COLLISION-FREE HASHFUNCTIONS BASED ON BLOCKCIPHER ALGORITHMS

Bart Preneel[1], Antoon Bosselaers, René Govaerts and Joos Vandewalle

[1] NFWO aspirant navorser, sponsored by the National Science Foundation of Belgium.

Katholieke Universiteit Leuven, Laboratorium ESAT
Kardinaal Mercierlaan, 94, B–3030 Heverlee, Belgium

Abstract

The concept of collision free hash functions has been shown to be a useful building block of signature schemes and message authentication schemes. In this paper, a fast and secure proposal is made for a $2n$–bit collison free hash function based on an n-bit encryption algorithm. In case of the DES, the length of the result is 128 bits, which suffices to thwart a birthday attack.

1. Introduction

Hash functions can be defined as a cryptographically secure method of computing a fixed length compression of a message. They play an important role in the design of *efficient* cryptographic protocols [26]. Although the requirements for a collision free hash function are well understood, many straightforward and apparently secure schemes have been broken with attacks based on algebraic manipulations or statistical principles.

Section 2 will discuss the application of hash functions to provide two security services, namely message integrity and non–repudiation of origin. A short overview of literature on hash functions will be given in section 3, followed by a discussion of attacks on those proposals. These attacks make it possible to define clearly the requirements for a collision free hash function. This enables us to come up with a new scheme, to give a critical analysis of its security and to compare it with a related proposal. Finally theoretical and practical conclusions are formulated.

2. Applications of Hash Functions

The development of public–key cryptosystems [12] created the possibility of generating efficient digital signatures. The technical term "non repudiation of origin" denotes a service whereby the recipient is given guarantee of the message's authenticity, in the sense that the recipient can subsequently prove to a third party that the message is authentic even if its originator subsequently revokes it.

The RSA public–key cryptosystem [35] was the first proposal of a public–key system with digital signature capability, as suggested by the title of the original paper. However, there are several reasons for compressing the message to a short imprint, prior to signing:

1. The size of the signature can be reduced to one block length, independent of the length of the signed message: this solves the dilemma between storage of message and signature or computing the message whenever it is needed.
2. Most known signature schemes are relatively slow, even with dedicated hardware.
3. Attacks against individual blocks are thwarted.
4. The algebraic structure of the message space can be destroyed (in case of RSA, the signature of the product of two messages equals the product of their signatures).
5. The reblocking problem can be avoided.

A second application of hash functions is the protection of information authentication. The best definition, in our opinion, was given by Simmons in [37]: information authentication is concerned with establishing the integrity of information purely on the basis of the internal structure of the information itself, irrespective of the source of that information. The price paid for an efficient approach is that the resulting scheme is only computationally secure [36]. This means that the security offered by the scheme depends on the fact that cheating, when using any of the known methods, requires an infeasible amount of computation, that is in principle possible. The idea is to add controlled redundancy to the information: the

presence of the redundancy distinguishes the authentic information from bogus information. This contrasts to the approach in [18], where modification of information is detected through the distortion of the internal reduncancy of the information. However, a universal algorithm has to protect the integrity of the information without any assumptions on the internal structure of the message.

Two major approaches for introducing redundancy can be distinguished. The first possibility is the compression of the information under control of a secret key. The resulting quantity will be called a "Message Authentication Code " (MAC). Subsequently the information and/or the MAC can be encrypted, but this is not necessary. This enables protection of the authenticity without secrecy. The authentication is not subliminal free [38]. The second approach consists of generating a "Manipulation Detection Code" (MDC) that is only function of the data to be checked. The required security component can be introduced in different ways, by encrypting MDC, or the information, or both. The advantage of the MDC are:

1. The computation requires only publicly known quantities. This results in a highly simplified key management.
2. The separation of the function of authentication from encryption. This implies that the authenthication is independent of the encryption algorithm or the mode of operation. In the context of the ISO Open System Interconnect Reference Model integrity and confidentiality can be protected at different layers.

The disadvantage of this approach is that in case of a compromise of the security of the encryption algorithm, the integrity protection is also jeopardized.

Finally a short note on terminology: the term hash function historically originated from computer science, where it denotes a function that allocates as uniformly as possible storage for the records of a file. The name hash function has also been adopted for cryptographically strong compression functions, but the result of the hash function, the controlled redundancy has been given a wide variety of names in literature: hash total, hash result, imprint, checksum, compressed encoding, seal, authenticator, authentication tag. The distinction between MAC and MDC, proposed by Jueneman [19], is certainly not perfect (a MAC or a MDC are actually no codes, and both can serve for message authentication), but the adoption of these terms offers a practical solution to the momentary "Babel of tongues".

3. Overview of Proposed Hash Functions

To minimize the implementation effort and the equipment complexity, designers of cryptographically secure hash functions tend to base their scheme on existing encryption algorithms. In this section a brief review is given of proposals for hash functions based on blockcipher algorithms and on modular arithmetic. The hash function is described as follows:

$$\begin{aligned} H_0 &= IV \\ H_i &= f(M_i, H_{i-1}) \qquad i = 1, 2, \ldots l \\ H &= H_l \end{aligned}$$

The information if necessary padded with zeroes, is divided in l blocks M_1 through M_l. The result of the hash function is denoted with H and IV is the abbreviation for Initial Value.

3.1 Modes of a blockcipher algorithm

For a first type of hash functions, the function f was chosen to imitate the CBC (Cipher Block Chaining), the CFB (Cipher Feedback), or the OFB (Output Feedback) mode of an encryption algorithm E with

$$C = E(K, P)$$

Here P denotes the plaintext, C the ciphertext and K the key. The key and initial value are public. Soon an improved variant was proposed: an extra message block M_{l+1} was added consisting of modulo 2^m ($m \geq 1$) sum of M_1 through M_l.

3.2 Invertible key chaining

A second type of hash functions is based on the fact that a good block cipher can withstand a chosen plaintext attack: this implies that it is computationally infeasible to determine the key even if a large number of chosen plaintext and ciphertext pairs are known. In this case, the function f is defined by

$$f = E(M_i \oplus s(H_{i-1}), H_{i-1})$$

with s any function. The first proposals for s were the zero function and the identity. Because all these schemes were considered to be insecure, two new proposals were made. The first one was repeating the message p times, the other one was repeating the whole scheme for p different initial values.

3.3 Non-invertible key chaining

A third type is also based on key chaining, but tries to avoid backward recurrence. Two proposals are [25]

$$f = E(s(H_{i-1}), M_i) \oplus M_i$$

and [41]

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

Some other schemes do not fit in this classification, but are not practical because they require a block cipher with a double key length (no widely accepted proposal is known) or consist of two encryptions and are insecure.

3.4 Modular arithmetic

A last type is based on modular arithmetic. A first proposal requires multiple precision arithmetic:

$$f = (H_{i-1} \oplus M_i)^d \bmod n$$

To avoid backwards recurrence, redundancy must be introduced ($M_{l+1} = M_1$). A special case ($d = 2$) has been studied in [13]. The recommendation was to introduce a 50 % redundancy.

The attempts of Jueneman [20,21,22,23] to avoid multiple precision arithmetic have failed to produce a 128 bit result with 32 bit arithmetic [24].

A last scheme originated by F. Cohen [4]:

$$f = [(1 + M_i) \bmod (H_{i-1} - 1)]^e \bmod m$$

has shown to be very weak. An new version [16] employing a secret key was recently broken by the authors [31] with a chosen message attack that enabled to calculate enough key bits to carry out all manipulations.

4. Methods of Attack on Hash Functions

An evaluation of an authentication protocol or a signature scheme strongly depends on the information at the disposal of an adversary, the actions he can undertake and finally on the consequences of both a successful and an unsuccessful attack.

According to the Kerckhoff principle, we assume that the scheme is public. An outsider knows theoretically only the hash result, but in practice it is very likely that corresponding plaintext–ciphertext pairs will be available. An insider knows the key of the encryption algorithm or is at least able to generate a large number of plaintext–ciphertext pairs. If the opponent knows the plaintext he wishes to subvert, the attack is greatly simplified. A short description is given of possible attacks.

4.1 Attacks independent of the algorithm

1. Random attack: The opponent selects a random message and hopes that the change will remain undetected. In case of a good MDC, his probability of succes equals $1/2^n$ with n the number of bits of the MDC. The feasability of this attack depends on the action taken in case of detection of an erroneous MDC, on the expected value of a successful attack and on the number of attacks that can be carried out.

2. Birthday attack: The idea behind this attack [42] is that for a group of 23 people the probability that at least two people have a common birthday exceeds 1/2. The adversary generates r_1 variations on a bogus message and r_2 variations on a genuine message. A second possibility is that he collects a large number of messages and is able to divide them in two categories. The probability of finding a bogus message and a genuine message that hash to the same result is given by

$$1 - \exp(-\frac{r_1 \cdot r_2}{2^n})$$

which is about 63 % when $r = r_1 = r_2 = 2^{\frac{n}{2}}$. The involved comparison problem does not require r^2 operations: after sorting the data, which requires $O(r \log r)$ operations, comparison is easy. Jueneman has shown in [22] that for $n = 64$ the processing and storage requirements are feasible in reasonable time with the computer power available in every large organisation. A time–memory–processor trade-off is possible. To avoid this attack with a reasonable safety margin, n should be at least 128 bits.

In case of digital signatures, a sender can attack his own signature or the receiver could offer the signer a message he's willing to sign and replace it later with the bogus message. Only the last attack can be thwarted through adding some random to a message just prior to signing. For information authentication, a birthday attack is only possible if the message is not encrypted or if the opponent can dispose of a large number of plaintext–ciphertext pairs.

4.2 Attacks dependent on the chaining

1. Meet in the middle attack: This attack is a variation on the birthday attack, but instead of the result of the hash function, intermediate chaining variables are compared. The attack enables an opponent to construct a message with a chosen hash result, which is not possible in case of a simple birthday attack. The opponent generates r_1 variations on the first part of a bogus message and r_2 variations on the last part. Starting from the initial value and going backwards from the result, the probability for a matching intermediate variable is given by the same formula. The attack can be thwarted by avoiding functions f that are invertible to the chaining variable H_{i-1} and to the message M_i.

2. Generalized meet in the middle attack: The previous attack was extended [5,14] to break the p–fold iterated schemes. It was shown that breaking these schemes would not require $O(2^{\frac{pn}{2}})$ but only $O(10^p \cdot 2^{\frac{n}{2}})$ operations. Here again, modest trade-offs between time, storage and processing are possible.

3. Correcting last block attack: This attack consists of substituting all blocks of the message except for the last one. This last block is then calculated such that the MDC takes a certain value. In spite of its name, the attack is not restricted to the last block. The hash functions based on modular arithmetic are especially sensitive to this attack. When redundancy is imposed on the message, it becomes computationally infeasible to find a correcting block with the necessary redundancy. The price paid for this solution is a decreased speed.

4. Analytical weaknesses: Some schemes allow manipulations as insertion, deletion, permutation and substitutions of blocks.

4.3 Attacks dependent on the blockcipher

Certain weaknesses of a blockcipher are not of great importance when it is used to protect the privacy, but can have dramatic consequences if the cipher is used in one of the special modes for hashing. These weaknesses can be exploited to insert special messages or carry out well chosen manipulations without changing the hash result. We will only discuss the weaknesses of the DES [2,30], because of its worldwide use.

1. Complementation property: One of the first properties that was known of the DES was the symmetry under complementation:

$$\forall P, K : C = DES(K, P) \Longleftrightarrow \overline{C} = DES(\overline{K}, \overline{P})$$

2. Weak keys: Another well known property of the DES is the existence of 4 weak keys. For these keys, encryption equals decryption, or DES is an involution. These keys are also called palindromic keys. This means that there exist 2^{32} values of P for which $DES(K_p, P) = P$. There exist also 6 pairs of semi–weak keys, for which $E(K_2, E(K_1, P)) = P$, $\forall P$. The anti–palindromic keys are 4 of these semi–weak keys with the property that there exist 2^{32} values of P for which $DES(K_{ap}, P) = \overline{P}$.

4.4 Key collisions

A recent breakthrough in the analysis of blockcipher algorithms was the presentation of a simple collision search algorithm by J.-J. Quisquater at Eurocrypt '89 [32]. A collision is a pair of keys K_1, K_2 such that $E(K_1, P) = E(K_2, P)$ for a plaintext P. The running time of the algorithm is $O(2^{\frac{n}{2}})$. The collision search is applicable to any blockcipher algorithm, but a good design of the hash function can make the collisions useless. On the other side, the collision can be used in some cases to invert the proposals of the category non–invertible chaining, which makes them vulnerable to a meet in the middle attack or to a substitution of the message with a collision.

4.5 High level attacks

Even if the above attacks would not be feasible, special care has to be taken to avoid replay of messages and construction of valid messages by cutting and splicing others. Attacks on this level can be thwarted by adding time stamps and serial numbers and through the use of sound cryptographic protocols.

5. Requirements for Hash Functions

At this point, we are able to state clearly which properties a hash function must satisfy to be useful for cryptographic applications.

1. The hash function must be *publicly known* and should not require any secret information for its operation.
2. The evaluation of the hash function must be *fast.*
3. The hash function must be *collision free:* this means that it is computationally infeasible to find two distinct messages which hash to the same result.

Note that this last property is stronger than the requirement of a one–way hash function: this means that given any possible hash result, it must be computationally infeasible to find a message that hashes to this result. In certain applications, where the adversary may not be able to use collisions he has found, the hash functions must satisfy a strong one–way property: given any possible message and corresponding hash result, it must be computationally infeasible to find a message that hashes to this result.

Considering the above attacks, we can state more specific requirements for a collision free hash function [22]:

1. The length of the result n must be 128 bits or more to avoid a birthday attack. Some authors state that 64 bits suffices for a strong one–way hash function. Others prefer to play for safety and impose also in this case the lower bound of 128 bits.

2. The probability that the hash result of two different texts will be equal is a uniformly distributed random variable, independent of the text with an average value of 2^{-n}.
3. The result must be sensible to deletion, insertion, and substitution of a message block and to permutation of two message blocks.
4. The hash function must not be invertible to avoid a meet in the middle attack nor subject to decomposition into separate and independent elements.

An important conclusion of the previous discussion is that all proposed schemes based on a 64 bit blockcipher are insecure. Only the schemes based on multiple precision modular arithmetic are secure if sufficient reduncancy is introduced. This redundancy results in a further slowing down of the scheme and involves a security risk: if more sophisticated attacks are developed, perhaps even more reduncancy is necessary. The fact that no widely accepted blockcipher with a blocklength and a key of 128 bit exists, forces us to look for another approach. Even when a proposal would be made now, some of the non–invertible key chaining proposals would be secure, but a thorough study of the algorithm and the development of fast hardware would be time–consuming. Therefore we propose the construction of a $2n$ bit hash function based on an n bit blockcipher.

6. Construction of a $2n$–bit Hash Function Based on an n–bit Blockcipher

In this section, we will give a description of our scheme and analyse it critically. First we define the function $R(X,Y) = E(X,Y) \oplus X \oplus Y$. Here $\oplus$ denotes the addition modulo 2.

$$\begin{aligned} H_1 &= IV_1 \\ H_2 &= IV_2 \\ H_{2i+1} &= c_3\left[M_{2i-1}, R\left(c_1\left[M_{2i-1}, M_{2i}\right], c_2\left[H_{2i-1}, H_{2i}\right]\right)\right] \\ H_{2i+2} &= c_4\left[M_{2i}, R\left(c_2\left[M_{2i-1}, H_{2i-1}\right], \left(c_1\left[M_{2i}, H_{2i}\right]\right)\right)\right] \\ & \quad i = 1, \ldots l \\ MDC &= H_{2l+1} || H_{2l+2} \end{aligned}$$

Here $||$ denotes the concatenation of the two strings. The functions c_1, c_2, c_3 and c_4 have to satisfy following conditions:

1. They have to compress two n bit variables to one n bit variable.
2. Their result must be uniform.
3. At least one of their output bits must change in case of the change of one input bit.

The choice of particular functions thwart attacks that exploit special properties of the blockcipher E. A different choice for c_1 and c_2 and for c_3 and c_4 might avoid certain symmetries.

In case of the DES, we will show that the simplest choice for the functions c_i, namely the addition modulo 2 results in a scheme with the same security level as the non–invertible key chaining but with a 128 bit result. The description can be simplified to

$$\begin{aligned} H_1 &= IV_1 \\ H_2 &= IV_2 \\ H_{2i+1} &= M_{2i-1} \oplus H_{2i-1} \oplus H_{2i} \\ &\quad \oplus DES(M_{2i-1} \oplus M_{2i}, H_{2i-1} \oplus H_{2i}) \\ H_{2i+2} &= M_{2i} \oplus H_{2i-1} \oplus H_{2i} \\ &\quad \oplus DES(M_{2i-1} \oplus H_{2i-1}, M_{2i} \oplus H_{2i}) \\ &\quad i = 1, \ldots l \\ MDC &= H_{2l+1} || H_{2l+2} \end{aligned}$$

Note that through symmetry properties analogous solutions are easily derived. The description is complete when we state that the 56 bits of the key are computed from the 64 bit input by omitting the parity bits.

The simple choice of the functions c_i avoids the attack of complementing one or more inputs without a change of the hash result: complementing 1, 2, 3, or 4 inputs of $\{M_{2i-1}, M_{2i}, H_{2i-1}, H_{2i}\}$ always results in a complementation of one of the outputs $\{H_{2i+1}, H_{2i+2}\}$. The function R is a simple combination of the non–invertible key chaining proposed by Winternitz [41] and Matyas et al. [25]. An opponent who wants to invert our hash function, has to solve the previous equations for the four inputs, given the two outputs. To solve the first equation, he has to assume values for the plaintext $H_{2i} \oplus H_{2i-1}$ and for M_{2i-1} to obtain the ciphertext. A search for key collision would result in the key $M_{2i-1} \oplus M_{2i}$ and thus in values for M_{2i-1} and for M_{2i}. To solve for the second equation, he has to assume values for the plaintext $M_{2i} \oplus H_{2i}$ and thus for H_{2i-1} and H_{2i}. All four inputs are fixed now. A key collision search will result in a value for $M_{2i-1} \oplus H_{2i-1}$, which is of course in general not equal to the fixed values. An extension of this attack would generate s collisions at each time, which increases the work for key collision search only with a factor $\log_2(s)$. The number of operations to invert the functions remains very large.

Because of the special structure in the equations, they can be simplified by taking the modulo 2 sum of the two equations and substituting $M_{2i-1} \oplus M_{2i}$

with K, $M_{2i-1} \oplus H_{2i-1}$ with K_1 and $M_{2i} \oplus H_{2i}$ with K_2:

$$H_{2i+1} \oplus H_{2i+2} = K \oplus DES(K, K \oplus K_1 \oplus K_2) \oplus DES(K_1, K_2)$$

After randomly choosing K_1 and K_2, this is an equation in K. The cryptanalyst is faced with the same problem as when he wants to go backwards in the case of non-invertible chaining. When $K = 0$, the equation is greatly simplified. Once a solution for K is obtained, M_{2i-1} is easily solved from the first equation. This simplification through modulo 2 addition can be prevented by taking an other choice for the functions c_3 and c_4.

Another proposal was presented by J.-J Quisquater and M. Girault at Eurocrypt '89 [33], also using modulo 2 additions. The simplification through summing the two equations is thwarted by adding the result of the first encryption to the plaintext of the second encryption. We pointed out that in case of the DES, replacing M_1 and M_2 with their complement has no effect on the final result. The scheme shows also some weaknesses when M_{2i} is a palindromic or a anti-palindromic key. In case the plaintext is a fixed or anti-fixed message, the scheme can be reduced to the simple equations of non-invertible chaining. Finally remark that the scheme is slower because the two encryptions can not be computed in parallel. However, Coppersmith has recently broken this scheme for every blockcipher due to linearities [34]. This points out that more complex schemes with more than one encryption for each message block might be necessary. A good choice for the functions c_i combined with the two stage approach can also offer a secure solution.

7. Tree Approach to Hash Functions

A collision free hash function based on a blockcipher algorithm is obtained with a special chaining mode. This can be considered as a Finite State Machine with state H_i and input M_i at time t_i.

$$H_i = f(M_i, H_{i-1})$$

To avoid a meet in the middle attack, it must be impossible to compute the previous state H_{i-1}, given the current state H_i and the input M_i. To avoid simple substitutions, it must be also difficult to compute the input M_i given the previous state H_{i-1} and the current state H_i. The conclusion is that f must be non-invertible to both arguments. This offers the possibility to define a parallel collision free hash function. We will only treat the simple case where $l = 2^k$ for some integer k:

$$\begin{aligned} H_i^1 &= f(M_{2i-1}, M_{2i}) \qquad i = 1, \ldots, 2^{k-1} \\ H_i^j &= f(H_{2i-1}^{j-1}, H_{2i}^{j-1}) \qquad i = 1, \ldots, 2^{k-j} \\ & \quad (j = 2, \ldots, k-1) \\ H &= f(H_1^{k-1}, H_2^{k-1}) \end{aligned}$$

The time to compute the result is $O(\log l)$ in stead of $O(l)$. To avoid permutation of message blocks, f may not be symmetric with respect to its arguments. Extensions of this approach are possible if l is no power of 2 or by combining the chaining and the tree approach [40].

8. Conclusions

The design of collision free hash functions is certainly a non-trivial task. We have tried to sketch the state of the art and to stress the importance of collision free hash functions for efficient cryptographic protocols. The design of hash functions based on blockcipher algorithms has the advantage that only one algorithm is required for encryption and authentication and that the available fast hardware can be optimally used.

Because of the variety of attacks that have emerged, and especially the recent key collision search, we do not claim to offer final solutions. A good choice for the functions c_i might be able to thwart these attacks, but it remains an open question whether a secure scheme exists with only one encryption for each message block. This clearly shows the need for a secure and efficient encryption algorithm with a blocklength and a key of 128 bits. This would greatly simplify the design of secure and efficient collision free hash functions. The theoretical approach of [6] seems also very promising.

Acknowledgements

The authors wish to thank J.-J. Quisquater and J. Van Mieghem for interesting discussions.

References

[1] S. G. Akl, "On the Security of Compressed Encodings", Advances in Cryptology Proc. Crypto '83, New York: Plenum Press, pp. 209–230.

[2] "American National Standard for Data Encryption Algorithm (DEA).", X3.92-1981, ANSI, New York.

[3] "American National Standard for Financial Institution Message Authentication (Wholesale)", X9.9-1986 (Revised), ANSI, New York.

[4] F. Cohen, "A cryptographic checksum for integrity protection", Computers & Security, Vol 6., pp. 505–510, 1988.

[5] D. Coppersmith, "Another Birthday Attack", Advances in Cryptology, Proc. Crypto '85, Springer Verlag, pp. 14-17.

[6] I. Damgård, "Collision Free Hash Functions and Public Key Signature Schemes", Advances in Cryptology, Proc. Crypto '87, Springer Verlag, pp. 203–216.

[7] D. Davies, "A Message Authenticator Algorithm Suitable For a Mainframe Computer", Advances in Cryptology, Proc. Crypto '84, Springer Verlag, pp. 393–400.

[8] D. Davies, W. L. Pryce, "Security for Computer Networks", Wiley & Sons, 1984.

[9] Y. Desmedt, "Unconditionally Secure Authentication Schemes and Practical and Theoretical Consequences", Advances in Cryptology, Proc. Crypto '85, Springer Verlag, pp. 42-55.

[10] Y. Desmedt, J. Vandewalle, R. Govaerts, "The Mathematical Relation between the Economic, Cryptographic and Information Theoretical Aspects of Authentication", presented at the Fourth Symposium on Information Theory in the Benelux, Haasrode, Belgium, 26-27 May 83.

[11] Y. Desmedt, F. Hoornaert and J.-J. Quisquater, "Several Exhaustive Key Search Machines and the DES.", Abstracts of Eurocrypt '86.

[12] W. Diffie, M.E. Hellman, "New directions in cryptography", IEEE Trans. Informat. Theory, IT–22, Vol. 6, p. 644–654, Nov. 1976.

[13] M. Girault, "Hash-functions Using Modulo-n Operations", Advances in Cryptology, Proc. Eurocrypt '87, Springer Verlag, pp. 217–226.

[14] M. Girault, R. Cohen and M. Campana, "A Generalized Birthday Attack", Advances in Cryptology, Proc. Eurocrypt '88, Springer Verlag, pp. 129–156.

[15] M. Hellman, R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig and P. Schweitzer, "Results of an Initial Attempt to Cryptanalyze the NBS Data Encryption Standard.", Information Systems Lab., Dept. of Electrical Eng., Stanford Univ., 1976.

[16] Y. Huang and F. Cohen, "Some Weak Points of One Fast Cryptographic Checksum Algorithm and Its Improvements", Computers & Security, Vol. 7, pp. 503–505, 1988.

[17] "Banking – Requirements for Message Authentication (Wholesale)", DIS 8730, APCS, London, July 1987.

[18] C.J.A. Jansen and D.E. Boekee, "Modes of Blockcipher Algorithms an their Protection Against Active Eavesdropping", Advances in Cryptology, Proc. Eurocrypt'87, Springer Verlag, pp. 327-347.

[19] R.R. Jueneman, "Analysis of Certain Aspects of Output Feedback Mode", Advances in Cryptology, Proc. Crypto '82, Plenum Press, New York, pp. 99–127.

[20] R.R. Jueneman, S.M.Matyas, C.H. Meyer, "Message Authentication with Manipulation Detection Codes.", Proc. of the 1983 IEEE Symposium on Security and Privacy, pp. 33-54, 1984.

[21] R.R. Jueneman, S.M. Matyas, C.H. Meyer, "Message Authentication", IEEE Comm. Mag., Vol. 23, No. 9, pp. 29-40, 1985.

[22] R.R. Jueneman, "A High Speed Manipulation Detection Code", Advances in Cryptology, Proc. Crypto '86, Springer Verlag, pp. 327-347.

[23] R.R. Jueneman, "Electronic Document Authentication", IEEE Network Mag., Vol. 1, No. 2, pp. 17-23, 1987.

[24] R.R. Jueneman, personal communciation.

[25] 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, pp. 5658–5659, 1985.

[26] R. Merkle, "Secrecy, Authentication, and Public Key Systems", UMI Research Press, 1979.

[27] C.H. Meyer and S.M. Matyas, Cryptogr-"phy: a New Dimension in Data Security", Wiley & Sons, 1982.

[28] J.H. Moore and G.J. Simmons, "Cycle Structure of the DES for Keys having Palindromic (or Antipalindromic) Sequences of Round Keys.", IEEE Trans. Software Eng., Vol. 13, pp. 262-273, 1987.

[29] J.H. Moore, "Protocol Failures in Cryptosystems", Proc. IEEE, Vol. 76, No. 5, pp. 594-602, 1988.

[30] "Data Encryption Standard (DES).", National Bureau of Standards (USA), FIPS 46, April 1977.

[31] B. Preneel, A. Bosselaers, R. Govaerts and J. Vandewalle, " A Chosen Text Attack on the Modified Cryptographic Checksum Algorithm of Cohen and Huang.", paper in preparation.

[32] J.-J. Quisquater and J.-P. Delescaille, "How Easy is Collision Search ? Application to DES.", Abstracts Eurocrypt '89.

[33] J.-J. Quisquater and M. Girault, "2n-bit Hash Functions Using n-bit Symmetric Block Cipher Algorithms.", Abstracts Eurocrypt '89.

[34] J.-J. Quisquater, personal communication.

[35] R.L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", CACM, Vol. 21, pp. 120–126, February 1978.

[36] G.J. Simmons, "The Prisoners' Problem and the Subliminal Channel", Advances in Cryptology, Proc. Crypto '83, New York: Plenum Press, pp. 51–67.

[37] G.J. Simmons, "A Natural Taxonomy for Digital Information Authentication Schemes", Advances in Cryptology, Proc. Crypto '87, Springer Verlag, pp. 269–288.

[38] G.J. Simmons, "A Survey of Information Authentication", Proc. IEEE, Vol. 76, No. 5, pp. 603–620, 1988.

[39] M.E. Smid and D.K. Branstad, "The Data Encryption Standard: Past and Future", Proc. IEEE, Vol. 76, No. 5, pp. 550-559, 1988.

[40] K. Van Espen, J. Van Mieghem, " Evaluatie en Implementatie van Authentiseringsalgoritmen Evaluation and Implementation of Authentication Algorithms – (in Dutch)", ESAT Laboratories, Katholieke Universiteit Leuven, Thesis grad. eng. 1989.

[41] S.R. Winternitz, "Producing a One-Way Hash Function from DES.", Advances in Cryptology, Proc. Crypto '83, New York: Plenum Press, pp. 203–207.

[42] G. Yuval, "How to Swindle Rabin.", Cryptologia, Vol. 3, pp. 187–189, 1979.