Attacks on Double Block Length Hash Functions

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

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

Abstract. Attacks on double block length hash functions using a block cipher are considered in this paper. We present a general free-start attack, in which the attacker is free to choose the initial value, and a real attack on a large class of hash functions. Recent results on the complexities of attacks on double block hash functions are summarized.

1 Introduction

A *hash function* is an easily implementable mapping from the set of all binary sequences of some specified minimum length or greater to the set of binary sequences of some fixed length. In cryptographic applications, hash functions are used within digital signature schemes and within schemes to provide data integrity (e.g., to detect modification of a message). An *iterated hash function* is a hash function $\mathrm{Hash}(\cdot)$ 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

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

where H_0 is a specified *initial value*. We will write $H = \mathrm{Hash}(H_0, M)$ to show explicitly the dependence on H_0. The function h will be called the *hash round function*. For message data whose total length in bits is not a multiple of l, one can apply deterministic "padding" [5, 10] to the message to be hashed by (1) to increase the total length to a multiple of l.

For iterated hash functions, we distinguish the following five attacks:

1. **Target attack:** Given H_0 and M, find M' with $M' \neq M$ but $\mathrm{Hash}(H_0, M') = \mathrm{Hash}(H_0, M)$.
2. **Free-start target attack:** Given H_0 and M, find H_0' and M' such that $(H_0', M') \neq (H_0, M)$ but $\mathrm{Hash}(H_0', M') = \mathrm{Hash}(H_0, M)$.
3. **Collision attack:** Given H_0, find M and M' such that $M' \neq M$ but $\mathrm{Hash}(H_0, M') = \mathrm{Hash}(H_0, M)$.
4. **Semi-free-start collision attack:** Find H_0, M and M' such that $M' \neq M$ but $\mathrm{Hash}(H_0, M') = \mathrm{Hash}(H_0, M)$.

5. **Free-start collision attack:** Find H_0, H_0', M and M' such that $(H_0', M') \neq (H_0, M)$ but $\mathrm{Hash}(H_0', M') = \mathrm{Hash}(H_0, M)$.

Remark. Target attacks are also called "preimag" attacks and free-start attacks are also referred as "pseudo" attacks [14]. In applications where H_0 is specified and fixed, attacks 2, 4 and 5 are not "real attacks". This is because the initial value H_0 is then an integral part of the hash function so that a hash value computed from a different initial value will not be accepted. However, if the sender is free to choose and/or to change H_0, attacks 2, 4 and 5 can be real attacks, depending on the manner in which the hash function is used. Note that the free-start and semi-free-start attacks are never harder than the attacks where H_0 is specified in advance.

For an m-bit hash function, brute-force target attacks, in which one randomly chooses an M' until one hits the "target" $H = \mathrm{Hash}(H_0, M)$, require about 2^m computations of hash values. It follows from the usual "birthday argument" that brute-force collision attacks require about $2^{m/2}$ computations of hash values. In particular, for hash round functions with $l \geq m$ so that all 2^m hash values can be reached with one-block messages, brute-force target attacks require about 2^m computations of the round function h while brute-force collision attacks require about $2^{m/2}$ computations of the round function h. We will say that the computational security of the hash function is **ideal** when there is no attack substantially better than brute force.

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 an 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 (m, k) block cipher has no known weaknesses, so the results can be applied to any block cipher. For the security of hash functions based on specific ciphers, see [1, 14].

Because an attack on the m-bit round function implies an attack of the same type on the corresponding m-bit iterated hash function with roughly the same complexity, the design of computationally secure round functions is a necessary (but not sufficient) condition for the design of computationally secure iterated hash functions. Moreover, under certain conditions (cf. [3, 6, 10, 13]), a computationally secure round function implies a computationally secure iterated hash function. To avoid some trivial attacks [8], the Merkle-Damgaard Strengthening *(MD-strengthening)* will always be assumed, in which the last block of the message to be hashed represents the binary length of the true message.

2 Double block length hash functions using block ciphers

A well-known example of an iterated hash function is the Davies-Meyer scheme (DM), where the hash round function is given by

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

Here $E_K(P)$ is the encrypted value of plaintext P using key K with block cipher E. The DM-scheme with MD-strengthening is generally considered to be secure if the underlying block cipher with block size m has no weaknesses. Thus, we will assume that, for the single block DM-scheme, the complexity of a free-start collision attack is about $2^{m/2}$ and the complexity of a free-start target attack is about 2^m.

Since most block ciphers have a block length of only 64 bits, the hash code of the DM-scheme is only 64 bits. A collision attack needs at most about 2^{32} encryptions, which can be done reasonably fast using today's technology. Therefore, much research has been done to construct hash functions with a block length of $2m$ bits based on the concatenation of two variants of the DM-scheme. One such scheme, the MDC-2 [9, 11] will be published as an ISO standard [5]. A systematic method proposed in [4] to analyze such hash functions is to consider the following general form of double length round functions.

General form of the $2m$-bit round function with rate 1:

$$\begin{cases} H_i^1 &= E_A(B) \oplus C \\ H_i^2 &= E_R(S) \oplus T \end{cases} \tag{3}$$

where, for a 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 (not necessarily binary linear) combinations of the vectors H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2 and H_i^1. For a rate 1/2 scheme, A, B and C are binary linear combinations of the m-bit vectors H_{i-1}^1, H_{i-1}^2, M_i and R, S and T are some combinations of the m-bit vectors H_{i-1}^1, H_{i-1}^2, M_i and H_i^1.

We can write, in case of a rate 1 scheme, A, B and C in matrix-form as

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{bmatrix} \tag{4}$$

for some binary values a_i, b_i and c_i $(1 \leq i \leq 4)$. For a rate 1/2 scheme, we have

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i \end{bmatrix} \tag{5}$$

for some binary values a_i, b_i and c_i $(1 \leq i \leq 3)$.

First, we consider the complexity of free-start attacks on such hash functions [4].

Theorem 1 *For the $2m$-bit iterated hash function with rate $1/2$ or 1 whose $2m$-bit round function is of type (3), 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 $2 \cdot 2^{m/2}$.*

Proof: We show the result for the case of rate $1/2$. The proof for rate 1 can then be easily derived.

First consider the *free-start target attack*: for a given value of $(H_{i-1}^1, H_{i-1}^2, M_i)$ with output (H_i^1, H_i^2), find a different value for $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value for (H_i^1, H_i^2). When the linear transformation matrix defined in (5) is non-singular, let D be the value of H_i^1 for the given value of $(H_{i-1}^1, H_{i-1}^2, M_i)$. We then generate 2^m different values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value D by first computing $C = D \oplus \mathbf{E}_A(B)$ for 2^m randomly chosen values of (A, B) and then, for each value of (A, B, C), we can determine the value of $(H_{i-1}^1, H_{i-1}^2, M_i)$ by computing the inverse transformation of (5). When the matrix $\prod$ is singular, there exist, for the value of (A, B, C) obtained from the given value of $(H_{i-1}^1, H_{i-1}^2, M_i)$, at least 2^m different values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value for (A, B, C), i.e., the same value for H_i^1. For the given and the 2^m newly generated values of $(H_{i-1}^1, H_{i-1}^2, M_i)$, we compute the value of H_i^2 according to (3). Because there are 2^m possible values of the m-bit block H_i^2, it follows that one must compute H_i^2 for about 2^m different values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ to have a probability of 0.63 to find a value of $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value for H_i^2 as the given value of $(H_{i-1}^1, H_{i-1}^2, M_i)$. Such an attack requires therefore at most $2 \cdot 2^m$ encryptions.

Next we consider the *free-start collision attack*, i.e., we will find two different values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value for (H_i^1, H_i^2) according to (3). This attack is similar to the free-start target attack just described, except that here, one only generates $2^{m/2}$ values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value of H_i^1. This follows from the usual "birthday paradox" which says that one only needs to try $2^{m/2}$ randomly chosen values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ to have a probability of 0.63 to find two values of $(H_{i-1}^1, H_{i-1}^2, M_i)$ yielding the same value for H_i^2. □

Remark. The basic idea behind the attacks in the proof of the above theorem is to attack the two equations in (3) separately. If one can find many values for (H_{i-1}^1, H_{i-1}) yielding the same value for H_i^1, then the attack on the $2m$-bit round function of type (3) is reduced to an attack on one m-bit round function. Thus, similar attacks will also work even if the mapping from $(H_{i-1}^1, H_{i-1}^2, M_i)$ to (A, B, C) in (3) is not a binary linear combination.

Such a method of "separately attack the two equations" can also be used in real attacks, namely, the **solving-one-half** attacks used in [7], as shown in the following results.

Theorem 2 *Consider a double block length hash function of rate 1 with hash round function of the form (6), where each h^i contains one encryption.*

$$\begin{cases} H_i^1 = h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \\ H_i^2 = h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2). \end{cases} \tag{6}$$

If for a fixed value of H_i^1 (or H_i^2 or $H_i^1 \oplus H_i^2$), it takes T computations of encryption or decryption 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 $(T+3) \cdot 2^m$ computations of encryption or decryption; and a collision attack on the hash function needs at most $(T+3) \cdot 2^{m/2}$ computations of encryption or decryption.

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

1. compute forward the pair (H_1^1, H_1^2) from the initial value and a randomly chosen pair of messages (M_1^1, M_1^2).
2. find the pair (M_2^1, M_2^2) from the pair (H_1^1, H_1^2) obtained above so that the 4-tuple $(H_1^1, H_1^2, M_2^1, M_2^2)$ yields the fixed value for H^1.
3. compute the value for H^2 from the 4-tuple $(H_1^1, H_1^2, M_2^1, M_2^2)$.

Repeat the above procedure 2^m times. Note that H^2 is m bits long, so after obtaining 2^m values of H^2, with a high probability we hit the given value of H^2.

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^1, H^2).

Choose a value for H^1 and fix it, then proceed as follows:

1. compute forward the pair (H_1^1, H_1^2) from the initial value and a randomly chosen pair of messages (M_1^1, M_1^2).
2. find the pair (M_2^1, M_2^2) from the pair (H_1^1, H_1^2) obtained above so that the 4-tuple $(H_1^1, H_1^2, M_2^1, M_2^2)$ yields the fixed value for H^1.
3. compute the value for H^2 from the 4-tuple $(H_1^1, H_1^2, M_2^1, M_2^2)$.

Repeat this procedure $2^{m/2}$ times. Because H^2 is m bits long, the "birthday argument" implies that some two values of the H^2 will be the same with high probability. □

Theorem 1 showed that for the class of hash-functions of the form (3) the complexities of free-start target and free-start collision attacks are upper bounded by 2^m and $2^{m/2}$, respectively. Hash functions achieving these upper bounds for the free-start attacks are said to be *optimum* against a free-start attack [4]. The Parallel-DM scheme was shown in [4] to be optimum. The idea is that given a specific initial value of the hash function one hopes that the complexity of usual collision and target attacks are higher than the proven lower bounds for free-start attacks. However, using the solving-one-half attack, the complexity of usual collision and target attacks are shown to be the same as the complexities for free-start attacks.

The Parallel-DM scheme. This scheme is a $2m$-bit hash function based on an m-bit block cipher with an m-bit key and is defined as follows

$$H_i^1 = E_{M_i^1 \oplus M_i^2}(H_{i-1}^1 \oplus M_i^1) \ \oplus \ H_{i-1}^1 \oplus M_i^1 \tag{7}$$

$$H_i^2 = E_{M_i^1}(H_{i-1}^2 \oplus M_i^2) \ \oplus \ H_{i-1}^2 \oplus M_i^2. \tag{8}$$

Attacks on the Parallel-DM scheme by applying Theorem 2.

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 (7). 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 $3 \cdot 2^{m/2}$. Note that the single block hash function DM-scheme has roughly the same complexities. More details can be found in [7].

Attacks on the PBGV scheme by applying Theorem 2.

This scheme was proposed in [15] and its round function is defined as follows.

$$H_i^1 = E_{M_i^1 \oplus M_i^2}(H_{i-1}^1 \oplus H_{i-1}^2) \oplus M_i^1 \oplus H_{i-1}^1 \oplus H_{i-1}^2 \tag{9}$$

$$H_i^2 = E_{M_i^1 \oplus H_{i-1}^1}(M_i^2 \oplus H_{i-1}^2) \oplus M_i^2 \oplus H_{i-1}^1 \oplus H_{i-1}^2. \tag{10}$$

Fix a value for H_i^1. Chose a fixed value K as the key input in (9). For any given value of (H_{i-1}^1, H_{i-1}^2), let $d = H_{i-1}^1 \oplus H_{i-1}^2$, then one can obtain one pair of (M_i^1, M_i^2) where

$$M_i^1 = E_K(d) \oplus d \oplus H_i^1 \text{ and } M_i^2 = K \oplus M_i^1$$

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 (9). Theorem 2 then implies that the complexity of target attack is about $4 \cdot 2^m$ and the complexity of collision attack is about $4 \cdot 2^{m/2}$. Note that the similar attacks have been reported before in [6, 14], but the above attack has a simpler form.

The result of Theorem 2 is for the "parallel" form of the round function in which the two encryptions work side-by-side. Similar attack can also be applied to the "serial" form in which one encryption is computed after the other.

Theorem 3 *Consider a double-block hash function of rate 1 with round function of the form (11), where each h^i contains one encryption.*

$$\begin{cases} H_i^1 = h^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \\ H_i^2 = h^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2, H_i^1). \end{cases} \tag{11}$$

If for a fixed value of H_i^1, it takes T computations of encryption or decryption 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 $(T+3) \cdot 2^m$ computations of encryption or decryption; and a collision attack on the hash function needs at most $(T+3) \cdot 2^{m/2}$ computations of encryption or decryption.

$h(\cdot,\cdot)$	PBGV[a]	GQ-I[b]	LOKI[c]	P-DM[d]	MK[e]	MDC-2[f]	optim[g]
$(m,k)_{\rightsquigarrow 1}$	(64,64)	(64,64)	(64,64)	(64,64)	(64,56)	(64,56)	(64,64)
targ	$2^{64}_{\rightsquigarrow 2}$	$2^{64}_{\rightsquigarrow 5}$	$2^{64}_{\rightsquigarrow 5}$	$2^{64}_{\rightsquigarrow 9}$	2^{112}	$2^{81}_{\rightsquigarrow 14}$	2^{128}
f-s targ	$o(1)_{\rightsquigarrow 3}$	$2^{32}_{\rightsquigarrow 6}$	$2^{32}_{\rightsquigarrow 6}$	$2^{64}_{\rightsquigarrow 10}$	2^{112}	$2^{54}_{\rightsquigarrow 15}$	2^{64}
colli	$2^{32}_{\rightsquigarrow 3}$	2^{64}	2^{64}	$2^{32}_{\rightsquigarrow 9}$	2^{56}	2^{54}	2^{64}
sem-f-s co	$2^{32}_{\rightsquigarrow 3}$	$2^{32}_{\rightsquigarrow 7}$	$2^{32}_{\rightsquigarrow 8}$	$2^{32}_{\rightsquigarrow 11}$	2^{56}	2^{54}	2^{64}
f-s coll.	$o(1)_{\rightsquigarrow 4}$	$o(1)_{\rightsquigarrow 7}$	$o(1)_{\rightsquigarrow 8}$	$2^{32}_{\rightsquigarrow 10}$	2^{56}	$2^{27}_{\rightsquigarrow 16}$	2^{32}
rate	1	1	1	1	1/18	1/2	$?_{\rightsquigarrow 17}$

a: Proposed in [15].
b: Proposed in [16].
c: Proposed in [2].
d: Proposed in [4].
e: Merkle's scheme [10] with hash-code length 112 bits. This scheme appears to have ideal security. However, each round can "digest" only 7 bits of message.
f: See [9, 11].
g: Upper bounds on the complexities [4, 6].
$\rightsquigarrow$ 1: m: block-length, k: key-length of the underlying cipher.
$\rightsquigarrow$ 2: See [6] and last section.
$\rightsquigarrow$ 3: See [14].
$\rightsquigarrow$ 4: A free-start collision attack is no harder than a free-start target attack.
$\rightsquigarrow$ 5: New attack [7], needs a memory of size 2^{64}.
$\rightsquigarrow$ 6: See [6, 8].
$\rightsquigarrow$ 7: See [12].
$\rightsquigarrow$ 8: See [4].
$\rightsquigarrow$ 9: See last section.
$\rightsquigarrow$ 10: Provable lower bound, see [4].
$\rightsquigarrow$ 11: A semi free-start collision attack is no harder than a "usual" collision attack.
$\rightsquigarrow$ 14,15: The MDC-2 has a 128-bit hash code, but round output has length 108 bits. A free-start target attack on one (54-bit) block takes about 2^{54} computations, then use the meet-in-middle attack [8]. See also [11].
$\rightsquigarrow$ 16: Collision is achieved on one (54-bit) block.
$\rightsquigarrow$ 17: It is an open question whether there exist schemes of rate 1 and of the form (3) achieving these upper bounds. Our guess is no. See [7].

Table 1. Complexity of known attacks on double block hash functions.

3. Complexity of known attacks on $2m$-bit hash functions

We consider here some known 128-bit iterated hash functions based on two uses of an $m = 64$-bit block cipher with key length $k = 64$ or $k = 56$ in each round. All these schemes can be considered as slight modifications of the 64-bit DM-scheme hash round function. The complexities of known attacks on these hash functions are listed in Table 1. We assume that all the iterated hash functions are used with MD-strengthening and that the underlying block cipher has no known weakness (such as weak keys).

Acknowledgement The authors would like to thank Bart Prennel and the referee(s) for their valuable comments.

References

1. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
2. 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.
3. I. B. Damgaard, "A Design Principle for Hash Functions", Advances in Cryptology-CRYPTO'89, LNCS 435, pp. 416-427, Springer-Verlag, 1990.
4. W. Hohl, X. Lai, T. Meier and C. Waldvogel, "Security of Iterated Hash Function Based on Block Ciphers", Preproceedings of Crypto'93, 1993.
5. ISO/IEC DIS 10118, Information technology – Security techniques – Hash-functions, Part 2: *Hash-functions using an n-bit block cipher*, I.S.O., 1993.
6. 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.
7. L. Knudsen and X. Lai, "New attacks on a class of hash functions including the Parallel DM", submitted to EUROCRYPT'94,
8. 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.
9. S. M. Matyas, "Key Processing with Control Vectors", Journal of Cryptology, Vol. 3, No. 2, pp. 113–136, 1991.
10. R. C. Merkle, "One Way Hash Functions and DES", Advances in Cryptology - CRYPTO'89, Proceedings, LNCS 435, pp. 428-446, Springer-Verlag, 1990.
11. 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.
12. S Miyaguchi, K. Ohta and M. Iwata, "Confirmation that Some Hash Functions Are Not Collision Free", Advances in Cryptology-EUROCRYPT'90, Proceedings, LNCS 473, pp. 326-343, Springer-Verlag, Berlin, 1991.
13. M. Naor and M. Yung, "Universal One-way Hash Functions and Their Cryptographic Applications", Proc. 21 Annual ACM Symposium on Theory of Computing, Seattle, Washington, May 15-17, 1989, pp. 33-43.
14. B. Preneel, *Analysis and Design of Cryptographic Hashfunctions* , Ph.D thesis, Katholieke Universiteit Leuven, Belgium, January 1993.

15. B. Preneel, A. Bosselaers, R. Govaerts and J. Vandewalle, "Collision-free Hash-functions Based on Blockcipher Algorithms." Proceedings of 1989 International Carnahan Conference on Security Technology, pp. 203-210.

16. J. J. Quisquater and M. Girault, "2n-bit Hash Functions Using n-bit Symmetric Block Cipher Algorithms", Abstracts of EUROCRYPT'89.