New Attacks on all Double Block Length Hash Functions of Hash Rate 1, including the Parallel-DM

Lars R. Knudsen[1] and Xuejia Lai[2]

[1] Aarhus University, Denmark
[2] R[3] Security Engineering, Aathal, Switzerland

Abstract. In this paper attacks on double block length hash functions using a block cipher are considered. We present attacks on all double block length hash functions of hash rate 1, that is, hash functions where in each round the block cipher is used twice, s.t. one encryption is needed per message block. In particular, our attacks break the *Parallel-DM* presented at Crypto'93[3].

1 Introduction

A ***hash function*** is an easily implementable mapping from the set of all binary sequences to the set of binary sequences of some fixed length. An ***iterated hash function*** is a hash function Hash$(\cdot)$ determined by an easily computable function $h(\cdot,\cdot)$ from two binary sequences of respective lengths m and l to a binary sequence of length m in the manner that the message $M = (M_1, M_2, ..., M_n)$, where M_i is of length l, is hashed to the ***hash value*** $H = H_n$ of length m by computing recursively

$$H_i = h(H_{i-1}, M_i) \qquad i = 1, 2, ..., n, \tag{1}$$

where H_0 is a specified ***initial value***. The function h will be called the ***hash round function***. We will consider iterated hash functions based on (m, k) block ciphers, where an (m, k) ***block cipher*** defines, for each k-bit key, a reversible mapping from the set of all m-bit plaintexts onto the set of all m-bit ciphertexts. We write $E_Z(X)$ to denote the encryption of the m-bit plaintext X under the k-bit key Z, and $D_Z(Y)$ to denote the decryption of the m-bit ciphertext Y under the k-bit key Z. We define the ***hash rate*** of such an iterated hash function (or equivalently, of a round function) as the number of m-bit message blocks processed per encryption or decryption. The *complexity* of an attack is the total number of encryptions or decryptions required for the attack. In our discussion we will always assume that the block length of the block cipher equals the key length and that the (m, m) block cipher has no known weaknesses.

To avoid some trivial attacks [7], the Merkle-Damgaard Strengthening *(MD-strengthening)* is often used, in which the last block of the message to be hashed represents the binary length of the true message. However, in the attacks presented in this paper the messages are of the same length, therefore we will not consider MD-strengthening anymore in this paper.

2 Double block length hash functions

Since most block ciphers have a block length of only 64 bits, for a single block length hash function the complexity of a brute force collision attack is only 2^{64-n} encryptions using a table of size about 2^n 64 bits quantities. As an example, with $n = 20$ and using todays technology this is computationally feasible, and the space requirements are not too large. Therefore many attempts have been made to construct hash round functions based on two parallel or consecutive runs of a block cipher, thereby obtaining a hash code of size $2m$ bits.
Natural requirements for double block length hash functions based on an m-bit block cipher are that the complexity of a target attack is higher than 2^m and that the complexity of a collision attack is higher than $2^{m/2}$. Recently, one such scheme has been submitted for publication as an ISO standard [4], also known as the MDC-2. It is believed that the complexities for target and collision attacks on MDC-2 based on DES is about 2^{81} and 2^{54} [5], where m above is 64. Since the hash rate of the MDC-2 is only 1/2, i.e. the hash function takes two encryptions per message block, attempts have been made to construct double block length hash functions of hash rate 1 [1, 3, 10]. Consider the following general form of a double block length hash function.

$$\begin{cases} H_i^1 &= E_A(B) \oplus C \\ H_i^2 &= E_R(S) \oplus T \end{cases} \tag{2}$$

where, for a hash rate 1 scheme, A, B and C are binary linear combinations of the m-bit vectors H_{i-1}^1, H_{i-1}^2, M_i^1 and M_i^2, and where R, S and T are some binary linear combinations of the vectors H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2 and H_i^1. In [3] the following result was proved.

Theorem 1 (HLMW-93 [3]) *For the $2m$-bit iterated hash function with hash rate $1/2$ or 1 whose $2m$-bit round function is of type (2), the complexity of a free-start target attack is upper-bounded by about $2 \cdot 2^m$ and the complexity of a free-start collision attack is upper-bounded by about $2 \cdot 2^{m/2}$.*

Hash functions obtaining these upper bounds as lower bounds for the free-start attacks are said to be *optimum* against a free-start attack [3]. The idea is, that given a specific initial value of the hash function the designer hopes that the complexity of collision and target attacks are higher than the proven lower bounds. In [3], the **Parallel-DM**, a new double block length hash function of rate 1 with optimum security against free-start attacks was proposed. We give two attacks on Parallel-DM, a target attack and a collision attack with about the same complexities as of the free-start target and free-start collision attacks. This means that the Parallel-DM is no more secure than the Davies-Meyer hash mode (DM), which was the purpose in the first place. Our attacks can be generalized and the following result holds

Theorem 2 *Consider a double block length hash function with round function of the form (3), where each h^i contains one encryption.*

$$\begin{cases} H_i^1 = h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \\ H_i^2 = h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \end{cases} \tag{3}$$

If for a fixed value of H_i^1 (or H_i^2 or $H_i^1 \oplus H_i^2$), it takes T operations to find one pair of (M_i^1, M_i^2) for any given value of (H_{i-1}^1, H_{i-1}^2), such that the resulting 4-tuple $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$ yields the fixed value for H_i^1 (or H_i^2 or $H_i^1 \oplus H_i^2$), then a target attack on the hash function needs at most $(T+3) \cdot 2^m$ operations; and a collision attack on the hash function needs at most $(T+3) \cdot 2^{m/2}$ operations. The attacks succeed with probability 0.63.

Proof: The target attack: Let (H_0^1, H_0^2) be the given initial value and (H_n^1, H_n^2) be the hash code of a message M. We proceed as follows:

1. Compute forward the pair (H_{n-1}^1, H_{n-1}^2) from the given hash value (H_{n-2}^1, H_{n-2}^2) and a pair of messages (M_{n-1}^1, M_{n-1}^2) randomly chosen.
2. Find the pair (M_n^1, M_n^2) from the pair (H_{n-1}^1, H_{n-1}^2) obtained above so that the 4-tuple $(H_{n-1}^1, H_{n-1}^2, M_n^1, M_n^2)$ yields the fixed value for H_n^1.
3. Compute the value for H_n^2 from the 4-tuple $(H_{n-1}^1, H_{n-1}^2, M_n^1, M_n^2)$.

Repeat the above procedure 2^m times. Note that H_n^2 is m bits long, so after obtaining 2^m values of H_n^2, with a high probability we hit the given value of H_n^2. Finally, note that step 1 takes two operations, step 2 T operations and step 3 one operation.

The collision attack: Let (H_0^1, H_0^2) be the given initial value. We shall find two different messages M and M', such that both messages yield the same hash code (H_n^1, H_n^2). Choose some random values and compute a value for H_n^1 and fix it, then proceed in the same way as in the target attack, i.e. perform steps 1, 2 and 3 above. Repeat this procedure $2^{m/2}$ times. Because H_n^2 is m bits long, the "birthday argument" implies that some two values of the H_n^2 will be the same with a high probability.□

We will show that for the Parallel-DM, the T of Theorem 2 is about zero. The scheme is defined

$$\begin{cases} H_i^1 &= \mathbf{E}_{M_i^1 \oplus M_i^2}(H_{i-1}^1 \oplus M_i^1) \oplus H_{i-1}^1 \oplus M_i^1 \\ H_i^2 &= \mathbf{E}_{M_i^1}(H_{i-1}^2 \oplus M_i^2) \oplus H_{i-1}^2 \oplus M_i^2 \end{cases} \tag{4}$$

Theorem 3 *There exists a target attack on the Parallel-DM scheme that given a message M and its hash value $H(M)$ finds a message M', s.t. $H(M) = H(M')$. The attack succeeds with probability* 0.63 *in time* 3×2^m. *There exists a collision attack on the Parallel-DM scheme that given IV finds two message $M \neq M'$, s.t. $H(IV, M) = H(IV, M')$. The attack succeeds with probability* 0.63 *in time* $3 \times 2^{m/2}$.

Proof: Let A and B be two fixed (given or chosen) values such that $H_i^1 = E_B(A) \oplus A$. For any given value of (H_{i-1}^1, H_{i-1}^2), one can obtain one pair of (M_i^1, M_i^2) where

$$M_i^1 = A \oplus H_{i-1}^1 \text{ and } M_i^2 = B \oplus M_i^1$$

such that the 4-tuple $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$ will yield the fixed value for H_i^1 in (4). Theorem 2 then implies that the complexity of a target attack is about $3 \cdot 2^m$ (with $T = 0$) and the complexity of a collision attack is about $3 \cdot 2^{m/2}$. □

Theorem 2 is for the "parallel" version of a double block length hash function, where the two encryptions work side-by-side. A similar result holds for the "serial" version of a double block length hash function, which is proved in a similar manner as Theorem 2.

Theorem 4 *Consider a double block length hash function of hash rate 1 with round function of the form (5), where each h^i contains one encryption.*

$$\begin{cases} H_i^1 = h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \\ H_i^2 = h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2, H_i^1) \end{cases} \tag{5}$$

If for a fixed value of H_i^1, it takes T operations to find one pair of (M_i^1, M_i^2) for any given value of (H_{i-1}^1, H_{i-1}^2), such that the resulting 4-tuple $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$ yields the fixed value for H_i^1, then a target attack on the hash function needs at most $(T+3) \cdot 2^m$ operations; and a collision attack on the hash function needs at most $(T+3) \cdot 2^{m/2}$ operations.

3 Attacks on all double block length hash functions of hash rate 1

In [11] it was shown that there exist basically two secure single block length hash functions. The Davies-Meyer scheme,

$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1} \tag{6}$$

is one of them, the other one is the following

$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1} \oplus M_i \tag{7}$$

All other secure single block length hash functions can be transformed into either (6) or (7) by a linear transformations of the inputs M_i and H_{i-1} [11]. It means that for a double block length hash function one can obtain *optimum* security against free-start attacks if the scheme is equivalent to either two runs of (6) or two runs of (7) by a simple invertible transformation of the inputs M_i^1, M_i^2, H_{i-1}^1 and H_{i-1}^2.
We show that the double block length hash functions of hash rate 1, where (at least) one of the hash round functions has the form of any single block length hash function, has a security not much higher than for the single block length hash function. Also we show target attacks on all double block length hash functions of rate 1. In the following we will consider double block length hash functions of the form (2). We consider schemes of hash rate 1, that is, we can write

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{bmatrix} \tag{8}$$

for some binary values a_i, b_i and c_i $(1 \leq i \leq 4)$. We denote by L the 3×4 matrix in eq. (8).

Theorem 5 *For the 2m-bit iterated hash function with rate 1, where (at least) one of the hash round functions has the form of a single block length hash function, i.e. the matrix L of (8) has a rank of less than or equal to two, the complexity of a target attack is upper-bounded by about* 3×2^m, *and the complexity of a collision attack is upper-bounded by about* $3 \times 2^{m/2}$. *The attacks succeed with probability about 0.63.*

Proof: We will show that the T of Theorem 2 is about zero. We assume w.l.g. that the hash round functions of type (8) is H_i^1 and that we are given the target (H_n^1, H_n^2).
$Rank(L) = 1$: Trivial, since with the same intermediate hash values (H_{n-1}^1, H_{n-1}^2) used in the computation of the target H_n^1, there are at least 2^m possible values of (M_n^1, M_n^2) obtaining H_n^1. Thus, Theorem 4 holds with $T \simeq 0$.
$Rank(L) = 2$: We can rewrite (8) as follows

$$\begin{bmatrix} A \\ B \end{bmatrix} = N_1 \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \end{bmatrix} \oplus N_2 \begin{bmatrix} M_i^1 \\ M_i^2 \end{bmatrix} \tag{9}$$

where N_1 and N_2 are 2×2 binary matrices. We distinguish between cases depending on the rank of N_2.
$Rank(N_2) \leq 1$: With the intermediate hash values (H_{n-1}^1, H_{n-1}^2) used in the computation of the target H_n^1, there are at least 2^m possible values of (M_n^1, M_n^2) obtaining H_n^1. Thus, Theorem 4 holds with $T \simeq 0$.
$Rank(N_2) = 2$: N_2 is invertible and we can rewrite (9) into

$$\begin{bmatrix} M_i^1 \\ M_i^2 \end{bmatrix} = N_2^{-1} \left[N_1 \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \end{bmatrix} \oplus \begin{bmatrix} A \\ B \end{bmatrix} \right] \tag{10}$$

Given the target H_n^1 and by letting (A, B) be the values used in the computation of the target H_n^1, we can find (M_n^1, M_n^2) for any values (H_{n-1}^1, H_{n-1}^2), s.t. we hit the target H_n^1. Thus, Theorem 4 holds with $T \simeq 0$, (time used to invert the matrix N_2 and to do the adding operations is negligible). The Parallel-DM [3] is an instance of this class of hash functions. □

Theorem 6 *For the double block length hash functions of hash rate 1, for which one of the m-bit hash round functions are of type (8), the complexity of a target attack is upper bounded by about* 4×2^m. *For two classes of hash functions, the attack needs a pre-computed table with* 2^m *2m-bit values.*

Proof: We will show that the T of Theorem 4 is at most 1. We assume w.l.g. that the hash round functions of type (8) is H_i^1 and that we are given the target (H_n^1, H_n^2). We denote by L the 3×4 matrix in (8).
$Rank(L) < 3$: Proved in Theorem 5.
$Rank(L) = 3$: The first hash round function in this scheme has the form $H_i^1 = E_A(B) \oplus C$, where A, B and C are linearly independent. A and B can be expressed as in (9). We split the proof into two cases.
$Rank(N_2) = 1$. Let M_Z be the set $\{M_i^1, M_i^2, M_i^1 \oplus M_i^2\}$ and let $M_{ab} \in M_Z$ be the message variable contained in A and B. If C does not contain any of the messages in M_Z or contains only M_{ab}, Theorem 4 holds with $T \simeq 0$, since in this case we use the same intermediate values (H_{n-1}^1, H_{n-1}^2) used in the computation of the target H_n^1 (i.e. use the same messages $M_1, ..., M_{n-1}$). Since the rank of N_2 is one, there are 2^m possible values of (M_n^1, M_n^2) obtaining the hash code H_n^1.
If C contains one message $M_c \in M_Z$, s.t. $M_c \neq M_{ab}$ then for any given (H_{n-1}^1, H_{n-1}^2), compute $E_A(B) = z$ for a random value of M_{ab}. Now use the correct value of the 2^m possible values of M_c to hit H_n^1, i.e. such that $C \oplus z = H_n^1$. In this case Theorem 4 holds with $T \simeq 1$. The PBGV hash function proposed in [9] is an instance of this class of hash functions.
$Rank(N_2) = 2$. H_i^1 can be written

$$\begin{aligned} H_i^1 &= E_A(B) \oplus C^0 \\ &= E_A(B) \oplus B \oplus C^1 \\ &= E_A(B) \oplus A \oplus B \oplus C^2 \end{aligned}$$

Since the rank of L is 3 and the rank of N_2 is 2, either C^0, C^1 or C^2 does not contain any of the messages M^1, M^2 or $M^1 \oplus M^2$. Let C^i denote that value of C.
In the case where $C^i = C^0$, for any given value of (H_{n-1}^1, H_{n-1}^2) and thereby also for C^0, it is possible to find (M_n^1, M_n^2) s.t. the target H_n^1 is hit. Simply decrypt $D_A(C^0 \oplus H_n^1) = B$ using one of the two free message variables in A and using the other free message variable to adjust to the given (H_{n-1}^1, H_{n-1}^2) appearing in B. Again Theorem 4 holds with $T \simeq 1$.
In the case where $C^i = C^1$, we first pre-compute (and sort) a table KT of 2^m triples (K_l, x_l, y_l), s.t.

$$K_l = E_{x_l}(y_l) \oplus y_l$$

for random values (x_l, y_l). Then for any given (H_{n-1}^1, H_{n-1}^2) compute $Q = C^1 \oplus H_n^1$. Look up $Q = K_j$ in table KT and set $A = x_j$ and set $B = y_j$ for A and B in equation (9). Since N_2 is invertible, by assumption, we find the values of (M_n^1, M_n^2), s.t. the target H_n^1 is hit. Theorem 4 holds with $T \simeq 0$. We have assumed here that the time to sort a table of size

2^m is negligible compared to the time of 2^m encryptions. The LOKI-DBH hash function proposed in [1] is an instance of this class of hash functions.
In the case where $C^i = C^2$, we first pre-compute (and sort) a table KT of 2^m triples (K_l, x_l, y_l), s.t.

$$K_l = E_{x_l}(y_l) \oplus x_l \oplus y_l$$

for random values (x_l, y_l) and proceed similar as in the previous case. □

4 Conclusion

We have shown attacks on double block length hash functions of hash rate 1. Our attacks show that a double block hash function of hash rate 1, which has optimum security against free-start attacks, is also vulnerable to real attacks with only slighty higher complexities. Furthermore we have shown that for all double block length hash functions of hash rate 1 based on a secret key block cipher, there exist target attacks with complexity of about 4×2^m. In some cases the attack needs a pre-computed table of size 2^m.

References

1. L. Brown, J. Pieprzyk and J. Seberry, "LOKI – A Cryptographic Primitive for Authentication and Secrecy Applications", Advances in Cryptology – AUSCRYPT'90, Proceedings, LNCS 453, pp. 229-236, Springer-Verlag, 1990.
2. I. B. Damgaard, "A Design Principle for Hash Functions", Advances in Cryptology - CRYPTO'89, LNCS 435, pp. 416-427, Springer-Verlag, 1990.
3. W. Hohl, X. Lai, T. Meier and C. Waldvogel, "Security of Iterated Hash Function Based on Block Ciphers", Advances in Cryptology - CRYPTO'93 Proceedings, pp. 379-390, LNCS 773, Springer Verlag, 1994.
4. ISO/IEC 10118, Information technology – Security techniques – Hash-functions, Part 2:*Hash-functions using an n-bit block cipher*, I.S.O., 1994.
5. X. Lai, *On the Design and Security of Block Ciphers*, ETH Series in Information Processing (Edt: J. L. Massey), Vol. 1, Hartung-Gorre Verlag, Konstanz, 1992.

6. X. Lai and L. Knudsen "Attacks on Double Block Length Hash Functions" To appear in the proceedings from The Algortihm Workshop, Cambridge, U.K., Dec. 1993.
7. X. Lai and J.L. Massey, "Hash Functions Based on Block Ciphers", Advances in Cryptology - EUROCRYPT'92 Proceedings, pp. 55-70, LNCS 658, Springer Verlag, 1993.
8. C. H. Meyer and M. Schilling, "Secure Program Code with Modification Detection Code", Proceedings of SECURICOM 88, pp. 111-130, SEDEP.8, Rue de la Michodies, 75002, Paris, France.
9. B. Preneel, A. Bosselaers, R. Govaerts and J. Vandewalle, "Collision-free Hashfunctions Based on Blockcipher Algorithms", Proceedings of 1989 International Carnahan Conference on Security Technology, pp. 203-210, 1989.
10. B. Preneel, *Analysis and Design of Cryptographic Hash Hashfunctions* , Ph.D thesis, Katholieke Universiteit Leuven, Belgium, January 1993.
11. B. Preneel, "Hash functions based on block ciphers: A synthetic approach", Advances in Cryptology - Proceedings of Crypto'93, pp. 368–378, LNCS 773, Springer Verlag , 1994.