Hash functions based on block ciphers: a synthetic approach

Bart Preneel*, René Govaerts, and Joos Vandewalle

Katholieke Universiteit Leuven, Laboratorium ESAT-COSIC,
Kardinaal Mercierlaan 94, B–3001 Heverlee, Belgium
bart.preneel@esat.kuleuven.ac.be

Abstract. Constructions for hash functions based on a block cipher are studied where the size of the hashcode is equal to the block length of the block cipher and where the key size is approximately equal to the block length. A general model is presented, and it is shown that this model covers 9 schemes that have appeared in the literature. Within this general model 64 possible schemes exist, and it is shown that 12 of these are secure; they can be reduced to 2 classes based on linear transformations of variables. The properties of these 12 schemes with respect to weaknesses of the underlying block cipher are studied. The same approach can be extended to study keyed hash functions (MAC's) based on block ciphers and hash functions based on modular arithmetic. Finally a new attack is presented on a scheme suggested by R. Merkle..

1 Introduction

Hash functions are functions that compress an input of arbitrary length to a string of fixed length. They are a basic building block for cryptographic applications like integrity protection based on "fingerprinting" and digital signature schemes. The cryptographic requirements that are imposed on hash functions are [4, 5, 19, 27, 28]:

one-wayness: in the sense that given X and $h(X)$, it is "hard" to find a second preimage, i.e., a message $X' \neq X$ such that $h(X') = h(X)$,

collision resistance: it should be "hard" to find a collision, i.e., two distinct arguments that hash to the same result.

The main motivation to construct a hash function based on a block cipher is the minimization of design and implementation effort. Designing secure constructions seems to be a difficult problem; this is illustrated by the large number of schemes that have been broken [23, 27].

The first constructions for hash functions based on a block cipher were one-way hash functions intended for use with the Data Encryption Standard (DES)

[*] N.F.W.O. postdoctoral researcher, sponsored by the National Fund for Scientific Research (Belgium).

D.R. Stinson (Ed.): Advances in Cryptology - CRYPTO '93, LNCS 773, pp. 368-378, 1994.

[10]. In this case the size of the hashcode (64 bits) is equal to the block length of the block cipher and the size of the key (56 bits) is approximately equal to the block size. This type of hash functions will be studied in this extended abstract. Later constructions for collision resistant hash functions were developed based on the DES; in that case it is required that the size of the hash code is at least 112...128 bits (because of the birthday attack [33]). Examples of schemes that have not been broken can be found in [20, 22]. Recent work considers the construction of hash functions if the key size is twice the block length [17], and if the key is kept constant [26]. The original constructions are still of interest since for some applications a one-way hash function is sufficient. Moreover, they can yield a collision resistant hash function if a block cipher with sufficiently large block length is available.

2 The General Model

The encryption of plaintext X with key K will be denoted with $E(K,X)$. The corresponding decryption operation applied to ciphertext C will be denoted with $D(K,C)$. Unless stated otherwise, it will be assumed that the block cipher has no weaknesses. The block length, i.e., the size of plaintext and ciphertext in bits is denoted with n and the key size in bits is denoted with k. The argument of the iterated hash function is divided into t blocks X_1 through X_t. If the total length is no multiple of n, the argument has to be padded with an unambiguous padding rule. The hash function h can subsequently be described as follows:

$$H_i = f(X_i, H_{i-1}) \qquad i = 1, 2, \ldots t\,.$$

Here f is the *round function*, H_0 is equal to the initial value (IV), that should be specified together with the scheme, and H_t is the hashcode. The *rate* R of a hash function based on a block cipher is defined as the number of encryptions to process a block of n bits.

The general model for the round function of the hash functions that will be studied in this extended abstract is depicted in Fig. 1. For simplicity it will be assumed that $k = n$. The block cipher has two inputs, namely the key input K and the plaintext input P, and one output C. One can select for the inputs one of the four values: X_i, H_{i-1}, $X_i \oplus H_{i-1}$, and a constant value V. It is also possible to modify with a feedforward FF the output C by addition modulo 2 of one of these four possibilities. This yields in total $4^3 = 64$ different schemes. In the following it will be assumed w.l.o.g. that V is equal to 0.

The exor operation was chosen because it has been used in the proposals that are generalized here; one can show that it can be replaced by any operation that is an easy-to-invert permutation of one of its inputs when the second input is fixed. The main restrictions of this model are that only 1 DES operation is used per round function and that the internal memory of the hash function is restricted to a single n-bit block.

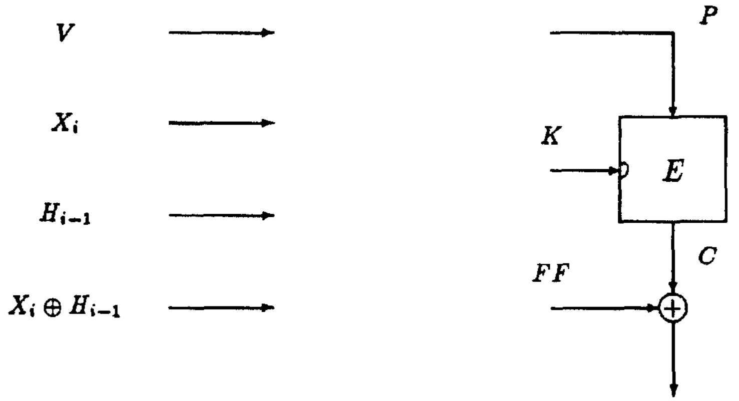

Fig. 1. Configurations where the size of the hashcode is equal to the block length. P, K, and FF can be chosen from the set $\{V, X_i, H_{i-1}, X_i \oplus H_{i-1}\}$.

3 A Taxonomy of Attacks

Five important attacks on the round function $f(X_i, H_{i-1})$ can be identified:

Direct Attack (D): given H_{i-1} and H_i, it is easy to find X_i. All schemes that are vulnerable to a direct attack can in principle be used for encryption, where the encryption of X_i is given by H_i. Of course the CBC and CFB mode belong to this class.

Permutation Attack (P): in this case H_i can be written as $H_{i-1} \oplus f'(X_i)$, where f' is a one-way function: X_i can not be recovered from H_i and H_{i-1}, but the hashcode is independent of the order of the message blocks, which means that a second preimage or collision can be found easily. Moreover one can also insert the same message block twice. These attacks are in fact trivial, as H_i depends only linearly on H_{i-1}.

Forward Attack (F): given H_{i-1}, H'_{i-1}, and X_i (note that this means that H_i is fixed), it is easy to find an X'_i such that $f(X'_i, H'_{i-1}) = f(X_i, H_{i-1}) = H_i$. In this case one can easily construct a second preimage for a given hashcode, but it is not necessarily easy to construct a preimage of a given element in the range.

Backward Attack (B): given H_i, it is easy to find a pair (X_i, H_{i-1}) such that $f(X_i, H_{i-1}) = H_i$. In this case it is trivial to find a preimage (or a second preimage) with a random initial value; a preimage (or a second preimage) can be found with a meet in the middle attack.

Fixed Point Attack (FP): find H_{i-1} and X_i such that $f(X_i, H_{i-1}) = H_{i-1}$. This attack is not very dangerous: if the hash function satisfies the one-way property, it is hard to produce a message yielding this specific value H_{i-1}.

The order of these attacks has some importance: the possibility of a direct attack means that a forward and a backward attack are also feasible, but the converse does not hold. In case of a permutation attack, one can also apply a backward attack by first selecting X_i and subsequently calculating H_{i-1}. It is easy to show that if both a forward and a backward or permutation attack are possible, a direct attack is also feasible. The proof will be given in the full paper.

4 Analysis of the 64 Schemes

Table 1 indicates which attacks are possible for each of the 64 schemes in the general model. The attacks are indicated with their first letter(s), while a "–" means that the round function f is trivially weak as the result is independent of one of the inputs. If none of these five attacks applies, a $\surd$ is put in the corresponding entry.

Table 1. Attacks on the 64 different schemes. The schemes are numbered according to the superscript.

		choice of P			
choice of FF	choice of K	X_i	H_{i-1}	$X_i \oplus H_{i-1}$	V
V	X_i	–	B^{13}	B^{25}	–
	H_{i-1}	D^{1}	–	D^{26}	–
	$X_i \oplus H_{i-1}$	B^{2}	B^{14}	F^{27}	F^{41}
	V	–	–	D^{28}	–
X_i	X_i	–	B^{15}	B^{29}	–
	H_{i-1}	$\surd^{3}$	D^{16}	$\surd^{30}$	D^{42}
	$X_i \oplus H_{i-1}$	FP^{4}	FP^{17}	B^{31}	B^{43}
	V	–	D^{18}	B^{32}	–
H_{i-1}	X_i	P^{5}	FP^{19}	FP^{33}	P^{44}
	H_{i-1}	D^{6}	–	D^{34}	–
	$X_i \oplus H_{i-1}$	FP^{7}	FP^{20}	B^{35}	B^{45}
	V	D^{8}	–	D^{36}	–
$X_i \oplus H_{i-1}$	X_i	P^{9}	FP^{21}	FP^{37}	P^{46}
	H_{i-1}	$\surd^{10}$	D^{22}	$\surd^{38}$	D^{47}
	$X_i \oplus H_{i-1}$	B^{11}	B^{23}	F^{39}	F^{48}
	V	P^{12}	D^{24}	F^{40}	D^{49}

Schemes 18 and 28 correspond to the CFB mode respectively the CBC mode for encryption as specified in [11, 14]; the fact that these modes are useful for keyed hash functions but not sufficient to construct one-way hash functions was pointed out by S. Akl in [1]. This is also connected to the fact that the integrity protection offered by these modes is limited. Scheme 13 was proposed by Rabin in 1978 [28]: however, R. Merkle has shown that a backward attack is possible, from which it follows that one can find a preimage with a meet in the middle attack.

The next proposal was scheme 14, attributed to W. Bitzer in [7, 9]. R. Winternitz [31] has shown that the meet in the middle attack by R. Merkle is applicable to this scheme as well. He also has pointed out that schemes 13 and 14 are vulnerable to a weak key attack: for a weak key K_w the DES is an involution which means that $E(K_w, E(K_w, X)) = X, \forall X$. Inserting twice a weak key as a message block will leave the hashcode unchanged in all the schemes.

The first secure scheme (scheme 3) was proposed by S. Matyas, C. Meyer, and J. Oseas in [18]. Its 'dual', scheme 19, is attributed to D. Davies in [31, 32], and to C. Meyer by D. Davies in [8]. D. Davies has confirmed in a personal communication to the authors that he did not propose the scheme. Nevertheless, this scheme is widely known as the Davies-Meyer scheme (see e.g., [23]). The fact that this scheme is vulnerable to a fixed point attack was pointed out in [25]. Scheme 10 was proposed by the authors and studied in [30]. It appeared independently in [24] as a mode for N-hash. In 1990 the same scheme was proposed by Japan to ISO/IEC [15]. The international standard ISO/IEC 10118 Part 2 [16] specifying hash functions based on block ciphers contains scheme 3. Scheme 20 was proposed as a mode of use for the block cipher LOKI in [3]. Finally it should be remarked that scheme 40 (together with its vulnerability to a forward attack) was described in [18].

It is the merit of this approach that all schemes based on the general model have been classified once and for all. The second advantage is that the properties of the 12 'secure' schemes can now be compared. First a further classification will be made based on an equivalence transformation.

5 Equivalence Classes

This large number of schemes can be classified further by considering linear transformations of the inputs. A class of schemes that is derived from a single scheme by linear transformation of variables will be called an equivalence class.

- In 7 equivalence classes the round function depends on two independent inputs (X_i and H_{i-1}), and 6 transformations are possible, as there are 6 invertible 2×2 matrices over $GF(2)$. It can be shown that in 2 cases the round function is secure or is vulnerable to a fixed point attack, and in 5 cases the round function is vulnerable to a direct attack, a permutation attack, or a backward attack.
- In 7 equivalence classes the round function depends on a single independent input. Hence one has three possible inputs, namely X_i, H_{i-1}, and $X_i \oplus H_{i-1}$, corresponding to the 3 nonzero vectors of length 2 over $GF(2)$. If the round function depends on the sum of the two inputs, it is not trivially weak. However, it is vulnerable to a direct attack (2 cases out of 7) or to a forward attack (5 cases out of 7).
- In 1 equivalence class the round function is simply constant.

Table 2 describes the equivalence classes. A further classification is made based on the number CI of constants in the choices. To characterize a class, a relation is given between plaintext P, key K, and feedforward FF.

Table 2. Overview of the 15 variants, sorted according to the number CI of constant inputs.

CI	characterization	class size	-	D	P	B	F	FP	√
0	$FF = P, (P \neq K)$	6						4	2
	$FF = P \oplus K, (P \neq K)$	6						4	2
	$FF = K, (P \neq K)$	6		2		4			
	$P = K, (FF \neq P)$	6		2	2	2			
	$FF = P = K$	3	2				1		
1	$FF = V, (P \neq K)$	6		2		4			
	$P = V, (FF \neq K)$	6		2	2	2			
	$K = V, (FF \neq P)$	6		4	1	1			
	$FF = V, (P = K)$	3	2				1		
	$P = V, (FF = K)$	3	2				1		
	$K = V, (P = FF)$	3	2				1		
2	$FF = P = V$	3	2				1		
	$FF = K = V$	3	2	1					
	$P = K = V$	3	2	1					
3	$FF = P = K = V$	1	1						
Total		64	15	14	5	13	5	8	4

One can conclude that only 4 schemes of the 64 are secure, and that 8 insecure schemes are only vulnerable to a fixed-point attack. These 12 schemes are listed (and re-numbered) in Table 3 and graphically presented in Fig. 2. The roles of X_i and H_{i-1} in the input can be arbitrarily chosen, and the dotted arrow is optional (if it is included, the key is added modulo 2 to the ciphertext). For the dash line, there are three possibilities: it can be omitted or it can point from key to plaintext or from plaintext to key. There are two equivalence classes that are secure, and their simplest representatives are the scheme by Matyas et al. (number 1) and the scheme by Miyaguchi and the authors (number 3). For each of these schemes it is possible to write a 'security proof' based on a black box model of the encryption algorithm, as was done for the Davies-Meyer scheme (number 5) in [32]. The basic idea is that finding a (pseudo)-preimage for a given hash value is at least as hard as solving the equation $H_i = f(X_i, H_{i-1})$ for a given value of H_i. The expected number of evaluations of $f()$ is shown to be 2^{n-1}.

In the full paper these 12 schemes will be compared in more detail based on their vulnerability to fixed point attacks, to attacks based on weaknesses of the underlying block cipher (in this case the DES), and to differential attacks [2]. Also their efficiency will be compared. The main results are summarized in Table 4: column 5 indicates what the output of the round function is if the key is a weak key and the plaintext is one of the corresponding fixed points, column 6 has a $\surd$ if one can exploit the complementation property, and the last column indicates which variables have to be modified if a differential attack

Table 3. A list of the 12 secure schemes for a one-way hash function based on a block cipher and a feedforward.

no.	function expression
1	$E(H_{i-1}, X_i) \oplus X_i$
2	$E(H_{i-1}, X_i \oplus H_{i-1}) \oplus X_i \oplus H_{i-1}$
3	$E(H_{i-1}, X_i) \oplus X_i \oplus H_{i-1}$
4	$E(H_{i-1}, X_i \oplus H_{i-1}) \oplus X_i$
5	$E(X_i, H_{i-1}) \oplus H_{i-1}$
6	$E(X_i, X_i \oplus H_{i-1}) \oplus X_i \oplus H_{i-1}$
7	$E(X_i, H_{i-1}) \oplus X_i \oplus H_{i-1}$
8	$E(X_i, X_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$E(X_i \oplus H_{i-1}, X_i) \oplus X_i$
10	$E(X_i \oplus H_{i-1}, H_{i-1}) \oplus H_{i-1}$
11	$E(X_i \oplus H_{i-1}, X_i) \oplus H_{i-1}$
12	$E(X_i \oplus H_{i-1}, H_{i-1}) \oplus X_i$

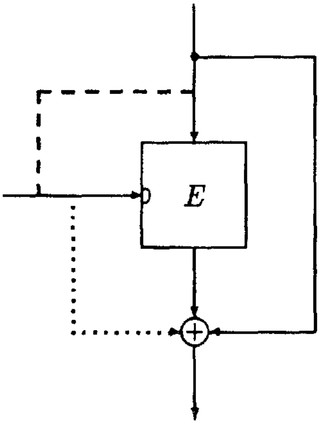

Fig. 2. Secure configuration for a one-way hash function based on an block cipher and a feedforward.

with a fixed key is used. (One could also think of a dual differential attack, where the plaintext is fixed and a given key difference is applied; this has not been considered here.)

If $k \neq n$, the question arises whether it is possible to use variables X_i and H_{i-1} of length $max(n, k)$, in order to maximize the security level. The idea is that bits that are not used in the key or as plaintext might influence the output through some exors. The following proposition shows that this is not possible

Table 4. Properties of the 12 secure schemes: fixed points, properties if the DES is used as the underlying block cipher, and variables to be modified in case of a differential attack.

no.	fixed points		properties if E = DES			differential
	X_i	H_{i-1}	rate R	K_w	compl.	attack
1	–	–	1	0	√	X_i
2	–	–	1	0	√	X_i
3	–	–	1	K_w	-	X_i
4	–	–	1	K_w	-	X_i
5	K	$D(K,0)$	n/k	0	√	H_{i-1}
6	K	$D(K,K) \oplus K$	n/k	0	√	H_{i-1}
7	K	$D(K,K)$	n/k	K_w	-	H_{i-1}
8	K	$D(K,0) \oplus K$	n/k	K_w	-	H_{i-1}
9	$D(K,K)$	$D(K,K) \oplus K$	1	0	√	X_i, H_{i-1}
10	$D(K,0) \oplus K$	$D(K,0)$	n/k	0	√	H_{i-1}
11	$D(K,0)$	$D(K,0) \oplus K$	1	K_w	-	X, H_{i-1}
12	$D(K,K) \oplus K$	$D(K,K)$	n/k	K_w	-	X_i, H_{i-1}

for the hash functions which follow our general model; it will be proven in the full paper.

Proposition 1 *The security level of the one-way hash function is determined by the minimum of k and n, with k the size of the key and n the block length.*

6 Merkle's Improvement to Rabin's Scheme

In order to avoid the backward attack in case of Rabin's scheme (Scheme 13), R. Merkle proposed to encrypt the message in CBC or CFB mode (with a random non-secret key K and initial value IV) before applying the hash function [6]. This implies a reduced performance: the rate equals 2. The idea is to introduce a dependency between the blocks that enter the hash function. It will be shown how the meet in the middle attack can be modified to take into account this extension.

- Generate a set of r messages for which the last ciphertext block of the CBC encryption with initial value IV and key K is equal to IV' ; this can be done easily with an appropriate selection of the last plaintext block (or with a meet in the middle attack).
- Generate a second set of r messages and encrypt these messages in CBC mode with initial value IV' and key K.
- As the two message parts are now independent, one can use the set of two 'encrypted' messages in a simple meet in the middle attack.

This shows that finding a preimage requires only $O(2^{n/2})$ encryptions.

7 Extensions

The same approach can be applied to keyed hash functions (or Message Authentication Code) based on a block cipher. The main result in this case is that the round function

$$f = E(K, X_i \oplus H_{i-1}) \oplus X_i$$

is preferable over the CBC or CFB mode that are specified in most standards (e.g., [13]). Note that an additional protection is required at the end. More details will be given in the full paper.

The design of hash functions based on modular squaring can also benefit from this synthetic approach. Since there is no key, only 16 cases have to be studied. Most hash functions in this class base their security on the fact that taking modular square roots is hard for someone who does not know the factorization of the modulus. Seven of these schemes are trivially weak, and 4 (from the remaining 9) have appeared in the literature. The main conclusion is that the round function

$$f = (X_i \oplus H_{i-1})^2 \bmod N \ \oplus X_i$$

is the most promising. In order to obtain a secure hash function a redundancy scheme has to be specified. The redundancy is necessary to thwart the exploitation of the algebraic structure like fixed points of the modular squaring and 'small' numbers for which no reduction occurs [12, 27].

Finally it should be noted that a similar structure has been employed in dedicated hash functions like the MD4 family [29] and Snefru [21]. The analysis with respect to differential attacks and fixed points can also be transferred to these schemes.

8 Conclusion

The construction of cryptographic hash functions based on block ciphers is apparently a difficult problem. In this extended abstract a general treatment has been developed for the simplest case, namely size of hashcode equal to block length and key size. This approach allows to identify the secure schemes and to compare these with respect to several criteria. It is also useful to study keyed hash functions, hash functions based on modular arithmetic, and dedicated hash functions.

References

1. S.G. Akl, "On the security of compressed encodings," *Advances in Cryptology, Proc. Crypto'83*, D. Chaum, Ed., Plenum Press, New York, 1984, pp. 209–230.
2. E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, Vol. 4, No. 1, 1991, pp. 3–72.
3. L. Brown, J. Pieprzyk, and J. Seberry, "LOKI – a cryptographic primitive for authentication and secrecy applications," *Advances in Cryptology, Proc. Auscrypt'90, LNCS 453*, J. Seberry and J. Pieprzyk, Eds., Springer-Verlag, 1990, pp. 229–236.

4. I.B. Damgård, "Collision free hash functions and public key signature schemes," *Advances in Cryptology, Proc. Eurocrypt'87, LNCS 304*, D. Chaum and W.L. Price, Eds., Springer-Verlag, 1988, pp. 203–216.
5. I.B. Damgård, "A design principle for hash functions," *Advances in Cryptology, Proc. Crypto'89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 416–427.
6. D. Davies and W. L. Price, "The application of digital signatures based on public key cryptosystems," *NPL Report* DNACS 39/80, December 1980.
7. D. Davies, "Applying the RSA digital signature to electronic mail," *IEEE Computer*, Vol. 16, February 1983, pp. 55–62.
8. D. Davies and W. L. Price, "Digital signatures, an update," *Proc. 5th International Conference on Computer Communication*, October 1984, pp. 845–849.
9. D. Denning, "Digital signatures with RSA and other public-key cryptosystems," *Communications ACM*, Vol. 27, April 1984, pp. 388–392.
10. FIPS 46, *"Data Encryption Standard,"* Federal Information Processing Standard, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.
11. FIPS 81, *"DES Modes of operation,"* Federal Information Processing Standard, National Bureau of Standards, US Department of Commerce, Washington D.C., December 1980.
12. M. Girault, "Hash-functions using modulo-n operations," *Advances in Cryptology, Proc. Eurocrypt'87, LNCS 304*, D. Chaum and W.L. Price, Eds., Springer-Verlag, 1988, pp. 217–226.
13. ISO/IEC 9797, *"Information technology - Data cryptographic techniques - Data integrity mechanisms using a cryptographic check function employing a block cipher algorithm,"* 1993.
14. ISO/IEC 10116, *"Information technology - Security techniques - Modes of operation of an n-bit block cipher algorithm,"* 1991.
15. *"Hash functions using a pseudo random algorithm,"* ISO-IEC/JTC1/SC27/WG2 N98, Japanese contribution, 1991.
16. ISO/IEC 10118, *"Information technology - Security techniques - Hash-functions - Part 1: General and Part 2: Hash-functions using an n-bit block cipher algorithm,"* 1993.
17. X. Lai and J.L. Massey "Hash functions based on block ciphers," *Advances in Cryptology, Proc. Eurocrypt'92, LNCS 658*, R.A. Rueppel, Ed., Springer-Verlag, 1993, pp. 55–70.
18. S.M. Matyas, C.H. Meyer, and J. Oseas, "Generating strong one-way functions with cryptographic algorithm," *IBM Techn. Disclosure Bull.*, Vol. 27, No. 10A, 1985, pp. 5658–5659.
19. R. Merkle, *"Secrecy, Authentication, and Public Key Systems,"* UMI Research Press, 1979.
20. R. Merkle, "One way hash functions and DES," *Advances in Cryptology, Proc. Crypto'89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 428–446.
21. R. Merkle, "A fast software one-way hash function," *Journal of Cryptology*, Vol. 3, No. 1, 1990, pp. 43–58.
22. C.H. Meyer and M. Schilling, "Secure program load with Manipulation Detection Code," *Proc. Securicom 1988*, pp. 111–130.
23. C. Mitchell, F. Piper, and P. Wild, "Digital signatures," in *"Contemporary Cryptology: The Science of Information Integrity,"* G.J. Simmons, Ed., IEEE Press, 1991, pp. 325–378.

24. S. Miyaguchi, M. Iwata, and K. Ohta, "New 128-bit hash function," *Proc. 4th International Joint Workshop on Computer Communications*, Tokyo, Japan, July 13–15, 1989, pp. 279–288.
25. S. Miyaguchi, K. Ohta, and M. Iwata, "Confirmation that some hash functions are not collision free," *Advances in Cryptology, Proc. Eurocrypt'90, LNCS 473*, I.B. Damgård, Ed., Springer-Verlag, 1991, pp. 326–343.
26. B. Preneel, R. Govaerts, and J. Vandewalle, "On the power of memory in the design of collision resistant hash functions," *Advances in Cryptology, Proc. Auscrypt'92, LNCS 718*, J. Seberry and Y. Zheng, Eds., Springer-Verlag, 1993, pp. 105–121.
27. B. Preneel, *"Cryptographic hash functions,"* Kluwer Academic Publishers, 1994.
28. M.O. Rabin, "Digitalized signatures," in *"Foundations of Secure Computation,"* R. Lipton and R. DeMillo, Eds., Academic Press, New York, 1978, pp. 155–166.
29. R.L. Rivest, "The MD4 message digest algorithm," *Advances in Cryptology, Proc. Crypto'90, LNCS 537*, S. Vanstone, Ed., Springer-Verlag, 1991, pp. 303–311.
30. K. Van Espen and J. Van Mieghem, *"Evaluatie en Implementatie van Authentiseringsalgoritmen (Evaluation and Implementation of Authentication Algorithms – in Dutch),"* ESAT Laboratorium, Katholieke Universiteit Leuven, Thesis grad. eng., 1989.
31. R.S. Winternitz, "Producing a one-way hash function from DES," *Advances in Cryptology, Proc. Crypto'83*, D. Chaum, Ed., Plenum Press, New York, 1984, pp. 203–207.
32. R.S. Winternitz, "A secure one-way hash function built from DES," *Proc. IEEE Symposium on Information Security and Privacy 1984*, 1984, pp. 88–90.
33. G. Yuval, "How to swindle Rabin," *Cryptologia*, Vol. 3, 1979, pp. 187–189.