Making Hash Functions from Block Ciphers Secure and Efficient by Using Convolutional Codes

Toru Inoue[1] and Kouichi Sakurai[2]

[1] Advanced Mobile Telecommunications Security
Technology Research Laborarities Co., Ltd
BENEX S3, 3-20-8, Shin-Yokohama, 222-0033 JAPAN
t-inoue@amsl.co.jp
[2] Kyushu University, Dept. of Computer Science
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan
sakurai@csce.kyushu-u.ac.jp

Abstract. We improve Knudsen-Preneel's constructions for cryptographic hash functions based on block ciphers with error correcting codes. We first modify to extend original constructions, which are effective only for non-binary codes, to the case with binary codes (e.g. BCH codes). We also revise the original method by introducing convolutional codes, whereas the previous adapts only block codes. This reduces the circuit complexity of the hardware-implementation $1/N$ times in terms of the number of (Davies-Meyer's) module functions than that based block error correcting codes.

Key words: *hash functions, block ciphers, error correcting codes, BCH codes, convolutional codes, Collision resistant compression*

1 Introduction

Hash function. Hash functions are important cryptographical techniques for making signature, or building public key crypto-systems. Currently used hash functions include SHA, SHA-1, MD-4, MD-5, etc. Although methods which use particular algorithms for these hash functions are reasonably speedy, some of the methods are known to vulnerable to certain types of attacks[Dob96a,Dob96b]. Meanwhile, a Davies-Meyer method is known as a method of obtaining a hash function through a repeated use of a block cipher as a compression function. An advantage of using a block cipher to a hash function is that if the security of the block cipher is guaranteed, the hash function using the block cipher is secure[KP96,KP97]. In addition, the security will not be lost even by an expansion of a single hash mode into a multiple hash mode[KP97]. In this paper, we assume to use a block cipher whose security is guaranteed. Moreover an error correction coding for data protection is easily adopted using same error correction encoders while encrypting data. We also assume the following hypotheses:

H. Imai, Y. Zheng (Eds.): PKC 2000, LNCS 1751, pp. 391–404, 2000.

1. Hash round function is secure.
2. A short cut which breaks the single hash mode does not exist.
3. If the security of the block cipher is guaranteed, the security of the hash function using the block cipher is also guaranteed.
4. A document is sufficiently long.

Despite this, if we use an ordinary 64-bit block cipher as it is, a collision (or a birthday attack) occurs for every 2^{32} trials on the average, which is not highly secure anymore in the present situation regarding ciphers.

Knudsen-Preneel's approach and the challenging problems. The constuction proposed by Knudsen and Preneel is known as a method which uses error correcting codes with the multiple Davies-Meyer functions for enhancing a collision resistance[KP96,KP97]. Their method requires to use quarternary (n, k, d) linear codes, where the symbol n denotes a code length, the symbol k denotes an information symbol number, and the symbol d denotes a minimum distance, and improves a collision resistance from $2^{m/2}$ to $2^{(d-1)m/2}$. However, Knudsen and Preneel's method requires to use only non-binary codes in the construction. For design flexibility we want to use binary codes, too. Another problem is when we use block codes, as a tendency, the code length becomes longer to increase the number of error correcting capability, the computation accordingly becomes more complex. In addition, many Davies-Meyer modules are required because of a long code length, then it is necessary to prepare a number of apparatuses for realizing Davies-Meyer functions. Thus, there is a room for improvement of Knudsen-Preneel's approach with respect to the efficiency for hardware constructions.

Our Contribution

Using binary codes. We first try a similar construction as Knudsen-Preneel by using binary codes, such as BCH codes. The direct method by Knudsen-Preneel fails in the case of binary codes. Then, we revise Knudsen-Preneel's construction for adapting the use of binary codes. However, this revision with binary codes are not so efficient as the Knudsen-Preneel's original with non-binary MDS codes: our hash rate becomes very low. Next, we devise efficiency by error correction encoding only message vectors, while our previous method encodes not only message vectors but also key vectors. Though it requires stronger (but still reasonable) assumption than that of Knudsen-Preneel, our second construction achieve good hash rate.

Using convolutional codes. We try to resolve the second problem by using convolutional codes. Our proposed method requires to select the number of multiple Davies-Meyer functions, to the same as the sub-block length n_0 of a convolutional code and to enter inputs in N time units where N is a constraint length, to thereby reduce the size of a Davies-Meyer function down to the sub-block length n_0 of a convolutional code from a code length n which is a code length in a case

where a block code is used. This method using convolutional codes reduces hardware $1/N$ in terms of the number of Davies-Meyer functions than where block error correcting codes are used under the same functional conditions. For example, the construction using (15,7,5)BCH codes require 15 Davies-Meyer functions. However, the construction using convolutional codes such as$(3, 2, 5; N = 14)$ CSOC, requires only three Davies-Meyer functions, namely, convolutional codes reduce the number of Davies-Meyer functions by the factor of $1/5$.

2 Hash Function

In a telecommunication system in which messages, data and the like are encoded and transmitted as cryptograms to protect the confidentiality, a hash function, i.e., a compression function for compressing and signing a message, is used. To compress a document of an optional length into a certain predetermined length, cryptographic hash function is used. For example, in order to sign using the DSS (Digital Signature Standard), a document of an optional length is converted using a hash function into a hashed value of a 160-bit block once, and a signature of 320 bits, for instance, is added to the 160-bit block hashed value. The hash function needs be devised so as to obviate a collision. A collision is an event that $h(x) = h(x')$ holds when $x \neq x'$. According to the definition of a resistance of a hash function against a collision, there are a weak resistance and a strong resistance. A weak collision resistance is called a preimage resistance or a 2nd-preimage resistance.

A preimage resistance expresses to what extent it is difficult to find x' which converts into $h(x')$ in relation to an inputted hashed value $H = h(x)$. A 2nd-preimage resistance expresses to what extent it is difficult to find a second input x' which satisfies hashed value $H = h(x) = h(x')$ in relation to the input x. That is, when we have a document x and a corresponding hashed value $h(x)$, if it is difficult to find a document x'which is converted into the same hashed value $h(x)$ whatever the document x is, and further, if it requires W trials on the average to find such a document x', a 2nd-preimage resistance of the hash function h is W. In this paper, a preimage and a 2nd-preimage will not be distinguished from each other, but instead, treated equally.

A strong collision resistance is called a collision resistance or a resistance against a birthday attack. In short, a strong collision resistance expresses to what extent it is difficult to find any input pair $(x, x'; x \neq x')$ which converts into $h(x) = h(x')$. More precisely, if W trials on the average are necessary to find a pair of different documents having the same hashed value, a resistance against a birthday attack of the hash function is W. A collision resistance normally means a strong resistance. A hash rate of a hash function based on an $m-$bit block cipher is defined by the number of m-bit message blocks which are processed during one encryption or decryption. The method using block cipher is called a Davies-Meyer method [MMO85,DP84]. An encryption algorithm for an $m-$bit block cipher is denoted at the symbol $E_K(x)$, and its $m-$bit key is denoted at the symbol K. The compression function is called a Davies-Meyer function.

We consider iterated hash function based on an easily computable compression function $h(\cdot,\cdot)$ originated from two binary sequences of lengths m and l to a binary sequence of length m. The message M is split into blocks M_i of l bits, $M = (M_1, M_2, \ldots, M_n)$. If the length of M is not a multiple of l, M is padded by using an deterministic padding rule. The hash value H_n of length m is obtained by computing iteratively,

$$H_i = h(H_{i-1}, M_i) \quad i = 1, 2, \ldots, t \quad (1)$$

where H_0 is an initial value, denoted by IV, namely $H_0 = IV$. The function $h(\cdot,\cdot)$ is called hash round function. Hash result

$$Hash(IV, M) = H_t \quad (2)$$

is obtained by repeating calculation (1). To relate the security of $Hash(\cdot)$ to that of $h(\cdot,\cdot)$, we needs to append an additional block at the end of the input string concerning its length, as MD-strengthening leading to the following result [Dam89,Me89].

Theorem-MD: Let $Hash(\cdot)$ be an iterated hash function appended MD-strengthening. Then preimage and collision attacks on $Hash(\cdot,\cdot)$ have roughly the same complexity as the corresponding attack on $h(\cdot,\cdot)$ [KP97]. In practical applications, the IV of a hash function is fixed in the specifications. This leads to a higher security level, so Theorem-MD gives a lower bound on the security of $Hash(IV,\cdot)$ [KP97].

ASSUMPTION 1 [KP96,KP97]: Encrypting (of the $m-$bit block) about $2^{m/2}$ times is necessary to find a collision to h as far as a secure block cipher is used, and encrypting about 2^m times is necessary to find a preimage to h.

The message M_i, the hashed value H_i and the immediately previous hashed value H_{i-1}hold:

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

where the symbol $\oplus$ denotes modulo-2 addition and the symbol H_i is an accumulated sum of hashed values and a message at a time i from the beginning of the document to a time $i-1$.

Definition 1 (Multiple Davies-Meyer function). *An $m-$bit block cipher which uses an $am-$bit key K which satisfies $a > 0$. Keys have different values from each other, so that $h_1, h_2, \ldots, h_n$ are Davies-Meyer functions which are different from each other. A multiple Davies-Meyer function affine transforms an $m-$bit message input and maps the affine transformed input to n pairs (X_i, Y_i) which will be used as inputs. Outputs are concatenation of $h_1, h_2, \ldots, h_n$. At the time of a collision or a preimage, if a pair (X_i, Y_i) which forms an input block is different from the original pair, $h(X_i, Y_i)$ is active. Conversely, two functions $h(X_i, Y_i)$ and $h(X_j, Y_j)$ are independently attackable, if a variable parameter (X_j, Y_j) of the function h_j does not change despite a change in a variable parameter (X_i, Y_i) of the function h_i.*

ASSUMPTION 2 [KP96,KP97]: Assume we found a collision or preimage to multiple Davies-Meyer compression functions. Consider P expresses the number of active functions and $P - v$ expresses the maximum number of independently attackable functions. At least $2^{vm/2}$ or 2^{vm}encryptions are respectively necessary for a collision or preimage to occur.

3 Construction Method Using Error Correcting Codes

3.1 Construction Method of Knudsen L. and Preneel B.

We will now describe a construction method proposed by Knudsen L. and Preneel B. which uses error correcting codes.

Theorem 1. *Assume input blocks are encoded using*(n, k, d) *codes on* $GF(2^{a+1})$ *satisfying* $(a + 1)k > n$ *but* $a \geq 1$ *and* $m >> \ log_2 n$. *In this condition, as far as the Assumption 2 holds, at least* $2^{(d-1)m/2}$*encryptions are necessary to find a collision to a compression function and at least* $2^{(d-1)m}$ *encryptions are necessary to find a preimage to the compression function[KP97]. This hash function requires an internal memory of* nm *bits, and a hash rate is* $(a + 1)(k/n) - 1$.

That is, if we use error correcting codes having a distance of 3, we can improve the security level for collision attacks from $2^{m/2}$ to 2^m or the security level for preimage attacks from 2^m to 2^{2m} and easily construct a secure hash function.

3.2 Construction Method Using BCH Codes

Let us apply Theorem 1 to binary BCH codes. Although $a \geq 1$ is assumed, we consider binary BCH codes inserting $a = 0$. This constructs (n, k, d) codes over $GF(2)$, which leads $k > n$. This is impossible because $k < n$ is required in order to construct error correction codes. To construct error correction codes if we take two BCH codewords, then $a = 0$ has no significance, and thus we obtain the construction of $2k > n$. Therefore, modification allotting the each bit of input block, to the k elements of BCH codeword, enables the new construction. Instead of assigning symbol elements of Galois fields from two $m-$bit inputs which are basic structures of the Davies-Meyer method, the method requires to assign two codewords of a binary code to the inputs so as to allow use of binary codes. These are one for previous hashed values, and the other for message block, then we obtain an $n \times 2$ input array. We allot one element of one binary codeword to one bit of one $m-$bit block, respectively, not allotting symbols of the Galois field element.

Theorem 2 (Theorem 1 extended). *Assume input blocks are encoded using* (n, k, d)*codes on* $GF(2)$*satisfying* $k < n$ *and* $2k > n$, *and* $m >> \ log_2 n$. *In this condition, as far as the Assumption 2 holds, at least* $2^{(d-1)m/2}$ *encryptions are necessary to find a collision to a compression function and at least* $2^{(d-1)m}$ *encryptions are necessary to find a preimage to the compression function[KP97]. This hash function requires an internal memory of* nm *bits, and a hash rate is* $(a + 1)(k/n) - 1$.

(Proof): At least d Davies-Meyer functions are active and also at least first k bits among n bits of functions h_i are independent each other. At least $d-1$ bits among the last $n-k$ bits are dependent on the first k input. The condition of $n-k \geq d-1$ gives Theorem 2 naturally. **(QED)**.

However, binary codes are not so efficient than non-binary codes are, because there is no non-trivial MDS codes for binary codes, therefore the hash rate becomes very low. We can construct using two (7,4,3) Hamming codes in order to improve collision resistance, for example, from $2^{m/2}$ to 2^m, although hash rate is down to 1/7. We devise efficiency by error correction encoding only message vectors.

4 Our Proposed Methods

A BCH codeword in the first column which consists of n pieces of $m-$blocks which are constructed only by hashed values of one unit time ago, encodes only a hashed value of a previous time point, which prohibits to enter information of a new message. Hence, this column will not be attacked nor have to be encoded against attacks. Only n pieces of $m-$blocks in the second column must be encoded and protected against attacks. We propose, in this paper again, not to encode the first column which consists of blocks of one unit time ago alone which are fed back but to encode only the second column. This realizes an efficient construction which allows to input more new messages instead.

4.1 Construction Method Using Block Codes

Now, let's assume a collision or preimage has occurred to n concatenated $m-$bit hashed values.

Theorem 3. *Assume error correction encoders to encode k message blocks into n $m-$bit blocks, then input n previous hashed values and error correction codeword making $n \times 2$ $m-$bit blocks to n multiple Davies-Mayer functions. Assume a collision or preimage among n consecutive blocks occur, and if P Davies-Meyer functions are active, then $P-(d-1)$ message inputs $Y_i s$ are independent, but $d-1$ message $Y_i s$ are depend. In this condition, as far as the Assumption 2 holds, at least $2^{(d-1)m/2}$ encryptions are necessary to find a collision to a compression function and at least $2^{(d-1)m}$ encryptions are necessary to find a preimage to the compression function.*

(Proof): At least d Davies-Meyer functions are active and also at least first k bits among n bits of vector $Y_i s$ are independent each other. At least $d-1$ bits among the last $n-k$ bits are dependent from the first k input. The condition of $n-k \geq d-1$ gives Theorem 3 naturally. **(QED)**.

The discussion mentioned above, holds as long as the state is continuous and the input vectors $X_i s$ are regarded as random oracles. However, the initial state

gives arbitrary values for key and message too. Therefore the initial values must be treated based on Knudsen and Preneel.

EXTENED ASSUMPTION 2

Initial state Suppose a collision or preimage is found for multiple Davies-Meyer compression functions. Let P be the number of active functions, and $P - v$ be the number of attackable functions, and $h_1, h_2, \ldots, h_n$ be Davies-Meyer functions which collide, simultaneously. Among these functions, the relationship $h_i(X_i, Y_i) = E_{X_i}(Y_i) \oplus Y_i$ holds. These are obtained by fixing $\lceil \log_2 n \rceil$ key bits to different values. The compression functions of a multiple Davies-Meyer scheme takes $2km-$bit input blocks, which are expanded by an affine mapping to the n pairs (X_i, Y_i). The output of the compression functions depend on all $2k$ input blocks, thus, the matrix of the affine mapping has the rank $2k$.

Steady state In the steady state, except initial state the compression function of multiple Davies- Meyer scheme take $km-$bit input blocks, which are expanded by an affine mapping to the pairs (X_i, Y_i) . The output of the compression functions depend on all k input blocks, therefore the matrix of the affine mapping has the rank k. Suppose a collision or preimage for the compression function of a multiple Davies-Meyer scheme is found, simultaneously for $h_1, h_2, \ldots, h_n$. Assume the two different inputs of $\{Z_i\}$ and $\{Z_i'\}$ give all the same outputs of n blocks. Let P functions group be $\{h_i\}$ which $Z_i \neq Z_i'$ holds under the condition of $P \leq n$. The matrix of functions h_j has the rank $P - v$.

The simultaneous collision requires $2^{mv/2}$ encryptions and the simultaneous preimage requires 2^{mv} encryptions concerning to P functions h_j. Consequently, the error-correction encoding of the each k bits within $nm-$bit blocks, gives n bits of initial value X_0, and the remaining $n - k$ bits are alloted in initial input vector Y_0.

Namely, the initial message bits are $r = 2k - n$ bits. Except initial condition, message bits are given by $r = k$. As a result, $d - 1$ vectors Y_j at the last $n - k$ vectors are required to depend on the first k vectors Y_js, so in the steady state, only the vector Y_js are required error correction encoding.

Corollary 1. *Let us consider (n, k, d) binary code. The previous hashed values $H_{i-1}s$ at a time $i-1$ are fed back to $nm-$bit blocks of the first column at a time i. The number of message blocks becomes k. The hash rate is given by k/n.*

This rate is the same rate as a code rate that an error correction code originally has. Therefore hash rate is greatly improved. As operational notice nm bit initial key values are required. In the new construction method, one codewords of binary codes allows to input to $m-$bit blocks which are one parts of n input pairs. The construction will be now described using $(7, 4, 3)$ Hamming codes for the simplicity of description. A parity matrix is as follows:

$$H = (\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0)$$

$$\begin{array}{r} 1\,1\,1\,0\,1\,0\,0 \\ =0\,1\,1\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1 \end{array}$$

A primitive polynominal is $X^3 + X + 1$.

$$H_1 = h_1(G_1, M_1)$$

$$H_2 = h_2(G_2, M_2)$$

$$H_3 = h_3(G_3, M_3)$$

$$H_4 = h_4(G_4, M_4)$$

$$H_5 = h_5(G_5, r_3)$$

$$H_6 = h_6(G_6, r_2)$$

$$H_7 = h_7(G_7, r_1),$$

where the H_i's are hash values denoted by eq. (1) and the symbols r_3, r_2 and r_1 are blocks of check bits of Hamming codes. The blocks r_3, r_2 and r_1 are expressed as:

$$r_3 = M_1 \oplus M_2 \oplus M_3$$

$$r_2 = M_2 \oplus M_3 \oplus M_4$$

$$r_1 = M_1 \oplus M_2 \oplus M_4$$

where the symbols G_i, M_i and r_i denote $m-$bit blocks. The $m-$bit block G_i expresses a hashed value at a previous time point. The $m-$bit block M_i is a message block. The $m-$bit block r_i expresses the check bits of the Hamming codes. Although we have used Hamming codes for the convenience of description, let's now consider double error correcting BCH codes for the sake of the larger distance. Use of double error correcting BCH codes makes it possible to construct the more secure hash function. Assume (31, 21, 5) BCH codes are shortened to (30, 20, 5) BCH codes. Since $r = 20$, a hash rate is $20/30 = 2/3$. For comparison, Table 1 shows results of an example where binary codes are used and also an example according to Knudsen L. and Preneel B.

As described above, according to this paper, use of binary codes omits computation of Galois fields and allows to obtain a hashed value which is highly secure.

4.2 Construction Method Using Convolutional Codes

When we use block codes, as a tendency, a code length becomes longer to increase the number of error correcting capability and computation accordingly becomes complex. In addition, since many Davies-Meyer units are necessary because of a long code length, it is necessary to prepare a number of apparatuses for realizing Davies-Meyer scheme. We will now introduce a construction method which

Table 1. A method using binary codes vs. a conventional method

	field	t	code	rate	collision	memory
ours	$GF(2)$	1	(7,4,3)	4/7=0.571	2^m	7m
	$GF(2)$	1	(15,11,4)	11/15=0.733	2^m	15m
	$GF(2)$	2	(15,7,5)	7/15=0.467	2^{2m}	15m
	$GF(2)$	2	(25,15,5)	15/25=0.60	2^{2m}	25m
	$GF(2)$	2	(30,20,5)	20/30=0.666	2^{2m}	30m
	$GF(2)$	2	(62,54,5)	54/62=0.871	2^{2m}	62m
Knudsen	$GF(2^2)$	1	(5,3,3)	1/5=0.20	2^m	5m
& Preneel	$GF(2^4)$	1	(6,4,3)	1/4=0.25	2^m	6m
	$GF(2^2)$	1	(8,5,3)	1/4=0.25	2^m	8m

requires to select the number of multiple Davies-Meyer functions, to the same as the sub-block length n_0 of a convolutional code and thereby reduce the size of a Davies-Meyer function down to the sub-block length n_0 of a convolutional code from a code length n which is a code length in a case where a block code is used. In short, the number of units of the basic structure of the Davies-Meyer function is reduced to $1/N$.

Theorem 4. *Encode k_0 message blocks each into n_0 convolutional subcode words and construct $n_0 \times 2$ inputs for Multiple Davies-Meyer scheme with n_0 previous hashed value and n_0 convolutional subcode words. Let the constraint length of the convolutional codes be N. A collision or a preimage happens on $N \times n_0$ consecutive $m-$bit blocks, and P Davies-Meyer functions are active, then $P-(d-1)$ input vectors $Y_j s$ are independent and remaining $d-1$ inputs depend on the first k inputs. In this condition, as far as the Assumption 2 holds, at least $2^{(d-1)m/2}$ encryptions are necessary to find a collision to a compression function and at least $2^{(d-1)m}$ encryptions are necessary to find a preimage to the compression function.*

As an operational notice, the initial key value of $n_0 m$ bits are required. Our construction is characterized by a Convolutional encoder, a multiple Davies-Meyer function, and an FIFO memory which accumulates hashed values which are concatenated to N time units. Roughly speaking, there are two types of Convolutional encoders for Convolutional codes. That is, type 1 encoders and type 2 encoders introduced by Massey J.L.[Mas63]. This will be described with reference to an example. Convolutional codes with sub-block length n_0 bits, sub-information symbol bits k_0, distance d and constraint length N which is a span of N time units, are called $(n_0, k_0, d; N)$ Convolutional codes. Consider a type 1 encoder for (3, 2, 3; 3) Convolutional codes. Two sub-generators are expressed as follows: [Lin70]

$$g(1,1) = 101, g(2,1) = 110$$

The respective sub-generator elements are:

$$g_0(1,1) = 1, g_1(1,1) = 0, g_2(1,1) = 1$$

$$g_0(2,1) = 1, g_1(2,1) = 1, g_2(2,1) = 0.$$

Constructed with these, an encoder is obtained[Mas63]. A type 2 construction of an encoder according to Massey J.L. is also available using this method [Mas63]. Let us consider an example of operations of a construction which uses the type 1 construction of Massey J.L. and the encoder for (3, 2, 3; 3) Convolutional codes. The symbols M_{1i} and M_{2i} denote messages which are $m-$bit blocks, the symbol H_i denotes n_0m-bit block, and the symbol C_i denotes an $m-$bit block of a check of a Convolutional code. H_j holds the relation $H_j = H_i$, in which case a delay circuit of one unit time ago is not necessary if a multiple Davies-Meyer function is of a D-latch input type. If the multiple Davies-Meyer function is constructed with wired logic, $H_j = H_{i-1}$ holds. On the other hand, C_{i-2} and M_i are inputted at the same time, since the input side of the multiple Davies-Meyer function is C_i. This applies to the following as well. H_0 (i = 0 for H_i) is supplied as an initial value to the input side of the multiple Davies-Meyer function, and initial values C_{10} and C_{20} are supplied respectively to check symbol registers C_1 and C_2. (Suppose C_1 is closer to multiple Davies-Meyer and C_1, C_2 are cascade connected). First messages M_{1i} and M_{2i} are added to each other by modulo-2 addition and input as a check symbol C_i. A hashed value H_i of $3 \times 64 = 192$ bits is obtained as a result of one operation of the multiple Davies-Meyer function and supplied to the FIFO memory while at the same time fed back to the input side. The next set of a message, a check symbol and a hashed values is then input and a hashed value H_2 is obtained as a result of the next operation. H_i are generated one after another in this manner, so that the FIFO memory always stores a hashed value $H_{i-2} \parallel H_{i-1} \parallel H_i$ which is equivalent to $3 \times 3 = 9$ pieces of $m-$bit blocks in total. Since Convolutional encoding is encoded after data are all input, $N - 1$, i.e., $2 \times 2 = 4$ pieces of $m-$bit blocks with a data value 0 are input and computation of a hashed value completes.

Although the size of a multiple Davies-Meyer function is reduced down to $1/N$ if Convolutional codes are used, in order to obtain a hashed value having a constraint length N and $N \times m$ bits, it is necessary to input 0 of $(N-1)m$ bits at the end of data so that the output data will be taken completely from an encoder. This causes a delay which is $(N - 1)$times as large as a unit time of encoding. In the construction according to this method, $n_0 \times 2$ pieces of two-dimensional arrangements are constructed on the input side of the multiple Davies-Meyer function, n_0 hashed values of one unit time ago are fed back to n_0 pieces of $m-$bit blocks in the first column, k_0 pieces of $m-$bit blocks of a new message are encoded into n_0 pieces of $m-$bit blocks and thereafter input. Table 2 shows the number of dummy data which are needed at initialization and termination of the construction according to this paper. Table 3 shows an example where Wyner-Ash codes are used and Table 4 shows an example where CSOC codes (Convolutional Self-Orthogonal Codes) are used, respectively [Lin70].

Table 2. Dummy data for the proposed construction

	initial	final
hash dummy (any character)	n_0 ($m-$bit block)	
hash dummy(0)		$(N-1)k_0$ ($m-$bit block)
check dummy (0)	$N-1$ ($m-$bit block)	

Table 3. A construction from Wyner-Ash codes

Code (W-A) $(n_0, k_0, d; N)$	hash rate	collision	memory(hash)
(4,3,3;2)	3/4	2^m	8m
(8,7,3;2)	7/8	2^m	16m

5 Summary

We developed the method which uses binary codes and the method which uses convolutional codes from the Knudsen and Preneel's method which requires to construct a hash function using error correcting codes, and described in this paper constructions according to our methods. In recent years, we have seen serious discussions on the security of key length of block ciphers, and ciphers with 128 bits or more are becoming a main stream these days. To respond to such current demands, researches for combining conventional methods to obtain secure hash functions should be more and more actively conducted.

In relation to the selection process which is ongoing for AES (Advanced Encryption Standard) in U.S., it is necessary to establish a method of constructing a hash function using a block cipher which has a longer key length. At the same time, continued researches on specific algorithms for hash functions are necessary.

A challenge from now is a block cipher, such as MISTY [Mat96] and IDEA [LM90], whose encryption/decryption key is 128 bits, i.e.,$a = 2$ despite an input satisfying $a \geq 2$, that is, $m = 64$ and an output of 64 bits. Meanwhile, according to tandem D-M (Tandem Davies-Meyer)[Sch96] and abreast D-M (Tandem Davies-Meyer)[Sch96], $2m-$bit hashed value is obtained in response to an $m-$bit key. Inputs to a hash function are the $2m-$bit hashed value at a time $i-1$ and the $m-$bit key input. Further, generally considering a cipher which uses a key having a key length of am bits satisfying $a \geq 2$ and for which a hashed value of bm bits is obtained, the direction of rows is $a+b$. A technique for synthesizing a hash function which is applicable to such a cryptographic method should be researched. Those researches are expected to considerably improve security.

Table 4. A construction from SCOC codes

code	hash rate	collision	memory
(CSOC) $(n_0, k_0, d; N)$			
(3,2,3;3)	2/3	2^m	9m
(3,2,5;14)	2/3	2^{2m}	42m

Acknowledgments

The first author would like to appreciate Prof. Shigeichi Hirasawa (Waseda University) and Prof. Takakazu Satoh (Saitama University) for useful discussions.

References

Dam89. Damgard, I.,B.,"A design principle for hash functions," Advances in Cryptology, Proc. Crypto'89, LNCS 435, Brassard, B. Ed., Springer-Verlag, 1990, pp.416-427. 394

BB94. den Boer,B. and Bosselaers,"Collisions for the compression function of MD5," Advances in Cryptology, Proc. Eurocrypt'93, LNCS 765, T. Helleseth, Ed., Springer-Verlag, 1994, pp.293- 304.

DP84. Davies D.W. and Price W.L.," Digital Signature An Update: Proceedings of International Conference on Computer Communications", Sydney, Oct 1984, North Holland: Elsevier, 1985, pp.843-847. 393

Dob96a. Dobbertin H., "Cryptanarysis of MD5 compress", Presented at the rump session of EUROCRYPT'96, May 1996. 391

Dob96b. Dobbertin,H. "Cryptanalysis of MD4," Fast Software Encryption, LNCS 1039, Gollman, D. Ed., Springer-Verlag, 1996, pp.53-69. 391

HLMW94. Hohl,W., Lai X., Meier T. and Waldvogel C.,"Security of iterated hash functions based on block ciphers," Advances in Cryptology, Proc., Crypto'93, LNCS 773, Stinson,D., Ed., Springer-Verlag, 1994, pp.379-390.

ISO. ISO/IEC 10118, "Information technology, Security techniques, Hash-functions, Part1: General and Part2:Hash-functions using an n-bit block cipher algorithm," .

Knu93. Knudsen L.R., "Analysis and design of cryptographic hash functions," Doctoral Dissertation, Katholieke Universiteit Leuven, 1993.

Knu92. Knudsen L.R., Govaerts R. and Vandewalle J.,"On the power of memory in the design of collision resistant hash functions," Advances in cryptology, Proc. Auscrypt'92, LNCS 718, Seberry J. and Zheng Y., ed., Spring-Verlag, 1993, pp.105-121.

Knu94. Knudsen L.R.,"A Key-schedule Weakness in SAFER K-64," Advances in Cryptology, Proc. Crypto'94, LNCS 839, Desmedt. Y. ed., Springer-Verlag, 1994, pp.274-286.

KL94. Knudsen L. R., and Lai X.,"New attacks on all double block length hash functions of hash rate 1, including the pararell-DM," Advances in Cryptology, Proc. Euro-crypto'94, LNCS 959, De Santes, A., Ed., Spring-Verlag,1995, pp.410-418.

KP96. Knudsen L.R.,and Preneel B., " Hash functions based on block ciphers and quaternary codes", Advances in Cryptology, Proc. Asiacrypto'96, LNCS 1163, K. Kim, T. Matsumoto, Eds., Springer-Verlag, 1996, pp.77-90. 391, 392, 394, 395

KP97. Knudsen L. and Preneel B. "Fast and secure hashing based on codes," Crypto'97, LNCS1294, pp.485-498, 1997. 391, 391, 392, 394, 394, 394, 395, 395, 395

Lai92. Lai X.,"On the Design and Security of Block Ciphers," ETH Series in Information Processing, Vol.1, Massey J. L. Ed., Hartung-Gorre Verlag, Konstanz, 1992.

LM90. Lai X. and Massey J., "A Proposal for a New Block Encryption Standard", Advances in Cryptology-EUROCRYPTO'90 Proceedings, Springer-Verlag, 1991,pp.389-404.

Lin70. Lin, S, "An Introduction to Error Correction Codes", Englewood Cliffs., N., J., 1970, Chapter 10.

MS78. Macwilliams F.J. and Sloane N.J.A.,"The Theory of Error-Correcting Codes," North-Holland Publishing Company, Amsterdam, 1978. 401 399, 400

MMO85. Matyas S. M., Meyer C. H. and Oseas,J.,"Generating strong one-way functions with cryptographic algorithm," IBM Techn. Disclosure, Bull., Vol. 27, No. 10A, 1985, pp.5658-5659. 393

Me89. Merkle R.,"One way hash functions and DES," Advances in Cryptology, Proc., Crypto'89, LNCS 435, Brassard G. Ed., Spring-Verlag, 1990, pp.428-446. 394

MS88. Meyer C. H. and Schilling M.,"Secure program load with Manipulation Detection Code," Proc. Securicom 1988, pp.111-130.

Mas63. Massey, J.L, "Threshold Decoding", MIT Press, 1963. 399, 400, 400

Mat96. Matsui M.,"New structure of block ciphers with provable security against differential and linear cryptoanalysis", the third international workshop of fast software encryption,1996. 401

MS87. Moore j. H. and Simmons G. J.,"Cyclic structure of the DES for keys having palindromic (or antipalindromic) sequences of round keys", IEEE Trans. on Software Engineering, Vol.SE-13, No.2, 1987, pp.262-273.

NY89. Naor M. and Yung M.,"Universal one-way hash functions and their cryptographic applications," Proc. 21st ACM Symposium on the Theory of Computing, ACM, 1989, pp.387-394.

PGV94. Preneel B., Govaets R. and Vandewalle J.,"Hash functions based on block ciphers: a synthetic approach," Advances in Cryptology, Proc., Crypto'93, LNCS 773, Stinson, D.,Ed., Spring- Verlag, 1994, pp.368-378.

QD89. Quisquater, J.-J. and Delescaille J.-P.,"How easy is collision search? Application to DES," Advances in Cryptology, Proc. Eurocrypt'89, LNCS 434, Quisquater, J.-J. and Delescaille J.-P., Ed., Springer-Verlag, 1990, pp.429-434.

Riv90. Rivest, R.L.,"The MD4 message digest algorithm," Advances in Cryptology, Proc. Crypto'90, LNCS 537.S. Vanstone, Ed., Spring-Verlag, 1991, pp.303-311.

Riv92. Rivest, R.L.,"The MD5 message digest algorithm," Request for Comments (RFC) 1321, Internet Activities Board, Internet Privacy Task force, April 1992.

RP95. Rijmen V. and Preneel B.,"Improved characteristics for differential cryptanalysis of hash functions based on block ciphers," Fast Software Encryption, LNCS 1008, Preneel B.,Ed., Springer-Verlag, 1995, pp.242-248.

Sch96. Schneier B.,: Applied cryptography, John Wiley & Sons, Inc., New York, pp.451-452, 1996. 401, 401

OW94. Van Oorshot, P. C. and Wiener M. J.,"Parallel collision search with application to hash functions and discrete logarithms," Proc. 2nd ACM Conference on Computer and Communications Security, ACM, 1994, pp.210-218.