On the Power of Memory in the Design of Collision Resistant Hash Functions

Bart Preneel*, René Govaerts, and Joos Vandewalle

Katholieke Universiteit Leuven, Laboratorium ESAT-COSIC,
Kardinaal Mercierlaan 94, B–3001 Heverlee, Belgium

Abstract. Collision resistant hash functions are an important basic tool for cryptographic applications such as digital signature schemes and integrity protection based on "fingerprinting". This paper proposes a new efficient class of hash functions based on a block cipher that allows for a tradeoff between security and speed. The principles behind the scheme can be used to optimize similar proposals.

1 Introduction

Although more theoretical definitions of collision resistant hash functions are available [Dam89], we will be satisfied with a more practical definition as given in [Mer89].

Definition 1 *A function $h()$ is a* **collision resistant hash function** *if:*

- *The description of $h()$ is* publicly known *and does not require any secret information for its operation (extension of Kerckhoff's principle).*
- *The argument X can be of* arbitrary length *and the result has a fixed length of h (with $h \geq 112 - 128$ bits in order to avoid the birthday or square-root attack* [QD89,Yuv79]*).*
- *Given $h()$ and X, the computation of $h(X)$ must be* "easy"*.*
- *The hash function must be* one-way *in the sense that given a Y in the image of h, it is "hard" to find a message X such that $h(X) = Y$, and given X and $h(X)$, it is "hard" to find a message $X' \neq X$ such that $h(X') = h(X)$.*
- *If is* "hard" *to find two distinct arguments that hash to the same result (the collision resistant property).*

Here "easy" and "hard" can to be substituted by adequate definitions. In this paper, "hard" will mean that it requires at least 2^S encryptions, with S the security level of the hash function. Note that under certain circumstances one-wayness is implied by the collision resistant property [Dam89].

Two arguments can be indicated to construct a hash function based on a block cipher. The first argument is the minimization of the design and implementation effort. The major advantage however is that the trust in an existing

* NFWO aspirant navorser, sponsored by the National Fund for Scientific Research (Belgium).

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

block cipher can be transferred to a hash function. It is important to note that for the time being significantly more research has been spent on the design of secure block ciphers compared to the effort to design hash functions. It is also not obvious at all that the limited number of design principles for encryption algorithms are also valid for hash functions. The main disadvantage of this approach is that dedicated hash functions are likely to be more efficient. Moreover some block ciphers show certain weaknesses that are only relevant if they are used in a hash function. One also has to take into account that in some countries export restrictions apply to encryption algorithms but not to hash functions.

For a hash function based on a block cipher, the following notations have to be fixed. The block length i.e., the size of plaintext and ciphertext in bits will be denoted with n. The argument of the hash function is padded with an unambiguous padding rule such that it is a multiple t of the block size. The hash function 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. Finally 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.

A large number of hash functions based on a block cipher have been proposed [MPW91,Pre93]. For most schemes the size of the hashcode is equal to the block length. However, the block ciphers that have been proposed in literature comprising DES [Fi46] have only a block length of $n = 64$ bits, which implies that for a collision resistant hash function one needs that the size of the hashcode $h = 2n$. An additional problem is that the key length of DES is only 56 bits. Existing proposals of this type are MDC-2 and MDC-4 [MS88], with rate 2 and 4 respectively, three schemes by R. Merkle [Mer89] with rate 18.3, 5.8, and 3.62. More efficient schemes have been proposed with a rate close to 1 [BPS90,PBG89,QG89], but currently all these proposals have been broken [Cop92,Pre93]. The constructions by R. Merkle are based on a collision resistant function (the "meta-method" [Dam89,Mer89]), and for these schemes it is possible to write down a proof based on a black box model of DES. On the other hand, if one wants to use the scheme with DES, it still has to be modified to take into account properties like the weak keys and the complementation property, and it should be checked for vulnerability to specific attacks (e.g., differential attacks). Two new schemes based on a block cipher with a double length key have been proposed recently [LM92]. The security level of all these schemes is 64 bits, 56 bits or even smaller.

In this paper a new class of schemes will be proposed, that allow for a tradeoff between security level, rate, and size of the hashcode. Their rate lies between 4 and 8, but they are *faster* than most other schemes because the key remains fixed during the evaluation of the hash function. This also implies that they can be applied *under more general circumstances.*

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

2 Design Principles

The basic principle is that the key remains fixed during the hashing process. This has the following advantages:

performance: in general, the key scheduling is significantly slower than the encryption operation. A first argument to support this is that the key scheduling can be designed as a very complex software oriented process to discourage exhaustive attacks. Here software oriented means that the variables are updated sequentially, which reduces the advantages of a parallel hardware implementation. Even when the key schedule is simple, it can be harder to optimize its implementation. E.g., for highly optimized DES software, an encryption with key change will between 2.5 and 4.5 times slower. Moreover, encryption hardware is in general not designed to allow fast modification of the key, as a key change can cause loss of pipelining, resulting in a serious speed penalty.

security: an attacker has no control at all over the key. Hence attacks based on weak keys can be eliminated completely in the design stage.

generality: the hash function can be based on any one-way function with small dimensions (e.g., 64 bit input and output).

collision resistant MAC: if the keys are kept secret, the scheme gives a construction for a MAC for which it is hard to produce collisions even for someone who knows the secret key. This is not possible with the widespread schemes for a MAC. An application where this property might be useful has been discussed in [MW88].

The other design principles of the scheme are:

atomic operation: the one-way function that is used will be encryption of the argument X with the key K followed by the addition modulo 2 of X to the ciphertext: $E(K, X) \oplus X$. It has shown to be very useful for single length hash functions (e.g., [MMO85]) and was also used by R. Merkle and in MDC-2 and MDC-4.

parallel operation : the one-way function will be used more than once, but it will be possible to evaluate the function in parallel; this opens the possibility of a fast parallel hardware implementation. It is also clear that a scheme in which several instances are used in a serial way is much harder to analyze.

tradeoff between memory, rate, and security level : the rate of the system can be decreased at the cost of a decreasing security level; it will also be possible to decrease the rate by increasing the size of the hashcode. This could also be formulated in a negative way, namely that the security level will be smaller than what one would expect based on the size of the hashcode. Observe that this property is also present in a limited way in MDC-2, MDC-4 [MS88], and in the schemes of R. Merkle.

the basic function is not collision resistant: the construction is not based on a collision resistant function, because it can be shown that this would not yield an acceptable performance (the efficiency will decrease with a factor

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

4 or more). This means that producing collisions for different values of the chaining variable is easy. As it should be hard to produce collisions for the data input, this input will be protected more strongly.

3 Description of the New Scheme

One iteration step consists of k parallel instances of the one-way function, each parameterized with a fixed and different key. In the following, these instances will be called 'blocks'. These k keys should be tested for specific weaknesses: in the case of DES it is recommended that all 16 round keys are different, and that no key is the complement of any other one. The total number of input bits is equal to kn. These inputs are selected from the x bits of the data X_i and from the h bits of the previous value of the chaining variable H_{i-1}. Every bit of X_i will be selected α times, and every bit of H_{i-1} will be selected only once (this can be generalized), which implies the following basic relation:

$$\alpha \cdot x + h = k \cdot n \,. \tag{1}$$

The rate R of this scheme is given by the expression:

$$R = \frac{n \cdot k}{x} \,. \tag{2}$$

As it does not make sense to enter the same bit twice to a single block, it will always be the case that $2 \leq \alpha \leq k$.

The output of the functions has also size kn. This will be reduced to a size of h by selecting only h/k bits from every output. Subsequently a simple mixing operation is executed, comparable to the exchange of left and right halves in MDC-2. The goal of this operation is to avoid that the hashing operation consists of k independent chains. If h is a multiple of k^2, this mixing operation can be described as follows. The selected output block of every function (consisting of h/k bits) is divided into k parts, and part j of block i is denoted with H^{ij} $(1 \leq i, j \leq k)$. Then $H^{ji}_{\text{out}} \longleftarrow H^{ij}_{\text{in}}$. Figure 1 depicts one iteration for the case $k = 4$ and $\alpha = 4$. It will become clear that the complexity of the scheme does not depend on α, but on the difference between k and α. Therefore, the parameter ϕ is defined as $k - \alpha$.

The next step in the design is the decision on how the data bits are to be distributed over the different blocks if $\phi > 0$. The construction is obtained by considering the following attack. Let $\mathcal{S}$ be a subset of the blocks. For a given value of H_{i-1}, fix the data input of the blocks in $\mathcal{S}$, which means that the output of these blocks will be the same for both input values. Subsequently, match the remaining outputs with a birthday attack. In order to maximize the effort for this attack, it is required that after fixing the input of the blocks in $\mathcal{S}$, the number of bits that can still be freely chosen by an attacker is minimal (this will be explained in more detail in Sect. 4). This number will be denoted with $A(\mathcal{S})$.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

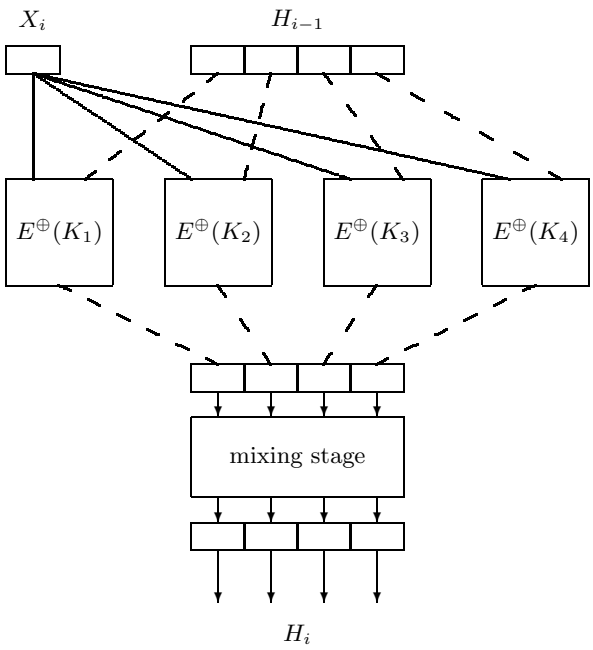

Fig. 1. One iteration of the new hash function proposal.

Theorem 1 *If the data is divided into* $\binom{k}{\phi}$ *parts, and every part goes to a different combination of* $\binom{k}{\phi}$ *blocks, an optimal construction is obtained. Let* A_s *be defined as* $\max_{|\mathcal{S}|=s} A(\mathcal{S})$*, then this construction results in the following expression for* A_s*:*

$$A_s = \frac{\binom{k-s}{k-\phi}}{\binom{k}{\phi}} \cdot x = \frac{\binom{\phi}{s}}{\binom{k}{s}} \cdot x \qquad 1 \le s \le \phi \tag{3a}$$

$$= 0 \qquad \textit{else} \ . \tag{3b}$$

This construction is optimal in the sense that for any other construction there will be an s *(with* $1 \le s \le k$*) such that* $A'_s \ge A_s$*. If equality holds for all values of* s *both constructions are equivalent.*

In order to clarify the scheme, a detailed description will be given for the case $n = 64$, $k = 4$, $\phi = 2$, $h = 148$, and hence $x = 54$. In this case Theorem 1 states that in order to optimize the security level, X_i has to be split into 6 parts of 9 bits each. The first 9-bit part of X_i goes to blocks 1 and 2, the second 9-bit part to encryption block 1 and 3, etc., and the sixth part goes to block 3 and 4. The 64-bit input of a single block cipher consists of $148/4 = 37$ bits of h and 3 parts of 9 bits each. The rate of this scheme is 4.7 (but with a fixed key), and from our evaluation it follows that the security level is about 55 bits. In the next sections it will be explained how the security level can be determined.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

4 Attacks on the Scheme

A security proof for the scheme can not be given for the time being. This disadvantage is shared with all other schemes (including MDC-2 and MDC-4); the only exceptions are the schemes by R. Merkle [Mer89]. The main difference with the other schemes is that the system is parameterized, and that the security level depends on the size of hashcode h.

In the following, the number of operations to produce a preimage and a collision for the hash function will be studied by considering a number of attacks that are faster than a straightforward exhaustive or birthday attack. Such attacks are possible as not all output bits depend in a strong way on the inputs of a single iteration step. Indeed, the data only enter α blocks, and hence if $\alpha < k$, the output of $\phi = k - \alpha$ blocks does not depend on these input bits. The diffusion of the H_{i-1} is limited to one block. Note that this property is shared with MDC-2. This limited dependency is solved by increasing the size of the hashcode. The required number of bits for the hashcode is estimated from studying a set of attacks that exploit the structure of the scheme. The generality of the proposed attacks should form an argument for the security. However, it is for the time being not possible to prove that there does not exist any more sophisticated attack. The advantage of the scheme is that the security level can always be increased at the cost of an increased memory and decreased efficiency. By construction the scheme is not vulnerable to attacks based on weak keys or based on the complementation property.

Before discussing the collision attacks in detail, expressions are required for the number of operations to produce a collision under certain constraints. Assume one has a random function with B output bits and A input bits that can be chosen arbitrarily. The function might have C inputs bits that can not be chosen freely; these input bits will be called parameters. If a collision is to be produced for this function for a certain value of a parameter, i.e., two inputs that result in the same output bits, two cases have to be distinguished:

$A > B/2$: in this case producing a collision requires $2^{B/2}$ function evaluations.

$A < B/2$: in this case, the number of expected collisions after a single trial is equal to $p = (2^A)^2/2^B$. This process will be repeated for several values of the parameter (it is assumed that C is sufficiently large). The expected number of trials is given by $1/p$ and the effort for a single trial is 2^A function evaluations. Hence the expected number of function evaluations is equal to $2^B/2^A$, which is always larger than $2^{B/2}$.

For the evaluation of the scheme one has to determine an expression for the number of operations to produce a 2^c-fold collision. It can be shown that for large values of c this number is given by 2^{B+c} (if A is sufficiently large). For smaller values of c expressions have been derived in [Pre93].

Four types of birthday attacks that exploit the structure of the scheme will be discussed. They are only valid if $\phi > 0$. All these attacks yield a *piece-wise linear relation* between memory h and security level S. Because of (1) and (2), there will be a hyperbolic relation between the rate R and the security level S.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

Attack A: fix the input bits to the first s blocks and match the output of the remaining $k-s$ blocks with a birthday attack. The number of output bits of these blocks is denoted with B_s. It is clear that $B_s = \frac{k-s}{k} \cdot h$. The binary logarithm of number of operations for this attack is given by the following expression:

$$\begin{cases} \frac{B_s}{2} + 1 + \log_2(k-s) & \text{if } A_s \geq \frac{B_s}{2} \\ B_s - A_s + 1 + \log_2(k-s) & \text{if } A_s < \frac{B_s}{2}\,. \end{cases}$$

Attack B: a more effective attack consists of generating a 2^c-fold collision for the first s blocks. In the next step, one has $A_s + c$ degrees of freedom available to match the remaining $k-s$ blocks. This attack has already two parameters: s and c. A problem that should be considered is the following: for large values of c, an attacker needs about $h\frac{s}{k} + c$ degrees of freedom to produce such a multiple collision. In most cases, there are not that many data bits that enter the first block(s). However, one can assume that an attacker can also introduce variations in the previous iteration steps. If we would have designed a collision resistant function, this assumption would not have been valid. In that case any attack has to produce a collision for a single iteration step.

Attack C: under certain circumstances, attack B can be optimized by exploiting the block structure: first a 2^{c_1}-fold collision is produced for the output of the first block (possibly using variations in previous rounds), subsequently a 2^{c_2}-fold collision is produced for the output of the second block. This continues until a 2^{c_s}-fold collision is produced for the output of block s. Finally the last s blocks are matched with a birthday attack with $A_s + c_s$ degrees of freedom. The attack is optimized with respected to the parameter c_s; the choice of c_s fixes the other c_i's as follows: in order to produce a 2^{c_s}-fold collision for block s, $h/k + c_s$ trials are necessary (assuming c_s is not too small). There are only a_s $(= A_{s-1} - A_s)$ bits available at the input of block s, which means that a $2^{c_{s-1}}$-fold collision will be required at the output of block $s-1$, with $c_{s-1} = h/k + c_s - a_s$. This procedure is repeated until the first block is reached. It is assumed that there are sufficient degrees of freedom are available through variations in the previous iterations.

Attack D: This attack is different from attacks B and C because it makes a more explicit use of interaction with the previous iteration step. It is based on the observation that if h is significantly smaller than the value that is being evaluated, it should be easy to produce collisions or even multiple collisions for H_{i-1}. Therefore it should also be easy to produce multiple collisions for the first s' blocks of H_{i-1}, that contain $h' = s'/h$ bits. The data bits that enter the first s' block are now fixed, and this implies that the output of these blocks will be identical. From block $s'+1$ on, the attack continues with a type C attack.

The next step consists in determining the number of operations to produce a multiple collision for the first s' blocks of H_{i-1}. This can be done by

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

determining the optimal attack on a reduced scheme. It might be that again attack D is optimal and then the evaluation program works in a recursive mode.
The problem that remains to be solved is how to evaluate the number of operations to produce a multiple collision for the first s' blocks of H_{i-1}. This number is estimated by calculating the number of operations for a collision S'; in general S' will be smaller than $h'/2$. Subsequently, the expression for the number of operations for a multiple collision is used where the size of the block is replaced by the effective block length $2S'$. The number S' can be found by comparing the efficiency of attacks A, B, C, and D for a scheme with the same configuration as the original one (this means the number of data bits has not been modified), but with h replaced by h' (at the output). If attack D is optimal, the program works in a recursive mode.

The non-linear behavior of the number of operations for the birthday attacks and the recursive nature of the attack D forced us to evaluate the security with a computer program. However, the results can be verified by using approximations for the expressions.

The applicability of differential attacks to hash functions based on block ciphers has been studied in [Pre93]. The main reasons why differential attacks on this scheme will not work if DES is used as the underlying block cipher is that there has not been found a good characteristic with an even number of rounds. Differential attacks on this scheme are harder because the attacker has no complete control over the plaintext, and because the keys can be selected in a special way to minimize the probability of iterative characteristics (which is not possible for MDC-2). On the other hand, the characteristic has only to hold for the subset of the output bits that has been selected. This means that in the last round the characteristic must not be completely satisfied. However, it should be noted that the success probability will decrease very fast if the characteristic is not satisfied in earlier rounds. Like in the case of MDC-2, every data bit is used at least twice, which implies that a characteristic with a high probability is required for an attack faster than exhaustive search. Once a detailed scheme has been fixed, more work can be done on selecting the keys in such a way that differential attacks are less efficient.

When looking for a preimage, one can also exploit the limited dependencies between H_i and H_{i-1}. Going one step backwards in the chain (for a fixed X_i) requires only $2^{h/k+\log_2(k)}$ operations. Subsequently one will try to obtain a match for the chaining variable with a meet in the middle attack. This implies that one can find a preimage with less than 2^h operations. However, for the time being no attack has been identified that is faster than the attacks that look for a collision.

5 A Detailed Study of the Security Level

In this section the security level of schemes with ϕ between 0 and 4 will be discussed. In order to keep the schemes practical, the value of k will be limited to 6. If $\phi = 0$ it can be shown that for practical values of the security level

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

the schemes with small values of k are more efficient. For all other schemes, larger values of k would imply that there are too many small parts (recall that the number of parts is equal to $\binom{k}{\phi}$), which would increase the overhead of bit manipulations to an unacceptable level. Finally note that the restriction $\alpha \geq 2$ is equivalent to $k \geq \phi + 2$.

5.1 The Case $\phi = 0$

In this case $\alpha = k$, which means that every data bit goes to every block. There are no problems in determining an optimal configuration of the scheme. As indicated above, none of above attacks applies, which means that the security level is equal to its upper bound $S = h/2$. It can be shown that under certain circumstances the security level is 1 bit lower.

The expression for the rate of this scheme reduces to $R = k/(1 - \frac{2S}{kn})$. It can be seen that it becomes advantageous to increase k by one if the security level is given by $S = n \cdot \frac{k(k+1)}{2k+1}$. This means that $k = 4$ will be the optimal solution for a security level between 54.9 and 71.1 bits. A graphical representation of the relation between R and S is given in Fig. 2.

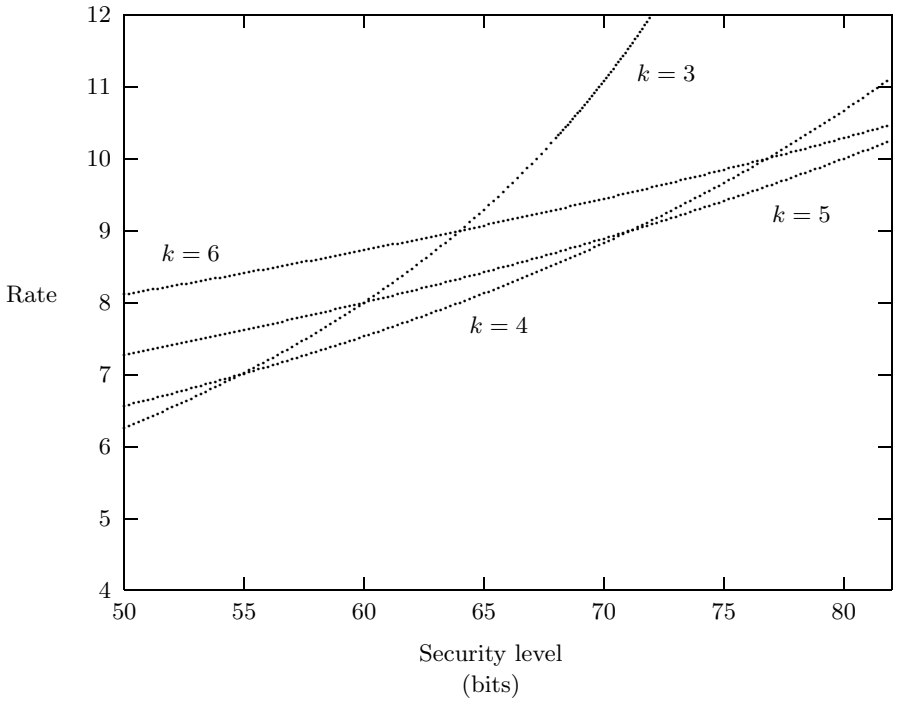

Fig. 2. Relation between the rate R and the security level S for $\phi = 0$ and k between 3 and 6. The optimal value of k increases with S.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

5.2 The Case $\phi = 1$

In this case the data input is split into k parts, and every part goes to $k-1$ blocks. Computer calculation has shown that the most efficient attack is attack D with $s = s' = 1$. The number of operations can be approximated by the following expressions:

1. number of operations to produce a collision for blocks 2 to k:

$$\frac{k-1}{k}\, h - \frac{x}{k} + 1 + \log_2(k-1) - c\,.$$

2. number of operations to produce a 2^c-fold collision for the first block of H_{i-1}:

$$\frac{k-1}{k}\frac{h}{k} + 2(1 + \log_2(k-1)) + c\,.$$

The approximation lies in the fact that the logarithm of the number of operations to produce a 2^c-fold collision is not a linear function of c for small values of c. The total number of operations should be minimized with respect to c, which yields the following expression for the security level:

$$S = \frac{h}{2}\left[\frac{k^2-1}{k^2} + \frac{1}{k(k-1)}\right] - n\,\frac{2}{k-1} + \frac{5}{2} + \frac{3}{2}\log_2(k-1)\,.$$

For k between 3 and 5, the relation between h and S is indicated in Table 1. For $k \geq 6$, the resulting expression is larger than $h/2$ for all values of h, which means that a simple birthday attack is more efficient. The theoretical results agree very well with the result obtained from computer calculations. The program shows that $k = 4$ is the best choice for S between 51 and 72 bits.

k	Security level S (bits)
3	$\frac{19}{36}h - 12.0$
4	$\frac{49}{96}h - 5.8$
5	$\frac{101}{200}h - 2.5$

Table 1. Relation between h and S for $\phi = 1$ and $k = 3$, 4, and 5.

5.3 The Case $\phi = 2$

In this case the data input is split into $k(k-1)/2$ parts, and every part goes to $k-2$ blocks. Computer calculation has shown that the most efficient attack is attack D with $s = 2$ and $s' = 1$. The number of operations can be approximated by the following expressions:

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

1. number of operations to produce a collision for blocks 3 to k:

$$h\left[\frac{k-2}{k}+\frac{2}{k(k-1)(k-2)}\right]-n\,\frac{2}{(k-1)(k-2)}+1+\log_2(k-2)-c\,.$$

2. number of operations to produce a 2^c-fold collision for block 2:

$$\frac{h}{k}+c\,.$$

3. number of operations to produce a 2^c-fold collision for the first block of H_{i-1}:

$$h\left[\frac{1}{k}+\frac{k-2}{k^2}+\frac{2}{k(k-1)}\right]-n\,\frac{2}{k-1}+2(1+\log_2(k-3))+c\,.$$

This number of operations should be minimized with respect to c. For smaller values of h, the third term is negligible, while for larger values of h, the second term is negligible. For k between 4 and 6, the relation between h and S is indicated in Table 2. The program shows that $k = 4$ is the best choice for S smaller than 69 bits.

k	Security level S (bits)	
4	$h \geq 132\ \frac{27}{48}h - 28.0$	$h \leq 132\ \frac{10}{24}h - 8.7$
5	$h \geq 122\ \frac{79}{150}h - 16.5$	$h \leq 122\ \frac{5}{12}h - 3.0$
6	$h \geq 110\ \frac{37}{72}h - 10.5$	$h \leq 110\ \frac{17}{40}h - 0.7$

Table 2. Relation between h and S for $\phi = 2$ and $k = 4$, 5, and 6.

5.4 The Case $\phi = 3$

In this case the data input is split into $\binom{k}{3}$ parts, and every part goes to $k-3$ blocks. Computer calculation has shown that the most efficient attack is attack D with $s = 3$ and $s' = 1$. The number of operations can be approximated by the following expressions:

1. number of operations to produce a collision for blocks 4 to k:

$$h\left[\frac{k-3}{k}+\frac{6}{k(k-1)(k-2)(k-3)}\right]-n\,\frac{6}{(k-1)(k-2)(k-3)}+1+\log_2(k-3)-c\,.$$

2. number of operations to produce a 2^c-fold collision for block 3:

$$\frac{h}{k}+c\,.$$

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

3. number of operations to produce a $2^{c'}$-fold collision for block 2 (with $c' = h/k + c - a_3$):

$$h\left[\frac{2}{k} + \frac{6}{k(k-1)(k-2)}\right] - n\,\frac{6}{(k-1)(k-2)} + 2(1+\log_2(k-3)) + c\,.$$

4. number of operations to produce a $2^{c''}$-fold collision for the first block of H_{i-1} (with $c'' = h/k + c' - a_2$). It can be shown that this number is significantly smaller than $\frac{h}{k} + c$.

This number of operations should be minimized with respect to c. The second term is always smaller than the third term. This results in the following expression for the security level:

$$S = \frac{h}{2}\left[\frac{k-1}{k} + \frac{6}{k(k-1)(k-3)}\right] - n\,\frac{3}{(k-1)(k-3)} + \frac{3}{2} + \frac{1}{2}\log_2(k-3)\,.$$

For k equal to 5 and 6, the relation between h and S is indicated in Table 3. The program shows that $k = 5$ is the best choice for S smaller than 82 bits.

k	Security level S (bits)
5	$\frac{19}{40}h - 22.0$
6	$\frac{9}{20}h - 10.5$

Table 3. Relation between h and S for $\phi = 2$ and $k = 4$, 5, and 6.

5.5 The Case $\phi = 4$

The only case that has been studied is $k = 6$. Here it is not possible to derive simple analytic expressions that are sufficiently accurate. This is because the result obtained by the different attacks lie very closely together; depending on the value of h, the best attack is attack D with $s = 4$ and $s' = 2$ or 4. Moreover, the optimal values of c are very small, which means that the system is non-linear, and the attacks are strongly dependent on the use of recursion. An upper bound on S can be easily obtained using method A:

$$S \le \frac{11}{30}h - 10.8\,.$$

A least squares fitting the computer evaluation yields a correlation coefficient of 0.99958 with the following expression:

$$S = 0.3756h - 14.974\,.$$

This can be approximated very well by $S = 3/8\ h - 15$.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

6 Extensions

Before the scheme can be applied in practice, one has to consider the following limitation: h and x are not continuous variables, but integers that have to satisfy certain constraints. The scheme can be extended by using every bit of H_i more than once, but this complicates the evaluation. Another extension is to design a collision resistant round function based on this scheme: this simplifies the evaluation but decreases the efficiency. Finally it is explained how the basic principles can be applied to other hash functions based on block ciphers.

The study of the previous scheme assumed that h and x are continuous variables. However, in practice they will have to be integers that satisfy certain constraints:

1. x has to be an integer multiple of $\binom{k}{\phi}$. Therefore define $x' = x/\binom{k}{\phi}$.
2. $nk - h$ has to be an integer multiple of $k - \phi$.
3. h has to be an integer multiple of k. Therefore define $h' = h/k$.

Note that in order to perform the mixing stage on the output of the k blocks, one needs in fact the requirement that h is an integer multiple of k^2. However, this mixing stage is not critical in the security analysis of the scheme. The following algorithm steps through all values of x and h that satisfy the constraints, for which $h > h_0$. First the starting values are generated:

$$x'_1 = \left\lceil \frac{n - \left\lceil \frac{h_0}{k} \right\rceil}{\binom{k-1}{\phi}} \right\rceil \tag{4}$$

$$x_1 = \binom{k}{\phi} x'_1 \tag{5}$$

$$h_1 = k\left(n - \binom{k-1}{\phi} x'_1\right) . \tag{6}$$

Here $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. The next values are calculated as follows:

$$x_{i+1} = x_i + \binom{k}{\phi} \tag{7}$$

$$h_{i+1} = h_i + k\binom{k-1}{\phi} . \tag{8}$$

In the overview of the results it will be graphically indicated which schemes satisfy the requirements. It is of course always possible to think of schemes for which the parts differ 1 or 2 bits in size just to match the constraints. This will affect the security level compared to the ideal situation, but the decrease will certainly be only marginal. These schemes are certainly less elegant, but as long as the asymmetry in the bit manipulations has no negative influence on the performance, this is not so important.

If the schemes are extended by using every bit of H_{i-1} more than once, the following elements will have to be considered. First, the allocation of the bits of

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

H_{i-1} to the different blocks will have to be made in a similar way as for the data bits. However, some additional work has to be done because both allocations should be as independent as possible, i.e., the bits of H_{i-1} and X_i will have to occur in as many combinations as possible. The study of attacks on this scheme is more complicated, especially for type D attacks.

Another issue is how to extend this method to construct a collision resistant function. In this case the attacks to be considered would be simpler, because the interaction with the previous iteration steps can be neglected. This is not completely true however, because an attacker could exploit the fact that the function is not complete, by producing collisions for part of the output blocks. However, if the program is adapted to evaluate this type of schemes, it becomes clear that they will never be very efficient: the best scheme with a security level of 56 bits under the constraint $k \leq 16$ has a rate of 20 (which is in case of software a little worse than MDC-4). It is a scheme with $\alpha = 3$ or $\phi = 12$. The size of the hashcode would be 272 bits, and every iteration processes 48 bits. The scheme is however very impractical because X_i has to be split into 455 blocks of 2 or 3 bits.

The basic principles used here can also be applied to other hash functions based on block ciphers where the key is modified in every iteration step. As an example it is indicated how MDC-2 could be extended in two ways to obtain a security level larger than 54 bits. The 2 parallel DES operations will be replaced by 3 parallel DES operations ($k = 3$).

- A trivial way would be $\alpha = 3$: every data bit is used 3 times as plaintext. The size of the hashcode would be 192 bits, and the effective security level is equal to 81 bits. The rate of this scheme is equal to 3 (comprising a key change).
- A second scheme can be obtained by selecting $\alpha = 2$: the data input of 96 bits is divided into 3 32-bit parts, that are allocated in the optimal way to the 3 blocks. The rate of this scheme is equal to 2 (the same as MDC-2), but the security level is slightly larger than 60 bits.

Of course it is possible to extend this for $k > 3$, which will result in faster schemes that require more memory.

A disadvantage of all these new schemes is that the decreased rate has to be paid for by increasing the memory. The additional 64 to 80 bits are no problem for the chaining variables (certainly not when the computations are performed in software), but the increased size of the hashcode might cause problems. This is not the case for digital signatures, as most signature schemes sign messages between 256 and 512 bits long. Exceptions to this rule are the scheme by Schnorr [Sch89] and DSA, the draft standard proposed by NIST [Fi91], where the size of the argument is 160 bits. If the hash function is used for fingerprinting computer files, an increased storage can pose a more serious problem. However, it can be solved by compressing the result to a number of bits equal to twice the security level S. This can be done with a (slow) hash function with $\phi = 0$, security level S, and size of the hashcode $2S$.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

7 Conclusion

A new class of hash functions based on block ciphers has been proposed, that allows for a tradeoff between security, memory, and speed. The parameters for some efficient schemes for a given value of ϕ and a given security level S are indicated in Table 4. For $k = 4$ and $k = 5$, the relation between the rate R and the security level S with parameter ϕ, with $0 \leq \phi \leq k-2$ can be found in Fig. 3 and 4. The solutions that take into account the discrete character are marked with a $\diamond$. The underlying principles can also be used to improve other proposals for hash functions based on block ciphers.

ϕ	k	security level $\simeq$ 54 bits				security level $\simeq$ 64 bits			
		R	h	S	$h(k-\phi)/\phi$	R	h	S	$h(k-\phi)/\phi$
0	4	6.92	108	54	108	8.00	128	64	128
1	4	5.82	124	58	93	6.40	136	64	102
2	4	4.74	148	55	74	6.10	172	69	86
3	5	4.00	160	53	64	4.57	180	62	72
4	6					4.27	204	62	68

Table 4. Overview of best results for small values of ϕ and k. The quantity $h(k-\phi)/\phi$ indicates how many output bits depend on a single input bit.

References

[BPS90] 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.

[Cop92] D. Coppersmith, "Two broken hash functions," *IBM T.J. Watson Center, Yorktown Heights, N.Y., 10598, Research Report RC 18397*, October 6, 1992.

[Dam89] 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.

[Fi46] *"Data Encryption Standard,"* Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.

[Fi91] *"Digital Signature Standard,"* Federal Information Processing Standard (FIPS), Draft, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., August 1991.

[GCC88] M. Girault, R. Cohen, and M. Campana, "A generalized birthday attack," *Advances in Cryptology, Proc. Eurocrypt'88, LNCS 330*, C.G. Günther, Ed., Springer-Verlag, 1988, pp. 129–156.

[LM92] X. Lai and J.L. Massey "Hash functions based on block ciphers," *Advances in Cryptology, Proc. Eurocrypt'92, LNCS*, R.A. Rueppel, Ed., Springer-Verlag, to appear.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

[MMO85] 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.

[Mer89] 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.

[MS88] C.H. Meyer and M. Schilling, "Secure program load with manipulation detection code," *Proc. SECURICOM 1988*, pp. 111–130.

[MW88] C. Mitchell and M. Walker, "Solutions to the multidestination secure electronic mail problem," *Computers & Security*, Vol. 7, 1988, pp. 483–488.

[MPW91] C.J. 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.

[PBG89] B. Preneel, A. Bosselaers, R. Govaerts, and J. Vandewalle, "Collision free hash functions based on blockcipher algorithms," *Proc. 1989 International Carnahan Conference on Security Technology*, pp. 203-210.

[Pre93] B. Preneel, "Analysis and design of cryptographic hash functions," *Doctoral Dissertation*, Katholieke Universiteit Leuven, 1993.

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

[QG89] J.-J. Quisquater and M. Girault, "$2n$-bit hash-functions using n-bit symmetric block cipher algorithms," *Advances in Cryptology, Proc. Eurocrypt'89, LNCS 434*, J.-J. Quisquater and J. Vandewalle, Eds., Springer-Verlag, 1990, pp. 102–109.

[Sch89] C.P. Schnorr, "Efficient identification and signatures for smart cards," *Advances in Cryptology, Proc. Crypto'89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 239–252.

[Yuv79] G. Yuval, "How to swindle Rabin," *Cryptologia*, Vol. 3, 1979, pp. 187–189.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.

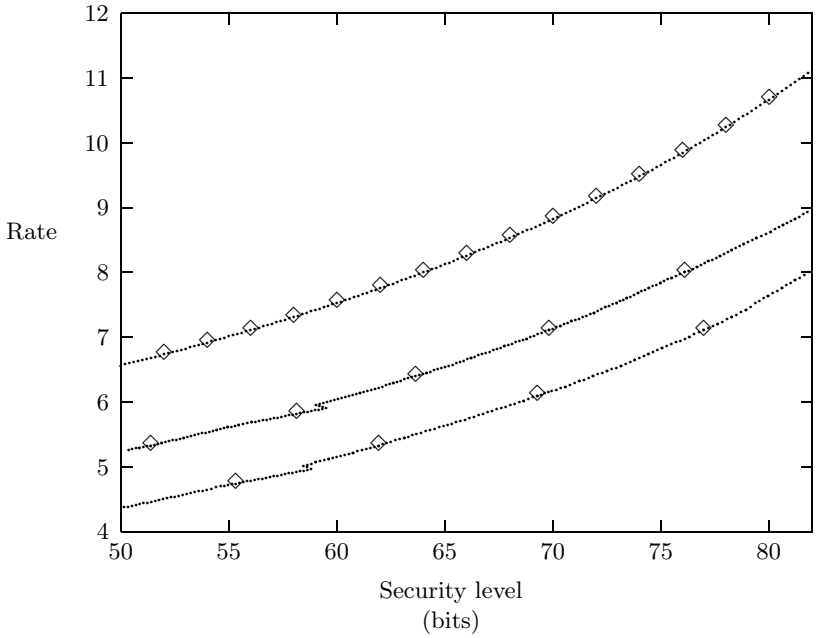

Fig. 3. Relation between the rate R and the security level S for $k = 4$ with parameter $\phi = 0$, 1, and 2.

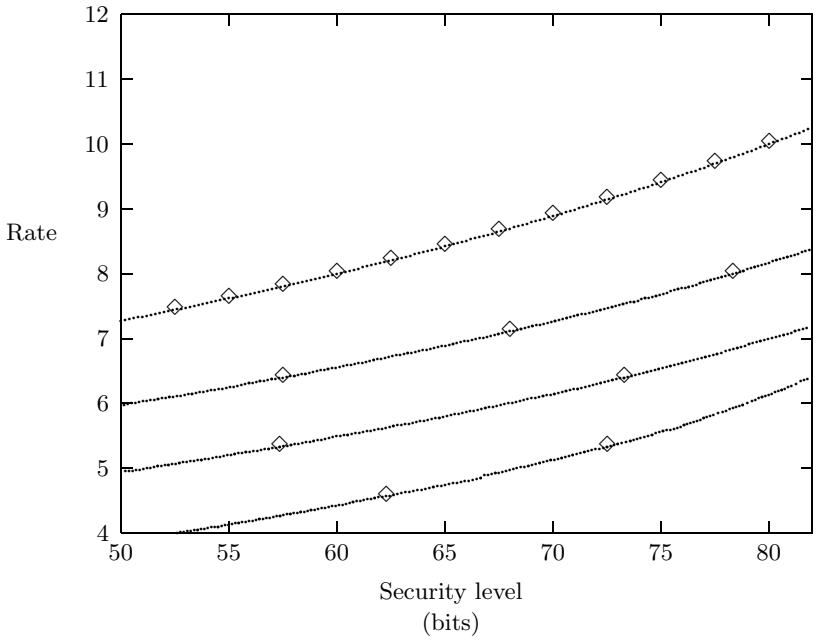

Fig. 4. Relation between the rate R and the security level S for $k = 5$ with parameter $\phi = 0$, 1, 2, and 3.

Appeared in *Advances in Cryptology – ASIACRYPT 1992*, Lecture Notes in Computer Science 718, J. Seberry, and Y. Zheng (eds.), Springer-Verlag, pp. 105–121.