ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

CRYPTOGRAPHIC HASH FUNCTIONS –A SURVEY

Prof. Pawan Kumar*[1]& Dr. Harsh Dev[2]
*[1]Army Institute of Management & Technology(AIMT), Greater Noida,(Uttar Pradesh),India
[2]Pranveer Singh Institute Of Technology (PSIT), Kanpur (Uttar Pradesh),India

Keywords: *Hash functions,encryption,Digital signature,MD-4,SHA-1,SHA-2,SHA-3 etc*

ABSTRACT

The use of cryptography started from late 1970s and became more prominent in 1980s.Commercial use of cryptograghy started in late 1990s.Many organization started using cryptographic tools for information security but many security challenges were faced by the organizations.The cryptographic designs were having more security flaws.The use of cryptography functions started from MD5 and SHA-1.Now we are going to enter into digital era ,therefore it is very important to discuss the role of cryptographic functions in our day to day activities. Cryptographic functions are used for encryption, digital signatures, secure hashing, message (data) authentication codes, key management, entity authentication, password, and random number generation etc. This paper explains the history of the usage ,design,concept, and the applications of hash functions.

HISTORY OF CRYPTOGRAPHY

There are two methods of providing security to the message ,one method is steganography and another is cryptography.In stenography,we generally conceal the message from enemy and in cryptography we cipher the message.In ancient times ,steganography was used by the kings to send their messages.

a) "Herodotus relates that one Histauaeus shaved the head of his messenger,wrote the message on his scalp, and waited for the hair to regrow. On reaching his destination, the messenger shaved his head again and the recipient,Aristogoras, read the message."

b) b"Invisible ink comes into this category; the recipient develops the message by applying heat or chemicals to it.

c) "*Cryptography* refers to the art of protecting transmitted information from unauthorized interception or tampering. The other side of the coin, *cryptanalysis*, is the art of breaking such secret ciphers and reading the information, or perhaps replacing it with different information. Sometimes the term *cryptology* is used to include both of these aspects. Historically, the term "cryptography" has been associated with the problem of designing and analyzing *encryption schemes* (i.e., schemes that provide secret communication over insecure communication media). However, since the 1970s, problems such as constructing unforgeable *digital signatures* and designing *fault-tolerant protocols* have also been considered as falling within the domain of cryptography. In fact, cryptography can be viewed as concerned with the design of any system that needs to withstand malicious attempts to abuse it."

INTRODUCTION

Cryptography has many aspects but in this paper,we keep our discussion restricted to cryptographic hash functions.It plays a very fundamental role in modern cryptography.In cryptographic hash functions,larger domains are mapped to smaller ranges in which it takes input text and produces message digest/hash value/hash result/hash code.

For a domain D and range R with , the function is many-to-one,where collision is unavoidable.But restricting h to a domain of t-bit inputs (), if h were "random" in the sense that all outputs were essentially equiprobable, then about inputs would map to each output, and two randomly chosen inputs would yield the same output with probability (independent of t).

Definition- A *hash function* (in the unrestricted sense) is a function h which has, as a minimum, the following two properties:

1. *Compression* — h maps an input of arbitrary finite bit length, to an output h() of fixed bit length n.

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

2. E*ase of computation*—given h and an input , h() is easy to compute.

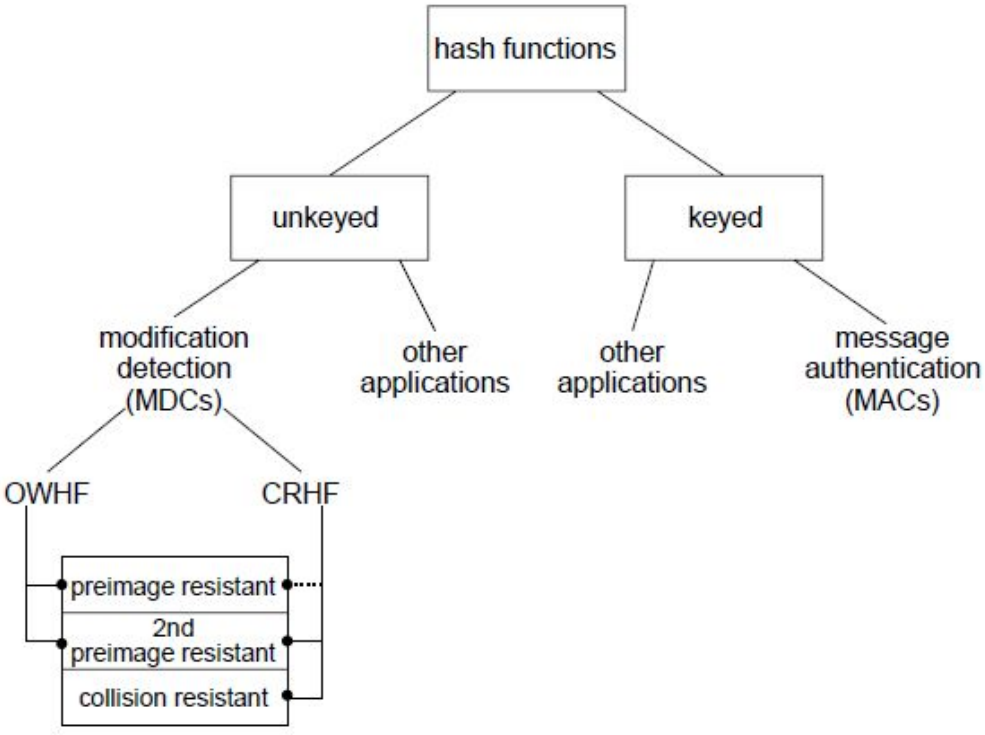
CLASSIFICATION OF HAS FUNCTIONS

There are two classes of hash functions –unkeyed and keyed hash functions.In unkeyed hash functions only single message is the input and in keyed hash function message along with secret key are used as input.Both produces fixed length hash output.

The functional classification of hash functions are follows
A) **Modification detection codes (MDCs)**:MDCs are a subclass of *unkeyed* hash functions and further divided into following types:

a) ***One-way hash functions*** **(OWHFs)**: for these, finding an input which hashes to a pre-specified hash-value is difficult;
b) ***Collision resistant hash functions*** **(CRHFs)**: for these, finding any two inputs having the same hash-value is difficult.

B) ***Message authentication codes (MACs)***- MACs have two functionally distinct parameters, a message input and a secret key; they are a subclass of *keyed* hash function.MAC algorithms are used for data integrity,authentication and identification in symmetric key schemes.



BASIC PROPERTIES OF HASH FUNCTIONS

For any unkeyed hash function h with inputs x, x0 and outputs y, y0.the following properties are as follows:

A) *Preimage resistance*—for essentially all pre-specified outputs, it is computationally infeasible to find any input which hashes to that output, i.e., to find any preimage ch that h() = y when given any y forwhich a corresponding input is not known.

B)*2nd-preimage resistance*—it is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find a 2nd-preimage such that h(x) = h().

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

C)Collision resistance—it is computationally infeasible to find any two distinct inputs x, which hash to the same output, i.e., such that h(x) = h().

A *message authentication code (MAC)* algorithm is a family of functions parameterized by a secret key k, with the following properties:

i)E*ase of computation* — for a known function hk, given a value k and an input x, is easy to compute. This result is called the *MAC-value* or *MAC*.
ii)C*ompression*— maps an input x of arbitrary finite bitlength to an output of fixed bitlength n.Furthermore, given a description of the function family h, for every fixed allowable value of k (unknown to an adversary).
iii)C*omputation-resistance*—given zero ormore text-MAC pairs (, it is computationally infeasible to compute any text-MAC pair (; for any new input x (including possibly for = for some i).

The following attack scenarios thus exist for MACs,for adversaries with increasing advantages:

i. *Known-text attack*. One or more text-MAC pairs pairs (are available.
ii. *Chosen-text attack*. One or more text-MAC pairs pairs (are available for chosen by the adversary.
iii. *Adaptive chosen-text attack*. The xi may be chosen by the adversary as above, now allowing successive choices to be based on the results of prior queries.

Hash functions are often used in applications which require the one-way property, but not compression. It is, therefore, useful to distinguish three classes of functions (based on the relative size of inputs and outputs):

i. G*eneral hash functions*. These are functions typically with additional one-way properties, which compress arbitrary-length inputs to n-bit outputs.
ii. C*ompression functions* (fixed-size hash functions). These are functions typically with additional one-way properties, but with domain restricted to fixed-size inputs – i.e., compressing m-bit inputs to n-bit outputs,m>n.

N*on-compressing one-way functions*. These are fixed-size hash functions as above,except that n = m. These include *one-way permutations*, and can be more explicitly described as computationally non-invertible functions.

BITSIZES REQUIRED FOR PRACTICAL SECURITY

i. For a OWHF, is required. Exhaustive off-line attacks require at most 2n operations; this may be reduced with precomputation .
ii. For a CRHF, is required. Birthday attacks are applicable.
iii. For a MAC, along with a MAC key of 64-80 bits is sufficient for most applications and environments.

The features of unkeyed hash functions are tabulated below:

↓Hash function	n	m	Preimage	Collision	Comments
Matyas-Meyer-Oseas[a]	n	n	2^n	$2^{n/2}$	for keylength $= n$
MDC-2 (with DES)[b]	64	128	$2 \cdot 2^{82}$	$2 \cdot 2^{54}$	rate 0.5
MDC-4 (with DES)	64	128	2^{109}	$4 \cdot 2^{54}$	rate 0.25
Merkle (with DES)	106	128	2^{112}	2^{56}	rate 0.276
MD4	512	128	2^{128}	2^{20}	Remark 9.50
MD5	512	128	2^{128}	2^{64}	Remark 9.52
RIPEMD-128	512	128	2^{128}	2^{64}	–
SHA-1, RIPEMD-160	512	160	2^{160}	2^{80}	–

[a]The same strength is conjectured for Davies-Meyer and Miyaguchi-Preneel hash functions.
[b]Strength could be increased using a cipher with keylength equal to cipher blocklength.

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

Name	Bitlength	Rounds × Steps per round	Relative speed
MD4	128	3 × 16	1.00
MD5	128	4 × 16	0.68
RIPEMD-128	128	4 × 16 twice (in parallel)	0.39
SHA-1	160	4 × 20	0.28
RIPEMD-160	160	5 × 16 twice (in parallel)	0.24

Name	String	Hash value (as a hex byte string)
MD4	""	31d6cfe0d16ae931b73c59d7e0c089c0
	"a"	bde52cb31de33e46245e05fbdbd6fb24
	"abc"	a448017aaf21d8525fc10ae87aa6729d
	"abcdefghijklmnopqrstuvwxyz"	d79e1c308aa5bbcdeea8ed63df412da9
MD5	""	d41d8cd98f00b204e9800998ecf8427e
	"a"	0cc175b9c0f1b6a831c399e269772661
	"abc"	900150983cd24fb0d6963f7d28e17f72
	"abcdefghijklmnopqrstuvwxyz"	c3fcd3d76192e4007dfb496cca67e13b
SHA-1	""	da39a3ee5e6b4b0d3255bfef95601890afd80709
	"a"	86f7e437faa5a7fce15d1ddcb9eaeaea377667b8
	"abc"	a9993e364706816aba3e25717850c26c9cd0d89d
	"abcdefghijklmnopqrstuvwxyz"	32d10c7b8cf96570ca04ce37f2a19d84240d3a89
RIPEMD-160	""	9c1185a5c5e9fc54612808977ee8f548b2258d31
	"a"	0bdc9d2d256b3ee9daae347be6f4dc835a467ffe
	"abc"	8eb208f7e05d987a9b044a8e98c6b087f15a0bfc
	"abcdefghijklmnopqrstuvwxyz"	f71c27109c692c1b56bbdceb5b9d2865b3708dbc

HASH FUNCTIONS ARE BASED ON THE MODULAR ARITHMETIC

The iterated hash functions are designed through the modular arithmetic using mod M which is the basis of compression functions.It is generally applied for reuse of software and hardware.

Mash

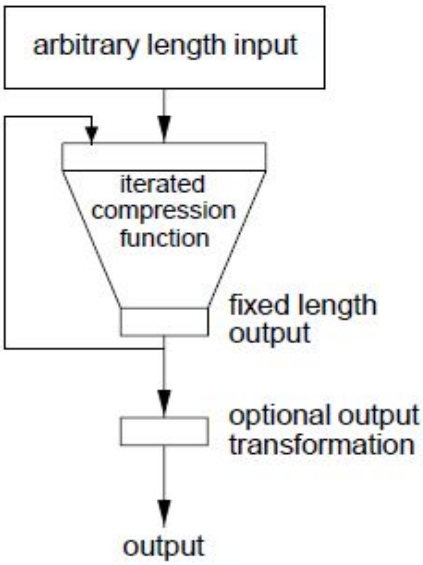
MASH-1 (*Modular Arithmetic Secure Hash, algorithm 1*) is a hash function based on modular arithmetic. It has been proposed for inclusion in a draft ISO/IEC standard. MASH-1involves use of an RSA-like modulus M, whose bitlength affects the security. M should be difficult to factor, and for M of unknown factorization, the security is based in part on the difficulty of extracting modular roots . The bitlength of M also determines the blocksize for processing messages, and the size of the hash-result (e.g., a 1025-bit modulus yields a 1024-bit hash-result). As a recent proposal, its security remains open to question. Techniques for reducing the size of the final hash-result have also been proposed,but their security is again undetermined as yet.

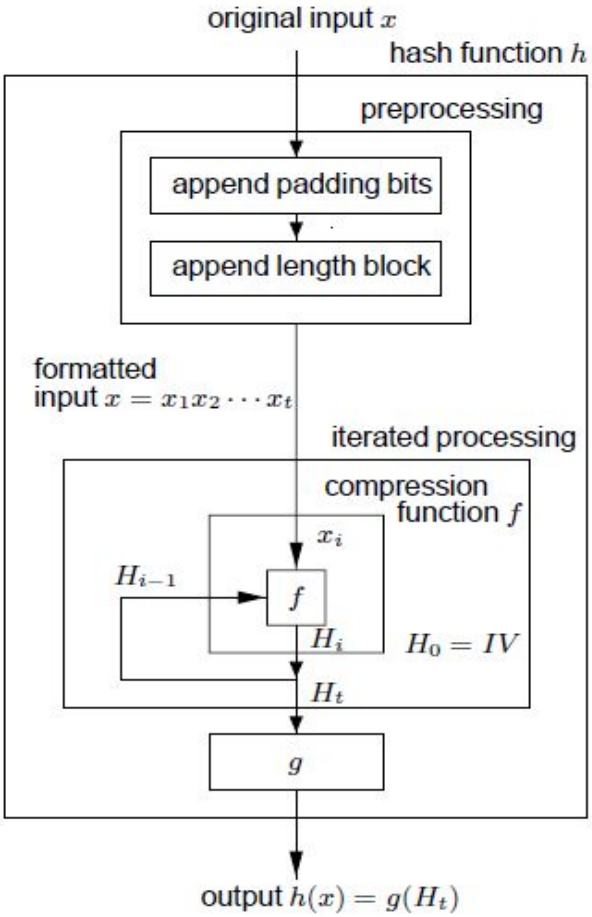
ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

BASIC CONSTRUCTION OF HASH FUNCTIONS





SECURITY OBJECTIVES AND BASIC ATTACKS

The basic attacks and security strength of the hash functions are given in the table.

Hash type	Design goal	Ideal strength	Adversary's goal
OWHF	preimage resistance; 2nd-preimage resistance	2^n 2^n	produce preimage; find 2nd input, same image
CRHF	collision resistance	$2^{n/2}$	produce any collision
MAC	key non-recovery; computation resistance	2^t $P_f = \max(2^{-t}, 2^{-n})$	deduce MAC key; produce new (msg, MAC)

Advanced attacks on hash functions are as follows:

Birthday attacks

Algorithm-independent attacks are those which can be applied to any hash function, treating it as a black-box whose only significant characteristics are the output bitlength n (and MAC key bitlength for MACs), and the running time for one hash operation. It is typically assumed the hash output approximates a uniform random variable. Attacks falling under this category include those based on hash-result bitsize ; exhaustiveMAC key search.

Pseudo-collisions and compression function attacks

The exhaustive or brute force methods produces preimages, 2nd-preimages,and collisions for hash functions, are always theoretically possible. They are not considered true "attacks" unless the number of operations required is

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

significantly less than both the strength conjectured by the hash function designer and that of hash functions of similar parameters with ideal strength. An attack requiring such a reduced number of operations is informally said to *break* the hash function,whether or not this computational effort is feasible in practice. Any attack method which demonstrates that conjectured properties do not hold must be taken seriously; when this occurs, one must admit the possibility of additional weaknesses.

Chaining attacks

Chaining attacks are those which are based on the iterative nature of hash functions and, in particular, the use of chaining variables.

Recent Developments in Hash family

The NIST SHA-3 Competition NIST had called open call for contribution for SHA-3 on November-2, 2007 which is new cryptographic hash functions. The main objective of developing SHA-3 was to substitute SHA-2 with hash hash result of 224,256,384 and 512 so that devices using SHA-2 could easily be compatible for SHA-3.NIST received 64 submissions for SHA-3 out of that only 51 algorithms was selected for first round. On july 24,2009,NIST selected 14 algorithms for 2nd round namely Blake, Blue Midnight Wish, CubeHash, ECHO, Fugue, Gr0stl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD and Skein. Most of these designs used iterated approach and four candidates (Blue Midnight Wish, Gr0stl, Sha-bal, and SIMD) used a modification of the Merkle-Damgard construction with a larger internal memory, also known as a wide-pipe construction, and three use the HAIFA approach (Blake, ECHO, and SHAvite-3).The hash functions Blue Midnight Wish, Cube Hash, Blake and Skein are of the ARX (Addition, Rotate, XOR) type; they derive their non-linearity from the carries in the modular addition.MD6 of rivest was not selected because of slower performance security threats was by differential attacks. The reader is referred to the SHA-3 Zoo and eBASH for security and performance updates; these sites are maintained by the ECRYPT II project. five finalists – BLAKE, Grøstl, JH, Keccak and Skein were selected in December 2010 to advance to the third and final round of the competition. Based on the public comments and internal review of the candidates, NIST announced Keccak as the winner of the SHA-3 Cryptographic Hash Algorithm Competition on October 2, 2012, and ended the five-year competition.

CONCLUSION

We are talking about digital and cashless economy in India. The first and foremost challenges will be to protect IT infrastructure .The cryptography plays very important roles in protection and security to the infrastructure. The hash algorithms like MD-family and SHA-0,SHA-1 are under attack and soon SHA-1 is going to be replaced by SHA-2.Therefore ,more research is required to provide good security and designing features to hash function in SHA-3 family.

REFERENCES

1. *S.M. Bellovin, E.K. Rescorla, "Deploying a new hash algorithm," Proceedings of the Network and Distributed System Security Symposium, NDSS 2006, The Internet Society, 2006.*
2. *R. Benadjila, O. Billet, S. Gueron, M.J.B. Robshaw, "The Intel AES instructions set and the SHA-3 candidates," Advances in Cryptology, Proceedings Asiacrypt 09, LNCS 5912, M. Matsui, Ed., Springer-Verlag, 2009, pp. 162-178.*
3. *G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, "On the indifferentiability of the sponge construction," Advances in Cryptology, Proceedings Eurocrypt'08, LNCS 4965, N. Smart, Ed., Springer-Verlag, 2008, pp. 181-197.*
4. *E. Biham, A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard," Springer-Verlag, 1993.*
5. *Handbook of Applied Cryptography, by A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996. www.cacr.math.uwaterloo.ca/hac*
6. *E. Biham, O. Dunkelman, "A framework for iterative hash functions - HAIFA," Proceedings Second NIST Hash Functions Workshop 2006, Santa Barbara (CA), USA, August 2006.*
7. *A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, A. Shamir, "Key recovery attacks of practical complexity on AES variants with up to 10 rounds," IACR Eprint 2009/374, 19 Aug. 2009.*
8. *A. Biryukov, D. Khovratovich, "Related-key cryptanalysis of the full AES-192 and AES-256," Advances in Cryptology, Proceedings Asiacrypt 09, LNCS 5912, M. Matsui, Ed., Springer-Verlag, 2009, pp. 1-18.*
9. *J. Black, P. Rogaway, T. Shrimpton, "Black-box analysis of the block-cipher-based hash-function constructions from PGV," Advances in Cryptology, Proceedings Crypto'02, LNCS 2442, M. Yung, Ed., Springer-Verlag, 2002, pp. 320-355.*

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

10. C. Bouillaguet, O. Dunkelman, P.-A. Fouque, A. Joux, "On the security of iterated hashing based on forgery-resistant compression functions," IACR Eprint 2009/077,6 Feb. 2009.
11. B.O. Brachtl, D. Coppersmith, M.M. Hyden, S.M. Matyas, C.H. Meyer, J. Oseas,S. Pilpel, M. Schilling, "Data Authentication Using Modification Detection Codes Based on a Public One Way Encryption Function," U.S. Patent Number 4,908,861, March 13, 1990.
12. F. Chabaud, A. Joux, "Differential collisions: an explanation for SHA-1," Advances in Cryptology, Proceedings Crypto 98, LNCS 1462, H. Krawczyk, Ed., Springer-Verlag, 1998, pp. 56-71.
13. D.X. Charles, E.Z. Goren, K.E Lauter, "Cryptographic hash functions from expander graphs," Proceedings Second NIST Hash Functions Workshop 2006, Santa Barbara (CA), USA, August 2006.
14. S. Contini, A.K. Lenstra, R. Steinfeld, "VSH, an efficient and provable collision-resistant hash function," Advances in Cryptology, Proceedings Eurocrypt 06,LNCS 4004, S. Vaudenay, Ed., Springer-Verlag, 2006, pp. 165-182.
15. D. Coppersmith, "Analysis of ISO/CCITT Document X.509 Annex D," IBM T.J. Watson Center, Yorktown Heights, N.Y., 10598, Internal Memo, June 11, 1989, (also ISO/IEC JTC1/SC20/WG2/N160).
16. J.-S. Coron, Y Dodis, C. Malinaud, and P. Puniya, "Merkle-Damgard revisited: how to construct a hash function," Advances in Cryptology, Proceedings Crypto'05, LNCS 3621, V. Shoup, Ed., Springer-Verlag, 2005, pp. 430-448.
17. I.B. Damgard, "Collision free hash functions and public key signature schemes," Advances in Cryptology, Proceedings Eurocrypt'87, LNCS 304, D. Chaum and W.L. Price, Eds., Springer-Verlag, 1988, pp. 203-216.
18. I.B. Damgard, "A design principle for hash functions," Advances in Cryptology, Proceedings Crypto'89, LNCS 435, G. Brassard, Ed., Springer-Verlag, 1990,pp. 416-427.
19. R.D. Dean, "Formal aspects of mobile code security," PhD thesis, Princeton University, January 1999.
20. C. De Canniere, C. Rechberger, "Preimages for reduced SHA-0 and SHA-1," Advances in Cryptology, Proceedings Crypto'08, LNCS 5157, D. Wagner, Ed.,Springer-Verlag, 2008, pp. 179-202.
21. FIPS 180-2, "Secure Hash Standard," Federal Information Processing Standard (FIPS), Publication 180-2, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., August 26, 2002 (Change notice 1 published on December 1, 2003).
22. P. Gauravaram, L.R. Knudsen, "On randomizing hash functions to strengthen the security of digital signatures," Advances in Cryptology, Proceedings Eurocrypt 08,LNCS 5479, A. Joux, Ed., Springer-Verlag, 2009, pp. 88-105.
23. M. Grassl, I. Ilic, S. Magliveras, R. Steinwandt, "Cryptanalysis of the Tillich-Zemor hash function," IACR Eprint 2009/376, 30 Jul. 2009.
24. M.E. Hellman, "A cryptanalytic time-memory trade-off," IEEE Trans. on Information Theory, Vol. IT-26, No. 4, 1980, pp. 401-406.
25. S. Hirose, "Some plausible constructions of double-block-length hash functions," Fast Software Encryption'06, LNCS 4047, M. Robshaw, Ed., Springer-Verlag, 2006,pp. 210-225.
26. H. Imai, A. Yamagishi, "Cryptrec," in Encyclopedia ofCryptography and Security, H.C.A. van Tilborg, Ed., 2005, pp. 119-123.
27. M.K. Franklin, Ed., Springer-Verlag, 2004, pp. 306-316.
28. A. Joux, T. Peyrin, "Hash functions and the (amplified) boomerang attack," Advances in Cryptology, Proceedings Crypto 07, LNCS 4622, A. Menezes, Ed.,Springer-Verlag, 2007, pp. 244-263.
29. B.S. Kaliski Jr., "The MD2 Message-Digest algorithm," Request for Comments (RFC) 1319, Internet Activities Board, Internet Privacy Task Force, April 1992.
30. J. Kelsey, T. Kohno, "Herding hash functions and the Nostradamus attack," Advances in Cryptology, Proceedings Eurocrypt 06, LNCS 4004, S. Vaudenay, Ed.,Springer-Verlag, 2006, pp. 183-200.
31. J. Kelsey, B. Schneier, "Second preimages on n-bit hash functions for much less than 2n work," Advances in Cryptology, Proceedings Eurocrypt'05, LNCS 3494, R. Cramer, Ed., Springer-Verlag, 2005, pp. 474-490.
32. L.R. Knudsen, X. Lai, B. Preneel, "Attacks on fast double block length hash functions," Journal of Cryptology, Vol. 11, No. 1, Winter 1998, pp. 59-72.
33. L.R. Knudsen, J.E. Mathiassen, F. Muller, S.S. Thomsen, "Cryptanalysis ofMD2,"Journal of Cryptology, 2010, 19 pp., in print.
34. X. Lai, J.L. Massey, "Hash functions based on block ciphers," Advances in Cryp-tology, Proceedings Eurocrypt 92, LNCS 658, R.A. Rueppel, Ed., Springer-Verlag,1993, pp. 55-70.