Practical Hash Functions Constructions Resistant to Generic Second Preimage Attacks Beyond the Birthday Bound

Charles Bouillaguet*, Pierre-Alain Fouque

Ecole Normale Supérieure, 45 rue d'Ulm, 75005 Paris, France, +33 1 44 32 20 48

Abstract

Most cryptographic hash functions rely on a simpler primitive called a compression function, and in nearly all cases, there is a reduction between some of the security properties of the full hash function and those of the compression function. For instance, a celebrated result of Merkle and Damgård from 1989 states that a collision on the hash function cannot be found without finding a collision on the compression function at the same time.

This is however not the case for another basic requirement, namely second preimage resistance. In fact, on many popular hash functions it is possible to find a second preimage on the iteration without breaking the compression function. This paper studies the resistance of two practical modes of operations of hash functions against such attacks. We prove that the known generic second preimage attacks against the Merkle-Damgård construction are optimal, and that there is no generic second preimage attack faster than exhaustive search on Haifa, a recent proposal by Biham and Dunkelman.

Keywords: hash functions, modes of operation, second preimage attacks, provable security

1. Introduction

Hash functions are ubiquitous cryptographic primitives. They are used in authentication, encryption, signature schemes, and serve as a building block in many higher-level cryptographic functionality. Their are often assumed to implement a public random function, and in all cases, the very minimum requirement which can be expected from a cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^n$ is that it should be:

- One-way. Given $h \in \{0,1\}^n$, finding a preimage M such that $H(M) = h$ should not be significantly faster than hashing 2^n random messages with H.
- Second-Preimage Resistant. Given $M \in \{0,1\}^*$, finding a second preimage M' such that $H(M) = H(M')$ should not be significantly faster than hashing 2^n random messages with H.
- Collision-Resistant. Finding two distinct messages M and M' such that $H(M) = H(M')$ should not be significantly faster than evaluating H about $2^{n/2}$ times.

Most modern hash functions are the product of the combination of a *compression function*, hashing a small number of bits (typically 512) into a smaller number (typically 256), and of a *mode of operation*, describing how the compression function should be used to process arbitrarily big messages. The most popular and well-known mode of operation is the *Merkle-Damgård* construction, introduced in 1989 and named after its two independent inventors [1, 2]. One of its distinctive features is that it promotes the collision-resistance and preimage resistance of the compression function to the full hash function: for instance, a collision on the compression function can be deduced efficiently from a collision on the full hash function. Thus, in order to build a collision-resistant hash function, it is sufficient to design a collision-resistant compression function.

However, this is unfortunately not for case for second preimage resistance. There is no proof that a second preimage adversary against the hash function could be turned into an adversary against the compression function. Even worse, in 2005, Kelsey and Schneier, extending a recent result of Joux [3], gave a *generic* second preimage attack against Merkle-Damgård [4]. Here, the adjective "generic" means that the attack does not exploit any property of the compression function, and that it would work even if the compression function were a public random function. In some sense, it is an attack against the mode of operation itself. This attack finds a second preimage of any message of length about ℓ by evaluating the compression function only $2^n/\ell$ times. Its inventor also re-exposed an older result of Dean [5], who showed a similar attack in the case where fixed points could be efficiently found on the compression function (and it is the case for many popular compression functions). Later on, Andreeva, Bouillaguet, Fouque, Hoch, Kelsey and Shamir found a new generic

*Corresponding author

Email addresses: charles.bouillaguet@ens.fr (Charles Bouillaguet), pierre-alain.fouque@ens.fr (Pierre-Alain Fouque) second preimage attack [6], capable of dealing with variants of the Merkle-Damgård mode specifically designed to avoid the Kelsey-Schneier attack, such as Rivest's dithered hashing [7].

1.1. Towards Provable Resistance to Generic Attacks.

The existence of generic second preimage attacks, and of a few other generic attacks [3, 8], demonstrated that there was definitely a problem with the Merkle-Damgård construction, and motivated further research, and new modes of operations have emerged. It also motivated hash function designers to provide *proofs* that their mode of operation is sounds, and that it does not suffer from generic attacks.

1.1.1. New Modes

In 2005, Lucks introduced the Wide-Pipe Hash [9]. The idea is to keep the internal state of the hash function twice as big as the hash, so to finding a collision on the internal state by brute-force is infeasible. Using this fact, Lucks provided proofs that generic second preimage attacks could not be faster than brute force. The drawback is a slightly larger memory footprint. This strategy has had a great success, and it has been implemented by many hash function proposals that have advanced to the second round of the ongoing SHA-3 competition, namely Blue Midnight Wish, ECHO, Fugue, Gröestl, JH, SIMD and Skein.

In 2006, Biham and Dunkelman introduced HAIFA, the HAsh Iterative FrAmework in [10]. HAIFA is a collection of slight tweaks to the original Merkle-Damgård mode, and it does not enlarge the internal state. Its inventors claimed that generic second preimage attacks against HAIFA would require a workload of 2^n. This claim was not backed by a security proof, but no attack has been found so far. The decisive modification suggested by HAIFA to the original Merkle-Damgård construction in order to thwart generic second preimage attacks is the addition of a *round counter* input to the compression function. This idea also has had a great success, and it is implemented in four second round SHA-3 candidates: BLAKE, ECHO, SHAvite-3 and Skein. The second-round candidate Shabal also has a round counter, but it has a very different mode of operation.

In this paper, we will disregard the wide-pipe hash and focus on *narrow-pipe*[1] modes of operation, and more specifically on Merkle-Damgård and HAIFA. The exact resistance to generic second preimage attack of these two is in fact unknown. Existing attacks give an upper-bound above the birthday paradox, and the fact that a second preimage is also a collision give a birthday-lower bound. However, there is still a gap between those. So, the generic second preimage security of Merkle-Damgård is known to lie somewhere between $2^{n/2}$ and $2^n/\ell$ queries, for messages of size ℓ, which that of HAIFA lie somewhere between $2^{n/2}$ and 2^n queries.

1.1.2. New proof Techniques

Since the introduction of the random oracle (RO) methodology by Bellare and Rogaway [11], cryptographic hash functions have been widely used in protocols where they act as the concrete realization of the random oracles (which are public random functions). In other terms, the implicit assumption was that a hash function should be a public function with no special property, "as good" as if it were randomly chosen amongst all functions with the same domain and range. This was clearly not the case of the Merkle-Damgård mode of operation, even with an ideal compression function.

To face this challenge, the community widely used the *indifferentiability framework* introduced by Maurer *et al.* [12] to assess the security of new modes of operation. The proof that a mode of operation H is indifferentiable from a random oracle shows that in any cryptographic protocol, a random oracle can be replaced by the H-iteration of a public random function (with fixed input length). This guarantee amongst other things the impossibility of generic attacks on the mode of operation.

This methodology has had some successes as well, in allowing Coron, Dodis, Malinaud and Puniya to show that a minor tweak of Merkle-Damgård, called *Prefix-Free-Merkle-Damgård* was impossible to differentiate from a RO in less than $2^{n/2}$ queries [13]. It also enabled Chang and Nandi to show that any generic attack against the wide-pipe hash would require $2^n/n$ queries [14].

Unfortunately, this proof technique cannot be used for our purpose. First of all, it cannot say anything about Merkle-Damgård: this venerable mode of operation is *not* indifferentiable from a random oracle, because of the so-called *length extension attack.* More fundamentally, indifferentiability can only be proved up to the birthday bound. This follows from the fact that once a collisions on the compression function has been found, then it can often be used to build efficiently several pairs of colliding message by exploiting the iterated nature of the process, and thus allowing a distinguisher to tell apart the iteration from a random function.

Obtaining provable security for one of the main property of cryptographic hash functions therefore requires an *ad hoc* approach.

1.2. Related Work.

Keyed hash functions provably achieving (keyed) second preimage resistance beyond the birthday bound in the standard model (based on the hardness of a computational problem) have been proposed as early as 1989 by Naor and Yung, under the name of Universal One-Way Hash Functions (UOWHF) [15]. The same (keyed) security notion has also been called "Target Collision Resistance" by Bellare and Rogaway in 1997 [16], and "everywhere second

[1] We call "narrow-pipe" a construction where the internal state has the same length as the digest

preimage resistance" (eSec) by Rogaway and Shrimpton in 2004 [17]. Modes of operation promoting the eSec property of the compression function to the whole construction were proposed by Bellare and Rogaway in 1997 [16] and by Shoup in 2000 [18]. Shoup's construction is remarkable as it has an n-bit internal state and provably achieves $\mathcal{O}\left(2^{n-\ell}\right)$ second preimage resistance for messages of size at most ℓ. This bound has recently been shown to be tight thanks to the second preimage attack of [6]. However, these schemes hardly made it to the world of practical cryptography, as keyed hash functions are rarely used, even though they are sufficient for signature applications.

In 2008, two key-less modes of operation were presented, that also obtain beyond-birthday second preimage security: Yasuda's split-padding [19] and Andreeva's Three-property-secure hash function [20]. However these proposal did not really make it to the practical world.

1.3. Our Goal and our Results.

The object of this paper is to provide beyond the birthday bound provable resistance against second preimage attacks for Merkle-Damgård and Haifa.

Our contribution is to show that if the compression function is treated as a random oracle, then the second preimage resistance is $\Omega\left(2^{n-k}\right)$ for 2^k-blocks messages in Merkle-Damgård, and $\Omega\left(2^{n}\right)$ for Haifa. We therefore demonstrate that the existing generic second preimage attacks against Merkle-Damgård are optimal and that there is no generic second preimage attack at all against Haifa, therefore closing the gap between attacks and proofs.

The main idea common to all the proofs presented in this paper is almost directly adapted from the existing generic second preimage attacks: we lower-bound the complexity of one particular step common to all these attacks, namely when some kind of a possible prefix has to be "connected" to the target message.

Our security proofs are particularly simple and apply to practical construction that were not designed with provable security in mind. This is nevertheless the first time, to our knowledge, that a non-trivial security result above the birthday bound is given for practical narrow-pipe constructions.

1.4. Organization of the Paper.

The organization of this paper is as follows: in Section 2 we define a few notations we use through the rest of the paper, give formal definitions of the modes of operation we consider and discuss our security model. In Section 3 we offer proofs of security for the Merkle-Damgård and Haifa modes of iteration in the random oracle model. We conclude the paper in Section 4.

2. Preliminaries and Definitions

Throughout the paper $\left|M\right|$ denotes the length of M in blocks (and not in bits as usually done), *i.e.*, $\left|M\right| = \ell$ if M has ℓ blocks.

2.1. Iterated Constructions of Hash Functions

2.1.1. The Merkle-Damgård mode of iteration.

The Merkle-Damgård mode of iteration was independently suggested in 1989 by Merkle [1] and Damgård [2]. The hash function $H^F : \{0,1\}^* \to \{0,1\}^n$ is built by iterating a compression function $F\colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. The hash process works as follows:

- Pad and split a message M into r blocks $x_1, \ldots, x_r$ of m bits each.
- Set h_0 to the initialization value IV .
- For each message block x_i compute $h_i = f\left(h_{i-1}, x_i\right)$.
- Output $H^F(M) = h_r$.

The padding is done usually by appending a single '1' bit followed by as many '0' bit as needed to complete a m-bit block including the length of M in bits (the well-known Merkle-Damgård strengthening). The strengthening guarantees that a collision on F can be directly read off from a collision on H^F.

2.1.2. The Haifa mode of iteration.

The HAsh Iterative FrAmework (Haifa), introduced by Biham and Dunkelman [10], is a Merkle-Damgård construction where a bit counter and salt are added to the input of the compression function, amongst other features. We shall disregard the salt and all the other features throughout this paper (as they do not affect our results). We will therefore describe a study a simplified version of Haifa, but our result are directly applicable to the full version. We describe an instance of Haifa with a 64-bit counter (this matches the sizes used in currently deployed hash functions). The HAsh Iterative FrAmework H^F is built by iterating a compression function $F\colon \{0,1\}^n \times \{0,1\}^m \times \{0,1\}^{64} \to \{0,1\}^n$ as follows:

- Pad and split a message M into r blocks $x_1, \ldots, x_r$ of m bits each.
- Set h_0 to the initialization value IV.
- For each message block x_i compute $h_i = F\left(h_{i-1}, x_i, i\right)$.
- Output $H^F(M) = h_r$.

The padding is done by appending a single '1' bit followed by as many '0' bit as needed to complete an m-bit block after the message length and the digest size are appended.

2.2. Computational Model

The proofs presented in this paper assume that the compression function is an *ideal primitive*, to which *information-theoretic* (*i.e.*, computationally unbounded) adversaries have *oracle access*. Their only obstacle to achieving an attack is the randomness of the query response. The number of queries sent to the primitive can then be used as a meaningful complexity measure (because the adversary cannot obtain any kind of advantage by computation alone without querying the function). In any case, it gives a lower-bound on the time complexity of the adversary. This setting is very similar to the analysis of block cipher-based constructions in the ideal cipher model of Black, Rogaway and Shrimpton [21].

We often denote by q the number of queries sent to the compression function F by an adversary $\mathcal{A}$. For the sake of convenience, we enforce second preimage adversaries not to abort, to always return a message M, even if they do not win the security game, and to evaluate $H^F(M)$ before terminating, by issuing the corresponding queries to the compression function. We also enforce adversaries not to ask the same query twice.

2.3. Second Preimage Resistance.

Amongst the numerous notions of second preimage resistance, we will consider the one defined by the following game: a second preimage adversary $\mathcal{A}$ has oracle access to a compression function F. It receives a randomly generated challenge M of length ℓ, and succeeds if it outputs a second message M' such that $M \neq M'$ and $H^F(M) = H^F(M')$, where H is an iteration mode for F (such as Merkle-Damgård or Haifa). Such an adversary (q, ℓ, ε)-breaks H^F if after at most q queries to F its success probability is lower-bounded by ε. A hash function H^F is (q, ℓ, ε)-second preimage resistant (SPR) if the advantage for messages of length ℓ of any attacker asking at most q queries is upper-bounded by ε.

3. Security Proofs

We now come to the core of this paper, namely that it is possible to achieve beyond-the-birthday-bound second preimage resistance when the compression function is considered to be ideal.

3.1. Second Preimage Resistance of the Merkle-Damgård Construction

We now prove that the complexity of known generic second preimage attacks against Merkle-Damgård, which is of order $2^n/\ell$ for messages of length ℓ, are optimal.

Theorem 1. *Let F be a public random function, H^F be the Merkle-Damgård iteration of F, and $\mathcal{A}$ be a second preimage adversary against H^F which (q, ℓ, ε)-break H^F. Then:*

$$\varepsilon \leq q \cdot \ell / 2^n.$$

Proof. Consider an adversary $\mathcal{A}$ that (q, ℓ, ε)-breaks the second preimage resistance of H^F. We denote by h_i, for $1 \leq i \leq \ell$, the chaining values obtained while hashing M, according to the description in section 2.1.1. If $\mathcal{A}$ succeeds in finding a second preimage, then in particular $\mathcal{A}$ has found a collision. As argued in section 2.1.1, in the presence of the Merkle-Damgård *strengthening*, this implies a collision on the compression function F. In our case, there exists an index i_0 such that one of the colliding chaining value is h_{i_0}. This collision on F is therefore actually a second preimage of h_{i_0} for F. Note that because F is a random function, all the h_i's are random values.[2]

We now give an upper bound on the probability that $\mathcal{A}$ finds a second preimage of one out of ℓ random chaining values. We simulate the execution of $\mathcal{A}$, and bookmark the queries sent to the oracle for F. Every time $\mathcal{A}$ submits a new query to the oracle, it receives a uniformly-distributed random value. The probability that $\mathcal{A}$ wins thanks to this particular query is upper-bounded by the probability that this random value is one of the h_i's. This probability is exactly $\ell \cdot 2^{-n}$. Since $\mathcal{A}$ sends at most q queries, $\mathcal{A}$ wins with probability at most $q \cdot \ell \cdot 2^{-n}$. □

It must be noted that this proof is fairly general, because it reduces the problem of finding a second preimage for H^F to the problem of finding a second preimage of one out of many random chaining values for F. It actually covers nearly all the existing iterated hash functions; for example, it could be adapted to the Envelopped Merkle-Damgård mode of iteration of Bellare and Ristenpart [22], to Shoup's UOWHF [18], to Rivest's dithered hash [7], to Haifa [10], etc.

3.2. Second Preimage Resistance of Haifa

The inventors of Haifa claim that it has optimal resistance against generic second preimage attacks. The bound given by theorem 1 is however not strong enough to back up their claim. A slightly more involved proof technique is required to prove that Haifa achieves optimal second preimage resistance. The next theorem captures the intuitive idea that the known generic second preimage attacks do not work against Haifa.

Theorem 2. *Let F be a public random function and H^F be the Haifa-iteration of F, and $\mathcal{A}$ a second preimage adversary that (q, ℓ, ε)-break H^F. Then:*

$$\varepsilon \leq q / 2^{n-1}.$$

Proof. We simulate the execution of the adversary $\mathcal{A}$, and bookmark the queries sent by A to F: it is a set S of tuples (x, m, c, y), with $y = F(x, m, c)$. We suppose that $\mathcal{A}$ evaluates $H^F(M)$, so $\mathcal{A}$ sends the corresponding queries to the oracles at some point. Let us denote these particular queries $(h_{i-1}, m_i, c_i, h_i)_{1 \leq i \leq \ell}$. In particular, $H^F(M) = h_\ell$.

Suppose now that $\mathcal{A}$ wins. We first eliminate the special case when $\mathcal{A}$ finds a preimage of h_ℓ for F (this essentially means that $\mathcal{A}$ has found a preimage without using the fact that M is known).

[2] We note that this claim is not necessarily true when the message is long and there are collision between the various chaining values. However, as this has a non-negligible probability only when $\ell \geq O(2^{n/2})$, we allow ourselves to disregard such very long messages.

1. If $|M| \neq |M'|$, then the values of the counter entering the compression function in its last invocation are different. Therefore, $\mathcal{A}$ has found a second preimage on F. Each query has a probability 2^{-n} to give this preimage, because F is a random function.

2. Otherwise, $|M| = |M'|$. This means that $\mathcal{A}$ has found collision with M, similarly to what happens in the proof of theorem 1. We model this situation with the following event, that we call $\mathbf{E}$. Intuitively, $\mathbf{E}$ is realized as soon as $\mathcal{A}$ submits a query to F the answer of which gives a second preimage of one of the h_i. Formally, $\mathbf{E}$ is realized if and only if there is in S a query (x, m, i_0, h_{i_0}) for a given value of i_0 (recall that h_{i_0} is the i_0-th chaining value obtained in the process of hashing M), and such that $(x, m) \neq (h_{i_0}, m_{i_0})$.

Claim 1. If $\mathcal{A}$ wins and $|M| = |M'|$, then $\mathbf{E}$ is realized.

Justification. Thanks to the result of Merkle-Damgård, we know that there is a collision on the compression function where one of the colliding hash value is one of the h_i. However, this is not sufficient to say that $\mathbf{E}$ is realized, because we would need to know that the values of the counter are actually the same. We now prove that it is indeed the case.

Lemma 3 (CR Preservation on Haifa). *Let H^F be the* Haifa *iteration of an arbitrary compression function F. If $H^F(M) = H^F(M')$ with $M \neq M'$ and $|M| = |M'|$, then there is a collision on F, with the same value of the counter (this means that $\mathbf{E}$ is realized).*

Proof. let us note $M = x_1, \ldots, x_r$, $M' = x'_1, \ldots, x'_r$, $h_0 = h'_0 = IV$, $h_i = F(h_{i-1}, x_i, i)$ and $h'_i = F(h'_{i-1}, x'_i, i)$.

Since $h_r = h'_r$, either there is a collision on F (with counter value r), or $(x_r, h_{r-1}) = (x'_r, h'_{r-1})$. In the latter case, either there is a collision for F (with counter value $r-1$) or $(x_{r-1}, h_{r-2}) = (x'_{r-1}, h'_{r-2})$. This argument repeats. Since $|M| = |M'|$, then either there is a collision for F at some point (with the same counter value), or $x_i = x'_i$, for all i, $1 \leq i \leq r$. In the latter case, $M = M'$, which is impossible. This completes the proof of the lemma. □

To complete the proof of theorem 2, we now show an upper-bound on the probability that $\mathbf{E}$ is realized. When $\mathcal{A}$ submits its i-th query to the simulator (and note that the number i is part of the query), a random value is chosen by the simulator and returned to $\mathcal{A}$. The event $\mathbf{E}$ is realized if and only if this value is h_i, and this happens with probability 2^{-n}. This query may also allow $\mathcal{A}$ to invert h_ℓ with probability 2^{-n}. Each query allows $\mathcal{A}$ to win with probability $2^{-(n-1)}$, and there are q queries, which completes the proof. □

4. Conclusion: What do we Learn From These Proofs?

These two results rely crucially on the fact that the compression function is modeled as a public random function. For this reason, it could be argued that since no actual compression function will ever satisfy this hypothesis, then our results are vacuous.

Considering the underlying primitive to be ideal is a natural idea when reasoning about *modes of iteration* of hash functions. The 64 block-cipher based compression functions first analyzed by Preneel, Govaerts and Vandewalle [23] were later proved secure in the Ideal Cipher model by Black, Rogaway and Shrimpton [21], *i.e.* assuming that the underlying primitive is ideal. At the very least, the security results obtained in our model imply security against generic attacks, so they say something meaningful about the security of the mode of iteration itself.

For example, we know that the existing generic second preimage attacks [6, 8, 4] are almost optimal: in order to find a second preimage of a message of size ℓ on Haifa in less than 2^n operations, or on Merkle-Damgård in less than $2^n/\ell$ operations, an attacker will have to take a look at what is happening inside the compression function.

Acknowledgements

The authors are thank to their former colleague Sebastien Zimmer for helpful discussions. This work is partly supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

References

[1] R. C. Merkle, One Way Hash Functions and DES, in: Brassard [24], pp. 428–446.

[2] I. Damgård, A Design Principle for Hash Functions, in: Brassard [24], pp. 416–427.

[3] A. Joux, Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions, in: M. K. Franklin (Ed.), CRYPTO'04, Vol. 3152 of Lecture Notes in Computer Science, Springer, 2004, pp. 306–316.

[4] J. Kelsey, B. Schneier, Second Preimages on n-Bit Hash Functions for Much Less than 2^n Work, in: R. Cramer (Ed.), EUROCRYPT'05, Vol. 3494 of Lecture Notes in Computer Science, Springer, 2005, pp. 474–490.

[5] R. D. Dean, Formal Aspects of Mobile Code Security, Ph.D. thesis, Princeton University (January 1999).

[6] E. Andreeva, C. Bouillaguet, P.-A. Fouque, J. J. Hoch, J. Kelsey, A. Shamir, S. Zimmer, Second Preimage Attacks on Dithered Hash Functions, in: N. P. Smart (Ed.), EUROCRYPT, Vol. 4965 of Lecture Notes in Computer Science, Springer, 2008, pp. 270–288.

[7] R. L. Rivest, Abelian Square-Free Dithering for Iterated Hash Functions, Presented at ECrypt Hash Function Workshop, June 21, 2005, Cracow, and at the Cryptographic Hash workshop, November 1, 2005, Gaithersburg, Maryland.

[8] J. Kelsey, T. Kohno, Herding Hash Functions and the Nostradamus Attack, in: S. Vaudenay (Ed.), EUROCRYPT'06, Vol. 4004 of Lecture Notes in Computer Science, Springer, 2006, pp. 183–200.

[9] S. Lucks, A Failure-Friendly Design Principle for Hash Functions, in: B. K. Roy (Ed.), ASIACRYPT'05, Vol. 3788 of Lecture Notes in Computer Science, Springer, 2005, pp. 474–494.

[10] E. Biham, O. Dunkelman, A Framework for Iterative Hash Functions — HAIFA, Cryptology ePrint Archive, Report 2007/278, `http://eprint.iacr.org/2007/278` (August 24–25 2006).

[11] M. Bellare, P. Rogaway, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, in: ACM Conference on Computer and Communications Security, 1993, pp. 62–73.

[12] U. M. Maurer, R. Renner, C. Holenstein, Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology, in: M. Naor (Ed.), TCC, Vol. 2951 of Lecture Notes in Computer Science, Springer, 2004, pp. 21–39.

[13] J.-S. Coron, Y. Dodis, C. Malinaud, P. Puniya, Merkle-Damgård Revisited: How to Construct a Hash Function, in: CRYPTO'05, 2005, pp. 430–448.

[14] D. Chang, M. Nandi, Improved Indifferentiability Security Analysis of chopMD Hash Function, in: K. Nyberg (Ed.), FSE, Vol. 5086 of Lecture Notes in Computer Science, Springer, 2008, pp. 429–443.

[15] M. Naor, M. Yung, Universal One-Way Hash Functions and their Cryptographic Applications, in: STOC, ACM, 1989, pp. 33–43.

[16] M. Bellare, P. Rogaway, Collision-Resistant Hashing: Towards Making UOWHFs Practical, in: B. S. J. Kaliski (Ed.), CRYPTO, Vol. 1294 of Lecture Notes in Computer Science, Springer, 1997, pp. 470–484.

[17] P. Rogaway, T. Shrimpton, Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance, in: B. K. Roy, W. Meier (Eds.), FSE, Vol. 3017 of Lecture Notes in Computer Science, Springer, 2004, pp. 371–388.

[18] V. Shoup, A Composition Theorem for Universal One-Way Hash Functions, in: EUROCRYPT'00, 2000, pp. 445–452.

[19] K. Yasuda, How to fill up merkle-damgård hash functions, in: J. Pieprzyk (Ed.), ASIACRYPT, Vol. 5350 of Lecture Notes in Computer Science, Springer, 2008, pp. 272–289.

[20] E. Andreeva, B. Preneel, A three-property-secure hash function, in: R. M. Avanzi, L. Keliher, F. Sica (Eds.), Selected Areas in Cryptography, Vol. 5381 of Lecture Notes in Computer Science, Springer, 2008, pp. 228–244.

[21] J. Black, P. Rogaway, T. Shrimpton, Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV, in: M. Yung (Ed.), CRYPTO, Vol. 2442 of Lecture Notes in Computer Science, Springer, 2002, pp. 320–335.

[22] M. Bellare, T. Ristenpart, Multi-Property-Preserving Hash Domain Extension and the EMD Transform, in: X. Lai, K. Chen (Eds.), ASIACRYPT, Vol. 4284 of Lecture Notes in Computer Science, Springer, 2006, pp. 299–314.

[23] B. Preneel, R. Govaerts, J. Vandewalle, Hash functions based on block ciphers: A synthetic approach, in: D. R. Stinson (Ed.), CRYPTO, Vol. 773 of Lecture Notes in Computer Science, Springer, 1993, pp. 368–378.

[24] G. Brassard (Ed.), CRYPTO '89, Santa Barbara, California, USA, August0-24, 1989, Proceedings, Vol. 435 of Lecture Notes in Computer Science, Springer, 1990.