The First 30 Years of Cryptographic Hash Functions and the NIST SHA-3 Competition

Bart Preneel

Katholieke Universiteit Leuven and IBBT
Dept. Electrical Engineering-ESAT/COSIC,
Kasteelpark Arenberg 10 Bus 2446, B-3001 Leuven, Belgium
bart.preneel@esat.kuleuven.be

Abstract. The first designs of cryptographic hash functions date back to the late 1970s; more proposals emerged in the 1980s. During the 1990s, the number of hash function designs grew very quickly, but for many of these proposals security flaws were identified. MD5 and SHA-1 were deployed in an ever increasing number of applications, resulting in the name "Swiss army knifes" of cryptography. In spite of the importance of hash functions, only limited effort was spent on studying their formal definitions and foundations. In 2004 Wang et al. perfected differential cryptanalysis to a point that finding collisions for MD5 became very easy; for SHA-1 a substantial reduction of the security margin was obtained. This breakthrough has resulted in a flurry of research, resulting in new constructions and a growing body of foundational research. NIST announced in November 2007 that it would organize the SHA-3 competition, with as goal to select a new hash function family by 2012. From the 64 candidates submitted by October 2008, 14 have made it to the second round. This paper presents a brief overview of the state of hash functions 30 years after their introduction; it also discusses the progress of the SHA-3 competition.

1 Early History and Definitions

Cryptographic hash functions map input strings of arbitrary (or very large) length to short fixed length output strings. In their 1976 seminal paper on public-key cryptography [31], Diffie and Hellman identified the need for a one-way hash function as a building block of a digital signature scheme. The first definitions, analysis and constructions for cryptographic hash functions can be found in the work of Rabin [74], Yuval [99], and Merkle [60] of the late 1970s. Rabin proposed a design with a 64-bit result based on the block cipher DES [37], Yuval showed how to find collisions for an n-bit hash function in time $2^{n/2}$ with the birthday paradox, and Merkle's work introduced the requirements of collision resistance, second preimage resistance, and preimage resistance. In 1987, Damgård [26] formalized the definition of collision resistance, and two years later Naor and Yung defined a variant of seoncd preimage resistant functions called Universal One Way Hash Functions (UOWHFs) [66] (also known as functions

offering eSEC [79]). In 2004 Rogaway and Shrimpton [79] formally studied the relations between collision resistance and several flavors of preimage resistance and second preimage resistance. Hash functions should also destroy the algebraic structure of the signature scheme; typical examples are the Fiat-Shamir heuristic [36] and Coppersmith's attack on the hash function in X.509 Annex D [24] that was intended for use with RSA [77] (this attack breaks the signature scheme by constructing message pairs (x, x') for which $h(x) = 256 \cdot h(x')$). This development resulted in the requirement that hash functions need an 'ideal' behavior which would allow them to instantiate the theoretical concept of random oracles (see e.g. Bellare and Rogaway [10]). Constructions of MAC algorithms based on hash functions (such as HMAC) have resulted in the requirement that the hash function can be used to construct pseudo-random functions, which has a.o. been studied by Bellare et al. [8, 6].

This paper is organized as follows. Section 2 describes brute force attacks and generic constructions for iterated hash functions, while Sect. 3 gives an overview of three types of hash function constructions. Section 4 presents the status of NIST's SHA-3 competition 1 year after the submission deadline and presents the planning for the future. Our concluding remarks are presented in Sect. 5. As cryptographic hash functions have become a rich subject, we don't attempt to be complete in this short contribution. We mostly provide some pointers to the literature, with an emphasis on very early work and on the most recent results.

2 Generic Analysis and Design

2.1 Brute Force Attacks

For an ideal hash function with a hash result of bitlength n, finding a (second) preimage takes $\Theta(2^n)$ evaluations of the hash function. However, if one considers multiple targets, then the expected cost to find a (second) preimage for one of these 2^t targets is reduced to $\Theta(2^{n-t})$ (note that for $t = n/2$ this corresponds to $\Theta(2^{n/2})$). If one intends to find a (second) preimage for all 2^t targets, one can apply Hellman's time-memory tradeoff [42]: after a precomputation of $\Theta(2^n)$, additional (second) preimages can be found at a cost of $\Theta(2^{2n/3})$; this method requires a storage of $\Theta(2^{2n/3})$. Wiener provides a detailed analysis in the full cost model [96]. The answer to this degradation in security is to parameterize the hash function with a salt (also known as spice, tweak or key) [60], so that each application can get a different function.

For an n-bit hash function, collisions can be found in time $\Theta(2^{n/2})$; there exist algorithms with low memory that are highly parallellizable [92]. This shows that for long term collision resistance (10 years or more), a hash result of 192 or 256 bits is required.

In practice, collision resistance is much harder to achieve than (second) preimage resistance. Simon [84] also proved that there is no black box reduction from preimage resistance to collision resistance. Fortunately, only few applications need collision resistance: the most notable ones are digital signatures (where

either the signer or the verifier can cheat) and binding commitments. It is important however to understand that circumventing the requirement of collision resistance is harder than expected (see for example the attack on the RMX mode in [40]).

2.2 Iterated Hash Functions

From the first designs (including the Rabin function [74]), it was understood that a hash function h should be constructed by iterating a compression function f with fixed size inputs. The input is first padded such that the length of the input is multiple of the block length. Next it is divided into t blocks x_1 through x_t. The hash result is then computed as follows:

$$H_0 = IV \tag{1}$$

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

$$h(x) = g(H_t)\,. \tag{3}$$

Here IV is the abbreviation of *Initial Value*, H_i is called the chaining variable, and the function g is called the output transformation. While many MAC algorithms have an output transformation, this is a relatively new feature for hash functions. However, it is easy to see that the absence of an output transformation leads to an extension attack, that is, one can compute $h(x||y)$ from $h(x)$ and y (without knowing x), which is undesirable for some applications.

In two articles presented at Crypto'89, Damgård [27] and Merkle [61] show under which conditions collision resistance of the compression function f is sufficient to obtain collision resistance of the function h. The standard way to satisfy these conditions is to fix the IV and to append the message length at the end; Lai and Massey [54] coined the name Merkle-Damgård strengthening for this construction.[1] Naor and Yung [66] obtained similar results for Universal One-Way Hash Functions, which is the eSEC variant of a second preimage resistant hash function. Lai and Massey [54] present a necessary and sufficient condition for ideal second preimage resistance of an iterated hash function (that is, finding a second preimage takes about 2^n evaluations of the compression function f); unfortunately later on their result turned out to be incorrect.

During the last five years, a number of limitations have been identified for these iterated constructions, for example the work on long-message second preimages by Dean [28] and Kelsey and Schneier [51], the multicollisions by Joux [47] and the herding attack by Kelsey and Kohno [50]. The (surprising) implication of the multicollision attack is that the concatenation of two iterated hash functions ($h(x) = h_1(x)||h_2(x)$) is as most as strong as the strongest of the two; more precisely, if the result of h_i has bitlength n_i, the cost of a collision attack on h is at most $n_1 \cdot 2^{n_2/2} + 2^{n_1/2}$ (here we assume w.l.o.g. that $n_1 \leq n_2$). This

[1] Some authors refer to any linear iterated hash function as described above as "the Merkle-Damgård construction," which is clearly not appropriate since this approach dates back to the earlier work by Rabin in 1978 [74].

complexity is much lower than one would expect intuitively, that is $2^{(n_1+n_2)/2}$. On the other hand, a large number of improvements have been proposed to these constructions including work by Andreeva et al. (ROX [2]), Bellare and Ristenpart (EMD [9]), Biham and Dunkelman (HAIFA [15], see also [19]), and Yasuda [98]. Maurer et al. [59] generalize the concept of indistinguishability to indifferentiability from random oracles. Coron et al. [25] have studied how the Merkle-Damgård construction can be modified to satisfy indifferentiability from random oracles. Other work in this direction can be found in [13, 65].

Merkle has introduced the so-called Merkle trees [60] for constructing digital signature schemes. Damgård has shown that the domain of a collision resistant compression function can also be extended by a tree construction [27]; an optimized version was proposed by Pal and Sarkar [68]. While the tree construction offers increased parallelism, it has the unfortunate property that for every size of the tree one obtains a different hash function, which is undesirable from an interoperability point of view.

3 Hash Function Constructions

During the 1980s, the need for an efficient and secure hash function was well understood (see for example the note presented at Eurocrypt'86 [70]). In the late 1980s and early 1990s a large number of designs was created; about 50 proposals were known in 1993, but and at least two thirds of them were broken (see the PhD thesis of the author for the status at that time [71]). After fifteen years of cryptanalysis, very few of those early schemes remain secure. Since then, about hundred new hash function designs have been proposed; 64 of these have been submitted to the SHA-3 competition (cf. Sect. 4). Many of them have not survived for long either.

Next we describe the status of the three main classes of hash functions: hash functions based on block ciphers, hash functions based on modular arithmetic and dedicated hash functions.

3.1 Hash Functions Based on Block Ciphers

The first constructions for hash functions were all based on block ciphers, more in particular based on DES [37]. This approach has several advantages: the design and evaluation effort of a block cipher can be reused, and one may obtain very compact implementations. However, it may well be that a block cipher has weaknesses in the key schedule which have only very limited impact on its use for encryption, but which may be undesirable when it is used in a hash function construction. Examples are the weak keys of DES [64] and the key schedule weaknesses of AES-192 and AES-256 [16, 17].

After cryptanalysis of several proposals, a more systematic approach for cryptanalysis has been used by Preneel et al. [72] and for security proofs in the ideal cipher model by Winternitz [97], Black et al. [18] and Stam [87]. The more difficult problem is how to construct hash functions with a result that is

larger than the block length, since most block ciphers have a block length of 64 or 128 bits, which is clearly not sufficient to obtain collision resistance. This area turned out to be very difficult; substantial progress has been made from the point of view of cryptanalysis (e.g. Knudsen et al. [52]) and design (e.g. MDC-2 [20, 88], Merkle [61] and Hirose [43]). Recent work by Rogaway and Steinberger [80] and Stam [86] has studied constructions based on permutations. It is fair to state that we are improving our understanding of the problem on how to construct hash functions from small building blocks; on the other hand, it is not clear that the most efficient hash functions can be designed by starting from a block cipher.

3.2 Hash Functions Based on Arithmetic Primitives

Public key cryptology, and in particular modular arithmetic, has also been a source of inspiration for hash function constructions. This has resulted in hash functions with a security proof based on number theoretic assumptions such as factoring and discrete logarithm. One example is the construction by Bellare et al. [8] based on the discrete logarithm problem in a group of large prime order. An interesting construction is VSH [23], for which finding collisions is provably related to factoring; however, due to structural properties identified a.o. by Saarinen [81], VSH does not have the properties expected from a general purpose hash function. In the area of 'ad hoc' constructions, a large number of proposals was broken; eventually MASH-1 and MASH-2 were standardized in ISO/IEC 10118-4 [46]; they use squaring and raising to the power $2^8 + 1$ respectively. Schemes based on additive or multiplicative knapsacks offer attractive performance results. However, in spite of theoretical support (e.g. Ajtai's work [1]), practical constructions have not fared well until now: see for example the attack by Patarin [69] on an additive knapsack scheme, the attack by Tillich and Zémor [90] on the LPS hash function [22] and the cryptanalysis by Grass et al. [41] of the 1994 scheme of Tillich and Zémor [89].

3.3 Dedicated Hash Functions

The limitations of block cipher based hash functions resulted in a series of designs from scratch. These hash functions were among the first algorithms to be designed to be efficient in software on microprocessors rather than in hardware implementations. The Binary Condensing Algorithm [91] and MD2 of Rivest [49] use 8-bit to 8-bit S-boxes, while N-Hash [63] is based on 8-bit additions. The first 32-bit proposals date back to the beginning of the 1990s and include MD4 [75], MD5 [76] and Snefru [62]. Around the same time, differential cryptanalysis of block ciphers was developed by Biham and Shamir [14]; they applied these techniques to cryptanalyze N-hash and Snefru.

MD5 was proposed by Rivest in 1991 as a strengthened version of MD4. As it was optimized for software implementations, MD5 was about 10 times faster than DES in software. Moreover, MD5 was available without any licenses and it was easier to export than an encryption algorithm. As a consequence, MD5

was adopted very quickly in many applications.[2] Unfortunately, weaknesses were identified early on: in 1992, den Boer and Bosselaers [30] found collisions for the compression function and in 1996, Dobbertin found collisions for MD5 but with a random IV rather than the fixed IV from the specifications [32]; his attack combined differential attacks with techniques such as continuous approximations and genetic programming. In 2004, Wang et al. [93–95] made a breakthrough with enhanced differential attacks that combine improved differential paths with clever message modification techniques. Optimized versions of their attacks can find collisions for MD5 in milliseconds [85] and collisions for MD4 by hand. It is important to point out that MD4 and MD5 have a 128-bit result: this implies that a brute force collision search with a budget of US$ 100,000 would find a collision in a few days [92]. In spite of these weaknesses, it was still unexpected to some that Sotirov et al. [85] announced on December 31, 2008 that they managed to create a rogue CA certificate using MD5; such a certificate makes it possible to impersonate any website on the Internet. While their attack required some cryptanalytic improvements (as CAs insert a serial number into the message before signing), the main surprise seems that more than four years after the announcements by Wang et al., the most popular CAs had not yet removed MD5 from their offerings.

NIST (National Institute for Standards and Technology, USA) was apparently not confident in the security of MD5 and proposed in 1993 a strengthened version of it called SHA (Secure Hash Algorithm) with a 160-bit result; it is now frequently called SHA-0. In 1995, NIST discovered a certificational weakness in SHA-0 (no details were published), which resulted in a new release of the standard published under the name SHA-1 [38]. In 2002, NIST published three new hash functions with longer hash results that are commonly called SHA-2: SHA-256, SHA-384 and SHA-512 [39]. In December 2003, SHA-224 has been added in a change notice to [39]. In 1998 Chabaud and Joux [21] showed how collisions for SHA-0 can be found in 2^{61} steps compared to 2^{80} for a brute force attack. Wang et al. [93, 95] present a major improvement in 2005 by showing that finding a collision for SHA-0/SHA-1 takes only $2^{39}/2^{69}$ steps. The best collision attack for SHA-0 by Manuel and Peyrin [57] takes only 2^{33} steps. For SHA-1 the situation is more complex: at least four teams have announced improved collision attacks with complexity between 2^{52} and 2^{63}; however, at this stage no one has found a collision and there is some doubt about the complexities of these attacks. On the other hand, Joux and Peyrin have found collisions for 70 (out of 80) steps of SHA-1 in time 2^{39} (4 days on a PC) [48].

There are still some older proposals that have withstood cryptanalysis, such as RIPEMD-160 [33] and Whirlpool [5] (both designs have been included in ISO 10118 [46], together with SHA-1 and SHA-2); for the most recent status of attacks on Whirlpool, see [55]. Moreover, early cryptanalysis of the SHA-2 family suggests that this second generation functions has a substantial security margin against collision attacks (the results by Indesteege et al. [45] and Sanadhya and Sarkar [82] can only break 24 out of 64 steps of SHA-256).

[2] In 2005, there were about 800 uses of MD5 in Microsoft Windows.

However, the breakthrough collision attacks on MD5 and SHA-1 have resulted in a serious concern about the robustness of our current hash functions. With the exception of the recent rogue CA attack of [85], the practical impact of these attacks has so far been rather limited, as most applications rely on (second) preimage resistance rather than collision resistance. For MD2, Knudsen et al. [53] find preimages in time 2^{73}. Leurent [56] has shown that preimages for MD4 can be found in 2^{102} steps, and Sasaki and Aoki have developed a shortcut preimage attack for MD5 [83] with complexity 2^{123}. Preimage attacks for SHA-1 seem to be completely beyond reach today: the best attack by Aoki et al. [4] works for 48 out of 80 steps). Somewhat surprisingly, preimages for SHA-256 can be found faster than brute force for 43 out of 64 steps [3].

In view of these developments, the cryptographic community agrees that we need new hash functions that offer an adequate security margin for the next 30 years or more; in view of this it would be prudent to develop alternatives for SHA-2. This has motivated NIST to call for an open competition; this is a procedure commonly used in cryptography, a.o. for the block ciphers DES and AES; there were also the European competitions NESSIE [73] and eSTREAM [78] as well as the Japanese Cryptrec initiative [44]. While the industry is currently migrating to SHA-256 as a replacement for MD5 and SHA-1, some players seem to be waiting for SHA-3.

4 The NIST SHA-3 Competition

After two open workshops and a public consultation period, NIST has published on November 2, 2007 an open call for contributions for SHA-3, a new cryptographic hash family [67]. The deadline for the call for contributions was October 31, 2008. A SHA-3 submission needs to support hash results of 224, 256, 384 and 512 bits to allow substitution for the SHA-2 family. It should work with legacy applications such as DSA and HMAC. Designers should present detailed design documentation, including a reference implementation, optimized implementations for 32-bit and 64-bit machines; they should also evaluate hardware performance. If an algorithm is selected, it needs to be available worldwide without royalties or other intellectual property restrictions.

Even if preparing a submission required a substantial effort, NIST received 64 submissions. Early December 2008, NIST has announced that 51 designs have been selected for the first round. Five of the 13 rejected designs have been published by their designers (see [35]); it is perhaps not surprising that four of these five designs have been broken very quickly. From the 51 Round 1 candidates, about half were broken in early July 2009. This illustrates that designing a secure and efficient hash function is a challenging task.

On July 24, 2009, NIST announced that 14 algorithms have been selected for Round 2, namely Blake, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD and Skein. By mid September 2009, several of these algorithms have been tweaked, which means that small modifications have been made that should not invalidate earlier analysis. The

majority of these designs use an iterated approach as described in Sect. 2.2 or a variant thereof: four Round 2 candidates (Blue Midnight Wish, Grøstl, Shabal, and SIMD) use a modification of the Merkle-Damgård construction with a larger internal memory, also known as a wide-pipe construction, and three use the HAIFA approach [15] (Blake, ECHO, and SHAvite-3). Five candidates (CubeHash, Fugue, Hamsi, Keccak, and Luffa) use a (variant of a) sponge construction [13]. Several designs (ECHO, SHAvite-3, Fugue, and Grøstl) employ AES-based building blocks; the first two benefit substantially from the AES instructions that will be offered in the 2010 Intel Westmere processor (see [12] for details). The hash functions Blue Midnight Wish, CubeHash, Blake and Skein are of the ARX (Addition, Rotate, XOR) type; they derive their non-linearity from the carries in the modular addition.

About half the Round 1 candidates originate from Europe, one third from North America, and one in six from Asia; two designs are from the Southern Hemisphere. Note that this is only an approximation as some algorithms have designers from multiple components and some designers have moved. A very large part of the Round 1 cryptanalysis was performed by researchers in Europe. In Round 2, 9 out of 16 (64%) of the designs are European, while 3 are from North America and 2 from Asia.

Two designs were expected for Round 2 but did not make it. MD6 by Rivest was probably not selected because of the slower performance; moreover, an error was found by the designer in the proof of security against differential attacks. Lane was probably removed because of the rebound attack on its compression function in [58]; it should be pointed out that this attack has a very high memory complexity, which makes it questionable whether it is more efficient than a brute force attack.

Two designs in Round 1 had remarkable security results: SWIFFT admits an asymptotic proof of security against collision and preimage attacks under worst-case assumptions about the complexity of certain lattice problems; the collision and preimage security of FSE can be reduced to hard problems in coding theory. However, both designs are rather slow; moreover, they require additional building blocks to achieve other security properties.

It is notably difficult to make reliable performance comparisons; all the Round 2 candidates have a speed that varies between 5 and 35 cycles per byte. It should be pointed out that due to additional implementation efforts, the best current SHA-2 implementations have a speed of about 15 cycles/byte; it will thus become more difficult for SHA-3 to be faster than SHA-2. 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 [35].

The following tentative time line has been announced for the remainder of the competition: NIST intends to select approximately 5 finalists in Q4 of 2010. The third and final conference will take place in early 2012; it will be followed by an announcement of the decision in Q2 of 2012. Overall, it seems that there are many interesting candidates and the review and selection process will be

extremely challenging. As a consequence of this competition, both the theory and practice of hash functions will make a significant step forward.

5 Concluding Remarks

During the last five years, we have seen a cryptographic meltdown in the security of widely used hash functions. Fortunately the practical implications have been limited, as most applications rely on (second) preimage resistance rather than on collision resistance. However, we have learned that upgrading cryptographic algorithms is always more difficult than anticipated. This is surprising, since in software implementations cryptographic algorithms are typically negotiated during the first phase of the protocol; Bellovin and Rescorla [11] explain the shortcomings of TLS in this context.

We can only regret that SHA-1 was not designed with 128 or 160 steps instead of 80; this would have avoided many of the problems we face today. While RIPEMD-160 seems a more secure alternative, its adoption is still limited: most users are upgrading to SHA-256, because of the longer hash result.

During the last five years, the theory and practice of cryptographic hash functions has advanced substantially. In view of this, one can expect that the SHA-3 competition will result in a robust hash function with a good performance. It is essential that the selection is not driven too much by performance; sufficient attention should be paid to the assurance in the security evaluation (that is, how easy or hard is the analysis of the design). Finally, note that (except for some tweaks), the design of SHA-3 will reflect the state of the art in 2008, rather than the state of the art in 2012.

For the long term, we face the challenging problem to design an efficient hash function for which the security can be reduced to a mathematical problem that is elegant and/or better understood.

References

1. M. Ajtai, "Generating hard instances of lattice problems," *Proceedings 28th ACM Symposium on the Theory of Computing*, 1996, pp. 99–108.
2. E. Andreeva, G. Neven, B. Preneel, T. Shrimpton, "Seven-property-preserving iterated hashing: ROX," *Advances in Cryptology, Proceedings Asiacrypt'07, LNCS 4833*, K. Kurosawa, Ed., Springer-Verlag, 2007, pp. 130–146.
3. K. Aoki, J. Guo, K. Matusiewicz, Y. Sasaki, L. Wang, "Preimages for step-reduced SHA-2," *Advances in Cryptology, Proceedings Asiacrypt'09, LNCS 5912*, M. Matsui, Ed., Springer-Verlag, 2009, pp. 578–597.
4. K. Aoki, Y. Sasaki, "Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1," *Advances in Cryptology, Proceedings Crypto'09, LNCS 5677*, S. Halevi, Ed., Springer-Verlag, 2009, pp. 70–89.
5. P.S.L.M. Barreto and V. Rijmen, "The Whirlpool hashing function," NESSIE submission, September 2000.
6. M. Bellare, "New proofs for NMAC and HMAC: security without collision-resistance," *Advances in Cryptology, Proceedings Crypto'06, LNCS 4117*, C. Dwork, Ed., Springer-Verlag, 2006, pp. 602–619.

7. M. Bellare, R. Canetti, H. Krawczyk, "Keying hash functions for message authentication," *Advances in Cryptology, Proceedings Crypto'96, LNCS 1109*, N. Koblitz, Ed., Springer-Verlag, 1996, pp. 1–15.
8. M. Bellare, O. Goldreich, S. Goldwasser, "Incremental cryptography: the case of hashing and signing," *Advances in Cryptology, Proceedings Crypto'94, LNCS 839*, Y. Desmedt, Ed., Springer-Verlag, 1994, pp. 216–233.
9. M. Bellare, T. Ristenpart, "Multi-property-preserving hash domain extension and the EMD transform," *Advances in Cryptology, Proceedings Asiacrypt'06, LNCS 4284*, X. Lai and K. Chen, Eds., Springer-Verlag, 2006, pp. 299–314.
10. M. Bellare, P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," *ACM Conference on Computer and Communications Security* ACM, 1993, pp. 62–73.
11. 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.
12. 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.
13. 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.
14. E. Biham, A. Shamir, *"Differential Cryptanalysis of the Data Encryption Standard,"* Springer-Verlag, 1993.
15. E. Biham, O. Dunkelman, "A framework for iterative hash functions – HAIFA," *Proceedings Second NIST Hash Functions Workshop 2006*, Santa Barbara (CA), USA, August 2006.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.

24. 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).
25. J.-S. Coron, Y Dodis, C. Malinaud, and P. Puniya, "Merkle-Damgård revisited: how to construct a hash function," *Advances in Cryptology, Proceedings Crypto'05, LNCS 3621*, V. Shoup, Ed., Springer-Verlag, 2005, pp. 430–448.
26. I.B. Damgård, "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.
27. I.B. Damgård, "A design principle for hash functions," *Advances in Cryptology, Proceedings Crypto'89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 416–427.
28. R.D. Dean, *"Formal aspects of mobile code security,"* PhD thesis, Princeton University, January 1999.
29. C. De Cannière, 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.
30. B. den Boer, A. Bosselaers, "Collisions for the compression function of MD5," *Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765*, T. Helleseth, Ed., Springer-Verlag, 1994, pp. 293–304.
31. W. Diffie, M.E. Hellman, "New directions in cryptography," *IEEE Trans. on Information Theory*, Vol. IT–22, No. 6, 1976, pp. 644–654.
32. H. Dobbertin, "The status of MD5 after a recent attack," *CryptoBytes*, Vol. 2, No. 2, Summer 1996, pp. 1–6.
33. H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD," *Fast Software Encryption'96, LNCS 1039*, D. Gollmann, Ed., Springer-Verlag, 1996, pp. 71–82.
    See also `http://www.esat.kuleuven.ac.be/~bosselae/ripemd160`.
34. Y. Dodis, T. Ristenpart, T. Shrimpton, "Salvaging Merkle-Damgård for practical applications," *Advances in Cryptology, Proceedings Eurocrypt'08, LNCS 5479*, A. Joux, Ed., Springer-Verlag, 2009, pp. 371–388.
35. ECRYPT II, The SHA-3 Zoo, `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.
36. A. Fiat, A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," *Advances in Cryptology, Proceedings Crypto'86, LNCS 263*, A.M. Odlyzko, Ed., Springer-Verlag, 1987, pp. 186–194.
37. FIPS 46, *"Data Encryption Standard,"* Federal Information Processing Standard, NBS, U.S. Department of Commerce, January 1977 (revised as FIPS 46-1(1988); FIPS 46-2(1993), FIPS 46-3(1999)).
38. FIPS 180-1, *"Secure Hash Standard,"* Federal Information Processing Standard (FIPS), Publication 180-1, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., April 17, 1995.
39. 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).
40. 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.
41. M. Grassl, I. Ilic, S. Magliveras, R. Steinwandt, "Cryptanalysis of the Tillich-Zémor hash function," *IACR Eprint 2009/376*, 30 Jul. 2009.

42. M.E. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Trans. on Information Theory*, Vol. IT–26, No. 4, 1980, pp. 401–406.
43. 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.
44. H. Imai, A. Yamagishi, "Cryptrec," in *Encyclopedia of Cryptography and Security*, H.C.A. van Tilborg, Ed., 2005, pp. 119–123.
45. S. Indesteege, F. Mendel, B. Preneel, C. Rechberger, "Collisions and other non-random properties for step-reduced SHA-256," *Selected Areas in Cryptology – SAC 2008, LNCS 5381*, R. Avanzi, L. Keliher, and F. Sica, Eds., Springer-Verlag, 2009, pp. 276–293.
46. ISO/IEC 10118, *"Information technology – Security techniques – Hash-functions, Part 1: General"*, 2000, *"Part 2: Hash-functions using an n-bit block cipher algorithm,"* 2000, *"Part 3: Dedicated hash-functions,"* 2003. *"Part 4: Hash-functions using modular arithmetic,"* 1998.
47. A. Joux, "Multicollisions in iterated hash functions. Application to cascaded constructions," *Advances in Cryptology, Proceedings Crypto'04, LNCS 3512*, M.K. Franklin, Ed., Springer-Verlag, 2004, pp. 306–316.
48. 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.
49. B.S. Kaliski Jr., "The MD2 Message-Digest algorithm," *Request for Comments (RFC) 1319*, Internet Activities Board, Internet Privacy Task Force, April 1992.
50. 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.
51. J. Kelsey, B. Schneier, "Second preimages on n-bit hash functions for much less than 2^n work," *Advances in Cryptology, Proceedings Eurocrypt'05, LNCS 3494*, R. Cramer, Ed., Springer-Verlag, 2005, pp. 474–490.
52. 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.
53. L.R. Knudsen, J.E. Mathiassen, F. Muller, S.S. Thomsen, "Cryptanalysis of MD2," *Journal of Cryptology*, 2010, 19 pp., in print.
54. X. Lai, J.L. Massey, "Hash functions based on block ciphers," *Advances in Cryptology, Proceedings Eurocrypt'92, LNCS 658*, R.A. Rueppel, Ed., Springer-Verlag, 1993, pp. 55–70.
55. M. Lamberger, F. Mendel, C. Rechberger, V. Rijmen, M. Schläffer, "Rebound distinguishers: results on the full Whirlpool compression function," *Advances in Cryptology, Proceedings Asiacrypt'09, LNCS 5912*, M. Matsui, Ed., Springer-Verlag, 2009, pp. 126–143.
56. G. Leurent, "MD4 is not one-way," *Fast Software Encryption'08, LNCS 5086*, K. Nyberg, Ed., Springer-Verlag, 2008, pp. 412–428.
57. S. Manuel, T. Peyrin, "Collisions on SHA-0 in one hour," *Fast Software Encryption'08, LNCS 5086*, K. Nyberg, Ed., Springer-Verlag, 2008, pp. 16–35.
58. K. Matusiewicz, M. Naya-Plasencia, I. Nikolic, Y. Sasaki, M. Schläffer, "Rebound attack on the full Lane compression function," *Advances in Cryptology, Proceedings Asiacrypt'09, LNCS 5912*, M. Matsui, Ed., Springer-Verlag, 2009, pp. 106–125.
59. U.M. Maurer, R. Renner, C. Holenstein, "Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology," *Theory of Cryptography Conference, TCC 2004, LNCS 2951*, M. Naor, Ed., Springer-Verlag, 2004, pp. 21–39.

60. R. Merkle, *"Secrecy, Authentication, and Public Key Systems,"* UMI Research Press, 1979.
61. R. Merkle, "One way hash functions and DES," *Advances in Cryptology, Proceedings Crypto'89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 428–446.
62. R. Merkle, "A fast software one-way hash function," *Journal of Cryptology*, Vol. 3, No. 1, 1990, pp. 43–58.
63. S. Miyaguchi, M. Iwata, K. Ohta, "New 128-bit hash function," *Proceedings 4th International Joint Workshop on Computer Communications*, Tokyo, Japan, July 13–15, 1989, pp. 279–288.
64. J.H. Moore and G.J. Simmons, "Cycle structure of the DES for keys having palindromic (or antipalindromic) sequences of round keys," *IEEE Transactions on Software Engineering*, Vol. 13, 1987, pp. 262–273.
65. Y. Naito, K. Yoneyama, L. Wang, K. Ohta, "How to confirm cryptosystems security: the original Merkle-Damgård is still alive!" *Advances in Cryptology, Proceedings Asiacrypt'09, LNCS 5912*, M. Matsui, Ed., Springer-Verlag, 2009, pp. 382–398.
66. M. Naor, M. Yung, "Universal one-way hash functions and their cryptographic applications," *Proceedings 21st ACM Symposium on the Theory of Computing*, 1990, pp. 387–394.
67. NIST SHA-3 Competition, `http://csrc.nist.gov/groups/ST/hash/`.
68. P. Pal, P. Sarkar, "PARSHA-256 – A new parallelizable hash function and a multithreaded implementation," *Fast Software Encryption'03, LNCS 2887*, T. Johansson, Ed., Springer-Verlag, 2003, pp. 347–361.
69. J. Patarin, "Collisions and inversions for Damgård's whole hash function," *Advances in Cryptology, Proceedings Asiacrypt'94, LNCS 917*, J. Pieprzyk and R. Safavi-Naini, Eds., Springer-Verlag, 1995, pp. 307–321.
70. D. Pinkas, "The need for a standardized compression algorithm for digital signatures," *Abstracts of Papers: Eurocrypt 1986, A Workshop on the Theory and Application of Cryptographic Techniques*, I. Ingemarsson, Ed., 20-22 May 1986, p. 7.
71. B. Preneel, *"Analysis and design of cryptographic hash functions,"* Doctoral Dissertation, Katholieke Universiteit Leuven, 1993.
72. B. Preneel, R. Govaerts, J. Vandewalle, "Hash functions based on block ciphers: a synthetic approach," *Advances in Cryptology, Proceedings Crypto'93, LNCS 773*, D. Stinson, Ed., Springer-Verlag, 1994, pp. 368–378.
73. B. Preneel, "NESSIE project," in *Encyclopedia of Cryptography and Security*, H.C.A. van Tilborg, Ed., 2005, pp. 408–413.
74. M.O. Rabin, "Digitalized signatures," in *Foundations of Secure Computation,* R. Lipton and R. DeMillo, Eds., Academic Press, New York, 1978, pp. 155-166.
75. R.L. Rivest, "The MD4 message digest algorithm," *Advances in Cryptology, Proceedings Crypto'90, LNCS 537*, S. Vanstone, Ed., Springer-Verlag, 1991, pp. 303–311.
76. R.L. Rivest, "The MD5 message-digest algorithm," *Request for Comments (RFC) 1321*, Internet Activities Board, Internet Privacy Task Force, April 1992.
77. R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications ACM*, Vol. 21, February 1978, pp. 120–126.
78. M.J.B. Robshaw, O. Billet, Eds., *"New Stream Cipher Designs — The eSTREAM Finalists", LNCS 4986*, Springer-Verlag, 2008.
79. P. Rogaway, T. Shrimpton, "Cryptographic hash function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance,

and collision resistance," *Fast Software Encryption'04, LNCS 3017*, B.K. Roy and W. Meier, Eds., Springer-Verlag, 2004, pp. 371–388.
80. P. Rogaway, J.P. Steinberger, "Constructing cryptographic hash functions from fixed-key blockciphers," *Advances in Cryptology, Proceedings Crypto'08, LNCS 5157*, D. Wagner, Ed., Springer-Verlag, 2008, pp. 433–450.
81. M.-J.O. Saarinen, "Security of VSH in the real world," *Progress in Cryptology – Indocrypt 2006, LNCS 4329*, R. Barua and T. Lange, Eds., Springer-Verlag, 2006, pp. 95–103.
82. S.K. Sanadhya, P. Sarkar, "New collision attacks against up to 24-step SHA-2," *Progress in Cryptology – Indocrypt 2008, LNCS 5365*, D. Roy Chowdhury, V. Rijmen, and A. Das, Eds., Springer-Verlag, 2008, pp. 91–103.
83. Y. Sasaki, K. Aoki, "Finding preimages in full MD5 faster than exhaustive search," *Advances in Cryptology, Proceedings Eurocrypt'08, LNCS 5479*, A. Joux, Ed., Springer-Verlag, 2009, pp. 134–152.
84. D. Simon, "Finding collisions on a one-way street: can secure hash functions be based on general assumptions?" *Advances in Cryptology, Proceedings Eurocrypt'98, LNCS 1403*, K. Nyberg, Ed., Springer-Verlag, 1998, pp. 334–345.
85. A. Sotirov, M. Stevens, J. Appelbaum, A.K. Lenstra, D. Molnar, D.A. Osvik, B. de Weger, "Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate," *Advances in Cryptology, Proceedings Crypto'09, LNCS 5677*, S. Halevi, Ed., Springer-Verlag, 2009, pp. 55–69.
86. M. Stam, "Beyond uniformity: better security/efficiency tradeoffs for compression functions," *Advances in Cryptology, Proceedings Crypto'08, LNCS 5157*, D. Wagner, Ed., Springer-Verlag, 2008, pp. 397–412.
87. M. Stam, "Blockcipher based hashing revisited," *Fast Software Encryption'09, LNCS 5665*, O. Dunkelman, Ed., Springer-Verlag, 2009, pp. 67–83.
88. J.P. Steinberger, "The collision intractability of MDC-2 in the ideal-cipher model," *Advances in Cryptology, Proceedings Eurocrypt'07, LNCS 4515*, M. Naor, Ed., Springer-Verlag, 2007, pp. 34–51.
89. J.-P. Tillich, G. Zémor, "Hashing with SL_2," *Advances in Cryptology, Proceedings Crypto'94, LNCS 839*, Y. Desmedt, Ed., Springer-Verlag, 1994, pp. 40–49.
90. J.-P. Tillich, G. Zémor, "Collisions for the LPS expander graph hash function," *Advances in Cryptology, Proceedings Eurocrypt'08, LNCS 4965*, N. Smart, Ed., Springer-Verlag, 2008, pp. 254–269.
91. Ph. Van Heurck, "Trasec: Belgian security system for electronic funds transfers," *Computers & Security*, Vol. 6, 1987, pp. 261–268.
92. P.C. van Oorschot, M.J. Wiener, "Parallel collision search with cryptanalytic applications," *Journal of Cryptology*, Vol. 12, No. 1, 1999, pp. 1–28.
93. X. Wang, Y.L. Yin, H. Yu, "Finding collisions in the full SHA-1," *Advances in Cryptology, Proceedings Crypto'05, LNCS 3621*, V. Shoup, Ed., Springer-Verlag, 2005, pp. 1–16.
94. X. Wang, H. Yu, "How to break MD5 and other hash functions," *Advances in Cryptology, Proceedings Eurocrypt'05, LNCS 3494*, R. Cramer, Ed., Springer-Verlag, 2005, pp. 19–35.
95. X. Wang, H. Yu, Y.L. Yin, "Efficient collision search attacks on SHA-0," *Advances in Cryptology, Proceedings Crypto'05, LNCS 3621*, V. Shoup, Ed., Springer-Verlag, 2005, pp. 17–36.
96. M.J. Wiener, "The full cost of cryptanalytic attacks," *Journal of Cryptology*, Vol. 17, No. 2, 2004, pp. 105–124.

97. R. Winternitz, "A secure one-way hash function built from DES," *Proceedings IEEE Symposium on Information Security and Privacy,* IEEE Press, 1984, pp. 88-90.
98. K. Yasuda, "How to fill up Merkle-Damgård hash functions," *Advances in Cryptology, Proceedings Asiacrypt'08, LNCS 5350*, J. Pieprzyk, Ed., Springer-Verlag, 2008, pp. 272–289.
99. G. Yuval, "How to swindle Rabin," *Cryptologia*, Vol. 3, 1979, pp. 187–189.