proof of preimage resistance and any attacks for MJH have not been reported yet.

We first provide collision and preimage attacks on MJH. Our collision attack uses a multi-collision in the first block, which is the similar approach to [5]. Assuming the cost of one block cipher encryption is equal to one key schedule operation, with $n = 128$, our collision attack on MJH has time complexity less than 2^{124}. We show that a pseudo-preimage can be found for MJH compression function with time complexity of about 2^n. This pseudo-preimage attack can be converted to a preimage attack with time complexity of $2^{3n/2+1}$ by the meet-in-the-middle technique in [8, Fact 9.99] and expandable messages with fixed-points [1, 4]. In fact, a block cipher with n-bit block and k-bit key for $n < k$ can be also used as the underlying primitive for MJH. However, our attacks work for such case, as well. In fact, the designers of MJH considered the combination of a secure double-block-length permutation based on two block cipher encryptions and JH-style domain extender. Our attacks imply that the resulting hash function is much weaker than what the designers expected, since they show that MJH can not reach the security levels which are traditionally expected for cryptographic hash functions, while it is provably secure in the ideal cipher model with about less than $2^{2n/3}$ queries. We think this weakness caused by a cancelation of feedforwards in its compression function.

The remaining parts of this paper are as follows. In Section 2, we give a brief description MJH. In Section 3, we explain collision attacks on MJH compression and hash functions. In Section 4, we explain preimage attacks on MJH compression and hash functions. Finally, we conclude this paper.

2 Description of MJH Hash Modes

MJH has two auxiliary components σ and $\cdot\theta$. σ is an involution on $\{0,1\}^n$ with no fixed point, and $\cdot\theta$ is a multiplication by a constant $\theta \neq 0, 1$ in $\mathbb{F}_{2^n}$. The MJH compression function $\mathrm{CF}^{\mathrm{MJH}}$ has $2n$-bit chaining variable and n-bit message block, based on the block cipher E with n-bit block and n-bit key. For the input chaining variable H and the message block M,

$V = \mathrm{CF}^{\mathrm{MJH}}(H, M)$ is computed as follows:

$$\begin{aligned} X &= H_L \oplus M; \\ V_L &= E_{H_R}(X) \oplus X; \\ Y &= E_{H_R}(\sigma(X)) \oplus \sigma(X); \\ V_R &= (Y \cdot \theta) \oplus H_L. \end{aligned}$$

MJH takes the Merkle-Damgård domain extender to hash arbitrary-length messages with the above compression functions. We assume they use a popular prefix-free padding, which embeds the message length information to the last message block.

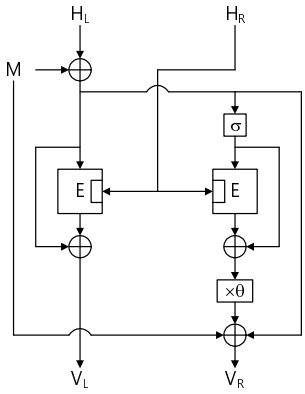

Figure 1: Original figure of $\mathrm{CF}^{\mathrm{MJH}}$

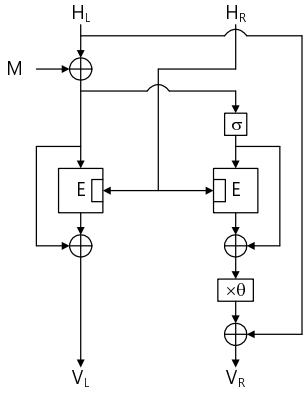

Figure 2: $\mathrm{CF}^{\mathrm{MJH}}(H, M) = V$

Indeed, the designers of MJH considered the combination of a secure double-block-length permutation based on two block cipher encryptions and JH-style domain extender (Fig. 1). However, due to the cancelation between two feedforwards, the original figure is simplified to the equivalent one (Fig. 2). We think that this trivial fact leads to the attacks which the designers unexpected.

In the estimation of time complexities of our attacks, we consider that the compression function of MJH requires two block cipher encryptions and one key schedule operation. Let T_E, T_K, T_{EK}, and T_{CF} be the time costs wasted in one block cipher encryption, one key schedule operation, one block cipher encryption with a key schedule operation, and one compression function operation, respectively. For the evaluation of the time complexity of the attack, we consider the case that T_E is almost equal to T_K (so, $T_{EK} \cong 2T_E$ and $T_{CF} \cong 3 \cdot T_E$), and the case that T_E is much larger than T_K (so,

$T_{EK} \cong T_E$ and $T_{CF} \cong 2 \cdot T_E$). Our complexity evaluations would be valid for most cases because the key schedule is not much more complicated than the block cipher encryption in usual.

3 Collision Attack on MJH

3.1 Collision Attack on MJH Compression Function

We find that we do not need to consider the right half of the hash value in the collision attack for the MJH compression function, because after finding a collision for the left half of the hash value, we can easily compute the left halves of the input chaining variable and the message blocks such that they give a same hash value. The following collision attack on the MJH compression function reflects on our observation.

1. Randomly choose H_R and V_R, and fix them.
2. Randomly choose r distinct $X = H_L \oplus M$: $X^{(1)}, X^{(2)}, ..., X^{(r)}$, compute $V_L^{(i)} = E_{H_R}(X^{(i)}) \oplus X^{(i)}$ for $i = 1, ..., r$, and then check whether there are at least one pair of (i, j) such that
$$X_i \neq X_j \text{ and } V_L^{(i)} = V_L^{(i)}. \quad (1)$$
3. If a pair (i_1, i_2) satisfying 1 is found, then compute $H_L^{(i_1)}$, $H_L^{(i_2)}$, $M^{(i_1)}$, and $M^{(i_2)}$ as follows:
$$\begin{aligned} H_L^{(i_j)} &= (E_{H_R}(\sigma(X^{(i_j)})) \oplus \sigma(X^{(i_j)})) \cdot \theta \oplus V_R \text{ for } j = 0, 1; \\ M^{(i_j)} &= H_L^{(i_j)} \oplus M^{(i_j)} \text{ for } j = 0, 1. \end{aligned}$$
4. Output $(H_L^{(i_1)}, H_R, M^{(i_1)})$ and $(H_L^{(i_2)}, H_R, M^{(i_2)})$ as a collision of the MJH compression function.

The time complexity of the above attack is dominated by r encryptions of the block cipher E. With $r = 2^{n/2}$, we expect at least one collision for MJH compression function, because the right half of the hash value is fixed.

3.2 Collision Attack on MJH Hash Function

We provide a 2-block collision attack on MJH hash function. Let H and V be the $2n$-bit output chaining variables of the first and second blocks,

respectively. The first step to find a collision for MJH hash function is similar to Knudsen et al.'s attack for MDC-2 [5]. We make a multi-collision for the right half H_R of the output chaining value in the first block. The multi-collision from the first block fix the key inputs to the block ciphers in the second step. In the second step, we take a different approach of choosing $X = M_1 \oplus H_L$ at random, instead of M_1. Due to the fixed key input, the computations in the second block are almost independent of the first block. With this observation, we make a collision attack on MJH hash function as follows.

1. Choose sufficiently many message blocks in the first block, and obtain an r-collision for H_R. Denote the corresponding left halves of the output chaining variable and the message blocks by $H_L^{(1)}, H_L^{(2)}, ..., H_L^{(r)}$, and $M_0^{(1)}, M_0^{(2)}, ..., M_0^{(r)}$, respectively.

2. Choose randomly q distinct $X = H_L \oplus M$: $X^{(1)}, X^{(2)}, ..., X^{(q)}$, compute $V_L^{(i)} = E_{H_R}(X^{(i)}) \oplus X^{(i)}$ for $i = 1, 2, ..., q$, and collect the pairs of (i, j) such that $i \neq j$ and $V_L^{(i)} = V_L^{(j)}$ for $1 \leq i, j \leq q$.

3. For the pairs of (i, j) colliding on V_L, compute $Y^{(k)}$ for $k = i$ or j as follows:
$$Y^{(k)} = (E_{H_R}(\sigma(X^{(k)})) \oplus \sigma(X^{(k)})) \cdot \theta,$$
and check whether there is at least one pair of (u, v) for $1 \leq u, v \leq r$ such that $u \neq v$ and $H_L^{(u)} \oplus Y^{(i)} = H_L^{(v)} \oplus Y^{(j)}$. For such a tuple (i, j, u, v), the same V_R is produced from $H_L^{(u)} \oplus Y^{(i)}$ and $H_L^{(v)} \oplus Y^{(j)}$ or from $H_L^{(v)} \oplus Y^{(i)}$ and $H_L^{(u)} \oplus Y^{(j)}$. If such a tuple is found, output $M_0^{(u)} \| (H_L^{(u)} \oplus X^{(i)})$ and $M_0^{(v)} \| (H_L^{(v)} \oplus X^{(j)})$, or $M_0^{(v)} \| (H_L^{(v)} \oplus X^{(i)})$ and $M_0^{(u)} \| (H_L^{(u)} \oplus X^{(j)})$ as a collision.

In the first step, we need $((r!) \cdot 2^{(r-1)n})^{1/r}$ block cipher encryptions with key schedule operations to get an r-collision for H_R. The second step requires q block cipher encryptions. The time complexity of the last step is negligible compared to other steps. Since there are $\binom{r}{2}\binom{q}{2}$ possibilities for pairing $(X^{(i)}, X^{(j)})$'s and $(H_L^{(u)}, H_L^{(v)})$'s, we expect a collision for V with $\binom{r}{2}\binom{q}{2} = 2^{2n}$, where $\binom{r}{2}\binom{q}{2} \cong \frac{(rq)^2}{4}$ and $q \cong 2^{n+1}/r$. Overall, the time complexity of the above attack is estimated as

$$((r!) \cdot 2^{(r-1)n})^{1/r} T_{EK} + 2^{n+1}/r T_E. \tag{2}$$

Table 1: Time complexity of the collision attack on MJH with an n-bit block cipher, compared to birthday complexity, where $T_E \cong T_K$.

n	r	Collision Attack	Birthday Attack
64	8	$2^{60.58}$	2^{64}
128	14	$2^{123.81}$	2^{128}
256	23	$2^{251.03}$	2^{256}

Table 2: Time complexity of the collision attack on MJH with an n-bit block cipher, compared to birthday complexity, where $T_E \gg T_K$.

n	r	Collision Attack	Birthday Attack
64	9	$2^{61.01}$	2^{64}
128	15	$2^{124.27}$	2^{128}
256	25	$2^{251.50}$	2^{256}

(2) is approximated to

$$((r!) \cdot 2^{(r-1)n})^{1/r} \cdot \frac{2}{3} + 2^{n+1}/r \cdot \frac{1}{3} \tag{3}$$

for the case of $T_E \cong T_K$, and

$$(((r!) \cdot 2^{(r-1)n})^{1/r} + 2^{n+1}/r) \cdot \frac{1}{2} \tag{4}$$

for the case of $T_E \gg T_K$., respectively. For $n = 128$, we get the most efficient complexities of $2^{123.81}$ with $r = 14$ for (3) and $2^{124.27}$ with $r = 15$ for (4).

4 Preimage Attack on MJH

4.1 Preimage Attack on MJH Compression Function

It is easy to find a preimage of MJH compression function with time complexity of about $2^n \cdot T_E$. The preimage attack on MJH compression function is as follows.

1. Given a target hash value $V \in \{0,1\}^{2n}$, choose randomly $H_R \in \{0,1\}^n$ and fix it.

2. Find $X \in \{0,1\}^n$ such that $E_{H_R}(X) \oplus X = V_L$ with brute force attack.

3. If such X is found, compute $Y = (E_{H_R}(\sigma(X)) \oplus \sigma(X)) \cdot \theta$, $H_L = Y \oplus V_R$, and $M = H_L \oplus X$.

4. Output (H_L, H_R, M) as a preimage for V.

4.2 Preimage Attack on MJH Hash Function

We have to consider the padding rule for constructing a preimage attack on a hash function from a preimage attack on a compression function. We assume that the last message block contains a length information of the message. In the attack described in Section , the attacker does not have a control on the message block unlike the preimage attack on MDC-4 compression function described in Section . So, the attacker can not intend to embed a predetermined length information to the preimage, and we can not use Knudsen et al.'s time-memory trade off technique to make a preimage attack for MJH hash function from the preimage attack for MJH compression function. Alternatively, we make it using the meet-in-the-middle technique [8, Fact 9.99] and expandable messages with fixed-points [1, 4].

A fixed-point for a compression function $\mathrm{CF}(H, M) = V$ is defined as (H, M) such that $\mathrm{CF}(H, M) = H$. We can find fixed-points for MJH compression function as follows.

1. Choose randomly $M \in \{0,1\}^n$ and $H_R \in \{0,1\}^n$, fix them.

2. Compute $X = E^{-1}_{H_R}(M)$, $H_L = M \oplus X$, $Y = (E_{H_R}(\sigma(X)) \oplus \sigma(X)) \cdot \theta$, and $H_L \oplus Y = V_R$.

3. If $V_R = H_R$, then output (H_L, H_R, M) as a fixed-point; else, repeat the computations in step 2 with new random choices of M and H_R.

On average, we expect to find a fixed-point with time complexity of 2^n.

An expandable message is constructed as follows.

1. Collect $2^{n/2}$ fixed-points $(H^{(1)}, M_2^{(1)}), ..., (H^{(2^{n/2})}, M_2^{(2^{n/2})})$ by repeating the above search.

2. Repeat to compute $\mathrm{CF}^{\mathrm{MJH}}(\mathrm{CF}^{\mathrm{MJH}}(IV, M_0), M_1)$ for a randomly chosen two-block message (M_0, M_1) until the result is matched with any $H^{(i)}$ for $i = 1, ..., 2^{n/2}$.

3. If a match is found, then output the corresponding $(M_0, M_1, M_2^{(i)})$ as a $(2, \infty)$-expandable message.

On average, the number of repetition in the step 2 should be $2^{3n/2}$ to expect a match. So, the time complexity for the above construction of an expandable message is about $2^{3n/2+1}$.

Assume that we are given a $(2, \infty)$-expandable message (M_0, M_1, M_2) made from a fixed-point (H, M_2). Let $\mathrm{len}(M)$ be the length information of the hashed message M contained in the last message block. With this expandable message, we can construct a preimage attack on MJH hash function as follows.

1. Given a target hash value V, collect $2^{n/2}$ preimages $(U^{(1)}, M_L^{(1)})$, $(U^{(2)}, M_L^{(2)})$, ..., $(U^{(2^{n/2})}, M_L^{(2^{n/2})})$ for the last compression function.
2. Repeat to compute $\mathrm{CF}^{\mathrm{MJH}}(H, M_{L-1})$ for a randomly chosen one-block message M_{L-1} until the result is matched with any $U^{(i)}$ for $i = 1, ..., 2^{n/2}$.
3. If a match is found, then output the corresponding $(M_0, M_1, M_2, ..., M_2, M_{L-1}, M_L)$ as a preimage for V, where the repetition number of M_2 depends on $\mathrm{len}(M)$ contained in M_L.

On average, the number of repetition in the step 2 should be $2^{3n/2}$ to expect a match. So, the time complexity for the above construction of an expandable message is about $2^{3n/2+1}$.

Finally, a preimage attack on MJH hash function is made from the preimage attack on MJH compression function and the expandable message by the meet-in-the-middle technique in [8, Fact 9.99]. Overall time complexity is about $2^{3n/2+2}$.

5 Conclusion

In this paper, we presented collision and preimage attacks on MJH. These are the first cryptanalytic results for it. Since we used generic setting, our attacks work for MJH with any secure block ciphers. Our results show that MJH can not reach the security levels which are traditionally expected for cryptographic hash functions, while it is provably secure in the ideal cipher model with about less than $2^{2n/3-\log n}$ queries.

References

[1] R. D. Dean, *Formal Aspects of Mobile Code Security*, Ph. D Dissertion, Princeton University, January 1999.

[2] S. Hirose, "Some Plausible Constructions of Double-Block-Length Hash Functions," In M. J. B. Robshaw (Ed.), *FSE 2006*, LNCS 4047, pp. 231–246, Springer-Verlag, 2006.

[3] ISO/IEC 10118-2:2010, "Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher," 1994, revised in 2010.

[4] J. Kelsey and B. Schneier, "Second Preimages on n-Bit Hash Functions for Much Less than 2^n Work," *EUROCRYPT 2005*, LNCS 3494, pp. 474–490, Springer-Verlag, 2005.

[5] L. R. Knudsen, F. Mendel, C. Rechberger, and S. Thomsen, "Cryptanalysis of MDC-2," *EUROCRYPT 2009*, LNCS 5479, pp. 106–120, Springer-Verlag, 2009.

[6] X. Lai and J. L. Massey, "Hash function based on block ciphers," In R. A. Rueppel (Ed.), *EUROCRYPT'92*, LNCS 658, pp. 55–70, Springer-Verlag, 1993.

[7] J. Lee and M. Stam, "MJH: A Faster Alternative to MDC-2," In A. Kiayias (Ed.), *CT-RSA 2011*, LNCS 6558, pp. 213–236, Springer-Verlag, 2011.

[8] A. J. Menezes, P. C. Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[9] C. Meyer and M. Schilling, "Secure Program Load with Manipulation Detection Code," *Proc. Securicom*, pp. 111–130, 1988.

[10] B. Preneel, R. Govaerts and J. Vandewalle, "Hash Functions Based on Block Ciphers: A Synthetic Approach," In D. R. Stinson (Ed.), CRYPTO 1993, LNCS 773, pp. 363–378, Springer-Verlag, 1994.