The NIST SHA-3 Competition: A Perspective on the Final Year

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. Cryptographic hash functions map input strings of arbitrary length to fixed length output strings. They are expected to satisfy several security properties that include preimage resistance, second preimage resistance, and collision resistance. The free availability of efficient software-oriented hash functions such as MD4, MD5 and SHA-1 has resulted in a very broad deployment of hash functions, way beyond their initial design purposes. In spite of the importance for applications, until 2005 the amount of theoretical research and cryptanalysis invested in this topic was rather limited. Moreover, cryptanalysts had been winning the battle from designers: about 4 of every 5 designs were broken. In 2004 Wang et al. made a breakthrough in the cryptanalysis of MD4, MD5 and SHA-1. Around the same time, serious shortcomings were identified in the theoretical foundations of existing designs. In response to this hash function crisis, in the last five years a substantial number of papers has been published with theoretical results and novel designs. Moreover, NIST announced in November 2007 the start of the SHA-3 competition, with as goal to select a new hash function family by 2012. We present a brief outline of the state of the art of hash functions in the last year of the competition and attempt to identify the lessons learned and some open research problems.

1 Background

Cryptographic hash functions first appeared in the cryptographic literature in the 1976 seminal paper of Diffie and Hellman on public-key cryptography [8]. Today, hash functions are used in a broad range of applications: to compute a short unique identifier of a string (e.g. for a digital signature), as one-way function to hide a string (e.g. for password protection), to commit to a string in a protocol, for key derivation and for entropy extraction. In addition, they have been deployed to instantiate random oracles and as building block for other cryptographic primitives.

Most of the first hash function designs were broken very quickly; in the late 1980s there was a clear understanding that there was a need for more secure and more efficient hash functions. The first theoretical result was the construction

of a collision-resistance hash function based on a collision-resistant compression function, proposed independently by Damgård [7] and Merkle [15] in 1989. Around the same time, the first cryptographic algorithms were proposed that were intended to be fast in software; the hash functions MD4 [19] and MD5 [20] fall in this category. Both functions were picked up quickly by application developers as they were ten times faster than DES; in addition they were not patent-encumbered and they posed less export problems than an encryption algorithm such as DES. As a consequence, hash functions were also used to construct MAC algorithms (e.g., HMAC as analyzed by Bellare et al. [4, 3]) and even block ciphers and stream ciphers.

During the 1990s, a growing number of hash functions were proposed [18], but unfortunately very few of these designs have withstood cryptanalysis. Notable results were obtained by Dobbertin, who found collisions for MD4 in 1995 [9] and collisions for the compression function of MD5 in 1996 [10]. Very few theoretical results were available in the area. At the same time however, MD5 and SHA-1, the latter introduced in 1995 by NIST (National Institute for Standards and Technology, US) [12], were deployed in an ever growing number of applications, resulting in the name “Swiss army knives” of cryptography.

In 2004, Wang et al. made substantial progress on the cryptanalysis of the MD4 family: by introducing a sophisticated variant of differential cryptanalysis they found collisions for MD4 by hand and for MD5 in a few minutes [22]. They managed to reduce the cost of collisions for SHA-1 by three orders of magnitude [21], which undermined the confidence in this widely used standard. As a consequence, the interest in hash functions surged: many new theoretical results were obtained, new designs were proposed and the cryptanalytic techniques of Wang et al. were further developed. Today RIPEMD-160 [11] seems to be one of the few older 160-bit hash functions for which no shortcut attacks are known. NIST introduced in 2002 the SHA-2 family of hash functions [13] with as goal to match the security levels provided by 3-DES and AES (output results of 224 to 512 bits). There is a concern that the attacks of Wang et al. would also apply to these functions, which have design principles that are quite similar to those of SHA-1; however, it should be pointed out that the more complex diffusion and the nonlinear message expansion of SHA-2 have held up against the current attack techniques.

2 The SHA-3 Competition

In November 2007, NIST announced that it would organize an open competition to select the SHA-3 algorithm [16]. In October 2008, 64 candidates were submitted; 51 of these were selected for the first round and in July 2009, 14 were admitted to the second round. In December 2010, NIST has announced 5 finalists: Blake, Grøstl, JH, Keccak, and Skein. The final winner will be selected by mid 2012.

Some preliminary conclusions can be drawn. A first observation is that during the SHA-3 competition many new designs have broken and many of the (strong)

designs been tweaked; this clearly shows that designing a secure hash function is very delicate. The largest design innovation is the introduction by Bertoni et al. [5] of sponge functions, that are very different from the Merkle-Damgård design; the idea of a sponge function is not to start from a strong compression function, but to use a larger intermediate state and to obtain security from the iteration. Keccak [6] is an instantiation of a sponge function, while JH is a generalization of a sponge function that has been studied under the name parazoa [2]. The most powerful attack that has emerged right before and during the competition has been the rebound attack [14]; it has been applied successfully to many designs (or to reduced-round versions). The five finalists represent a large diversity: JH and Keccak use small S-boxes or Boolean functions only, Grøstl uses an 8-bit S-box as found in AES, while Blake and Skein use ARX (addition/rotation/xor) operations on larger words of 32/64 bits. From a theoretical perspective, substantial progress has been made with security reductions, but major gaps still remain as explained in [1]. The performance of all finalists is very good; Keccak seems to offer the best hardware performance, while Blake and Skein (that use CPU arithmetic) have some advantage in software on high end CPUs.

3 Conclusions

Based on the five finalists, it seems safe to predict that SHA-3 will be a robust and efficient hash function. The design itself will be very different from SHA-2, and it will likely co-exist for an extended period with SHA-2. One can expect that NIST will standardize a tree mode for hash functions to obtain improved performance on multi-core processors (see [7, 17] and several SHA-3 submissions). 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 for which we have a convincing security reduction.

References

1. Andreeva, E., Mennink, B., Preneel, B.: Security reductions of the second round SHA-3 candidates. In: M. Burmester, S. Magliveras, and G. Tsudik (eds.) ISC 2010. LNCS vol. 6531, pp. 39–53, Springer, Heidelberg (2010)
2. Andreeva, E., Mennink, B., Preneel, B.: The parazoa family: generalizing the sponge hash functions. Cryptology ePrint Archive: Report 2011/028, 14 January 2011.
3. Bellare, M.: New proofs for NMAC and HMAC: security without collision-resistance. In: Dwork, C. (ed.) CRYPTO'06. LNCS, vol. 4117, pp. 602–619. Springer, Heidelberg (2006)
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication, In: Kobitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
5. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indistinguishability of the sponge construction. In: Smart, N. (ed) EUROCRYPT '08. LNCS, vol. 4965, pp. 181–197. Springer-Verlag, Berlin (2008)

6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The KECCAK sponge function family (2009), submission to the NIST SHA-3 competition
7. Damgård, I.B.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
8. Diffie, W., Hellman, M.E.: New directions in cryptography. *IEEE Trans. Information Theory* **22** (1976) 644–654
9. Dobbertin, H.: Cryptanalysis of MD4. *J. Cryptology* **11** (1998) 253–271 See also In: Gollmann, D. (ed.) FSE'96. LNCS, vol. 1039, pp. 53–69. Springer, Heidelberg (1996)
10. Dobbertin, H.: The status of MD5 after a recent attack. *CryptoBytes* **2** (1996) 1–6
11. Dobbertin, H., Bosselaers, A., Preneel, B.: RIPEMD-160: a strengthened version of RIPEMD. In: Gollmann, D. (ed.) FSE'96. LNCS, vol. 1039, pp. 71–82. Springer, Heidelberg (1996)
12. 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
13. 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)
14. Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schlffer, M.: Rebound distinguishers: Results on the full Whirlpool compression function. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 126143. Springer, Heidelberg (2009)
15. Merkle, R.: One way hash functions and DES. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 428–446. Springer, Heidelberg (1990)
16. NIST SHA-3 Competition, <http://csrc.nist.gov/groups/ST/hash/>.
17. Pal, P., Sarkar, P.: PARSHA-256 – A new parallelizable hash function and a multithreaded implementation. In: Johansson, T. (ed.) FSE'03. LNCS, vol. 2887, pp. 347–361. Springer, Heidelberg (2003)
18. Preneel, B.: *Analysis and design of cryptographic hash functions* Doctoral Dissertation, Katholieke Universiteit Leuven (1993)
19. Rivest, R.L.: The MD4 message digest algorithm. In: Vanstone, S. (ed.) CRYPTO'90. LNCS, vol. 537, pp. 303–311. Springer, Heidelberg (1991)
20. Rivest, R.L.: The MD5 message-digest algorithm. *Request for Comments (RFC) 1321*, Internet Activities Board, Internet Privacy Task Force, (April 1992)
21. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Shoup, V. (ed.) CRYPTO'05. LNCS, vol. 3621, pp. 1–16. Springer, Heidelberg (2005)
22. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Cramer, R. (ed.) EUROCRYPT'05. LNCS, vol. 3494, pp. 19–35. Springer, Heidelberg (2005)