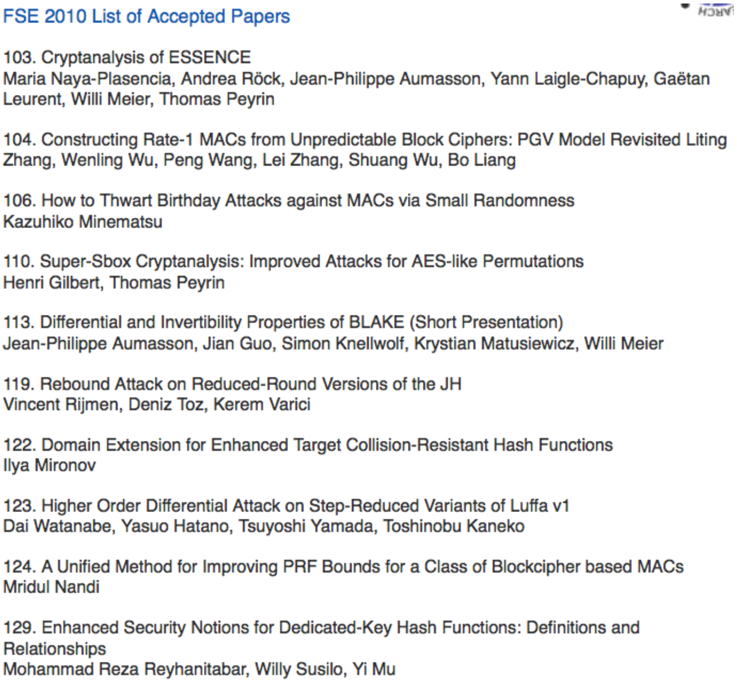
10 years of cryptographic hashing

Jean-Philippe Aumasson

HASH, x. There is no definition for this word—nobody knows what hash is.

Ambrose Bierce, *The Devil's Dictionary*



FSE 2010 List of Accepted Papers

103. Cryptanalysis of ESSENCE
Maria Naya-Plasencia, Andrea Röck, Jean-Philippe Aumasson, Yann Laigle-Chapuy, Gaëtan Leurent, Willi Meier, Thomas Peyrin

104. Constructing Rate-1 MACs from Unpredictable Block Ciphers: PGV Model Revisited Liting
Zhang, Wenling Wu, Peng Wang, Lei Zhang, Shuang Wu, Bo Liang

106. How to Thwart Birthday Attacks against MACs via Small Randomness
Kazuhiko Minematsu

110. Super-Sbox Cryptanalysis: Improved Attacks for AES-like Permutations
Henri Gilbert, Thomas Peyrin

113. Differential and Invertibility Properties of BLAKE (Short Presentation)
Jean-Philippe Aumasson, Jian Guo, Simon Knellwolf, Krystian Matusiewicz, Willi Meier

119. Rebound Attack on Reduced-Round Versions of the JH
Vincent Rijmen, Deniz Toz, Kerem Varici

122. Domain Extension for Enhanced Target Collision-Resistant Hash Functions
Ilya Mironov

123. Higher Order Differential Attack on Step-Reduced Variants of Luffa v1
Dai Watanabe, Yasuo Hatano, Tsuyoshi Yamada, Toshinobu Kaneko

124. A Unified Method for Improving PRF Bounds for a Class of Blockcipher based MACs
Mridul Nandi

129. Enhanced Security Notions for Dedicated-Key Hash Functions: Definitions and Relationships
Mohammad Reza Reyhanitabar, Willy Susilo, Yi Mu

133. Pseudo-cryptanalysis of the Original Blue Midnight Wish (Short Presentation)
Søren Steffen Thomsen

138. Rotational Cryptanalysis of ARX
Dmitry Khovratovich, Ivica Nikolic

139. Improving the Generalized Feistel
Tomoyasu Suzaki, Kazuhiko Minematsu

146. Security Analysis of the Mode of JH Hash Function
Rishiraj Bhattacharyya, Avradip Mandal, Mridul Nandi

151. Fast Software AES Encryption
Dag Arne Osvik, Joppe W. Bos, Deian Stefan, David Canright

152. Lightweight Privacy Preserving Authentication for RFID Based on a Stream Cipher
Olivier Billet, Jonathan Etrog, Henri Gilbert

153. Finding Preimages of Tiger Up to 23 Steps
Lei Wang, Yu Sasaki

155. Nonlinear Equivalence of Stream Ciphers
Sondre Rønjom, Carlos Cid

156. Attacking the Knudsen-Preneel Compression Function
Onur Özen, Thomas Shrimpton, Martijn Stam

163. Another Look at Complementation Properties
Charles Bouillaguet, Orr Dunkelman, Gaëtan Leurent, Pierre-Alain Fouque

166. Cryptanalysis of the DECT Standard Cipher
Karsten Nohl, Erik Tews, Ralf-Philipp Weinmann

Hashing at FSE

Year	Ratio hash/total	
2000	0.0 %	(0/21)
2001	7.4 %	(2/27)
2002	4.8 %	(1/21)
2003	3.7 %	(1/27)
2004	3.2 %	(1/31)
2005	13.8 %	(4/28)
2006	37.0 %	(10/27)
2007	21.4 %	(6/28)
2008	36.7 %	(11/31)
2009	61.9 %	(13/21)

This talk

INTRODUCTION

PART ONE: Hashes under attack

PART TWO: New paradigms

PART THREE: From SHA-2 to SHA-3

EPILOGUE

INTRODUCTION

Before 2000: the dark ages

Main results in the period $[-\infty; 2000]$:

- Iterative hashing [Rabin-78, Merkle-89, Damgård-89]
- Blockcipher-based modes [Preneel-Govaerts-Vandewalle-93]
- MD5 pseudo-collisions [den Boer-Bosselaers-93]
- SHA-0 collision attack [Chabaud-Joux-98]

Before 2000: the dark ages

Main results in the period $[-\infty; 2000]$:

- Iterative hashing [Rabin-78, Merkle-89, Damgård-89]
- Blockcipher-based modes [Preneel-Govaerts-Vandewalle-93]
- MD5 pseudo-collisions [den Boer-Bosselaers-93]
- SHA-0 collision attack [Chabaud-Joux-98]

Poor understanding of hashing

Hashes around: mostly MD5 and SHA-1

Attention focused on block ciphers (AES…)

Post-AES state-of-the-hash

Need for research on hashing

- MD5 and SHA-1 look fragile
- Lack of sound security definitions
- Better hashes needed to instantiate RO's
- No real understanding of operation modes (only BC-based)

Post-AES state-of-the-hash

Need for research on hashing

- MD5 and SHA-1 look fragile
- Lack of sound security definitions
- Better hashes needed to instantiate RO's
- No real understanding of operation modes (only BC-based)

Better armed community:

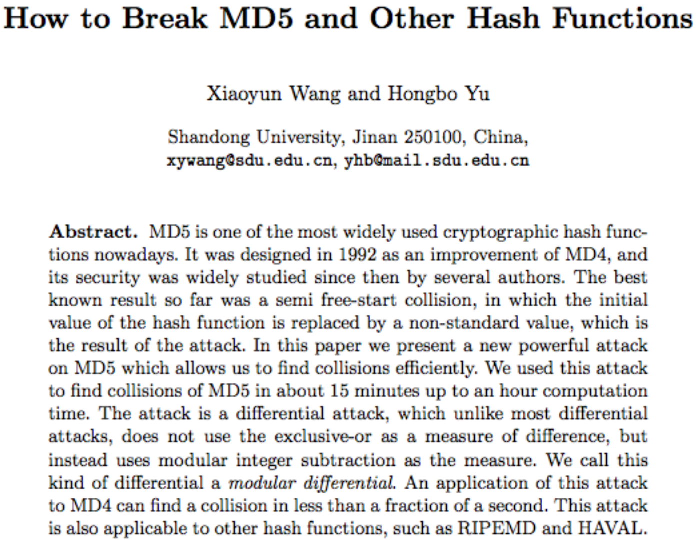
- Experience from the AES competition
- More research groups

It's not only SHA-3…

Part One

Hashes under attack

EUROCRYPT 2005



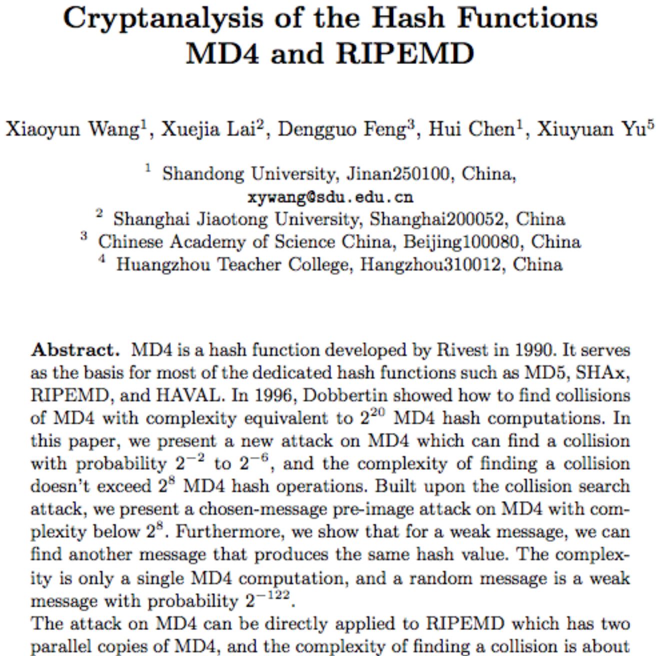
How to Break MD5 and Other Hash Functions

Xiaoyun Wang and Hongbo Yu

Shandong University, Jinan 250100, China,
xywang@sdu.edu.cn, yhb@mail.sdu.edu.cn

Abstract. MD5 is one of the most widely used cryptographic hash functions nowadays. It was designed in 1992 as an improvement of MD4, and its security was widely studied since then by several authors. The best known result so far was a semi free-start collision, in which the initial value of the hash function is replaced by a non-standard value, which is the result of the attack. In this paper we present a new powerful attack on MD5 which allows us to find collisions efficiently. We used this attack to find collisions of MD5 in about 15 minutes up to an hour computation time. The attack is a differential attack, which unlike most differential attacks, does not use the exclusive-or as a measure of difference, but instead uses modular integer subtraction as the measure. We call this kind of differential a *modular differential*. An application of this attack to MD4 can find a collision in less than a fraction of a second. This attack is also applicable to other hash functions, such as RIPEMD and HAVAL.

EUROCRYPT 2005



Cryptanalysis of the Hash Functions MD4 and RIPEMD

Xiaoyun Wang[1], Xuejia Lai[2], Dengguo Feng[3], Hui Chen[1], Xiuyuan Yu[5]

[1] Shandong University, Jinan250100, China,
xywang@sdu.edu.cn
[2] Shanghai Jiaotong University, Shanghai200052, China
[3] Chinese Academy of Science China, Beijing100080, China
[4] Huangzhou Teacher College, Hangzhou310012, China

Abstract. MD4 is a hash function developed by Rivest in 1990. It serves as the basis for most of the dedicated hash functions such as MD5, SHAx, RIPEMD, and HAVAL. In 1996, Dobbertin showed how to find collisions of MD4 with complexity equivalent to 2^{20} MD4 hash computations. In this paper, we present a new attack on MD4 which can find a collision with probability 2^{-2} to 2^{-6}, and the complexity of finding a collision doesn't exceed 2^{8} MD4 hash operations. Built upon the collision search attack, we present a chosen-message pre-image attack on MD4 with complexity below 2^{8}. Furthermore, we show that for a weak message, we can find another message that produces the same hash value. The complexity is only a single MD4 computation, and a random message is a weak message with probability 2^{-122}.

The attack on MD4 can be directly applied to RIPEMD which has two parallel copies of MD4, and the complexity of finding a collision is about

Collisions for MD5 et al.

Main results:

- 2^{37} collision attack for MD5
- 2^{8} (2^{18}) collision attack for MD4 (RIPEMD)

Differential path found “by hand”

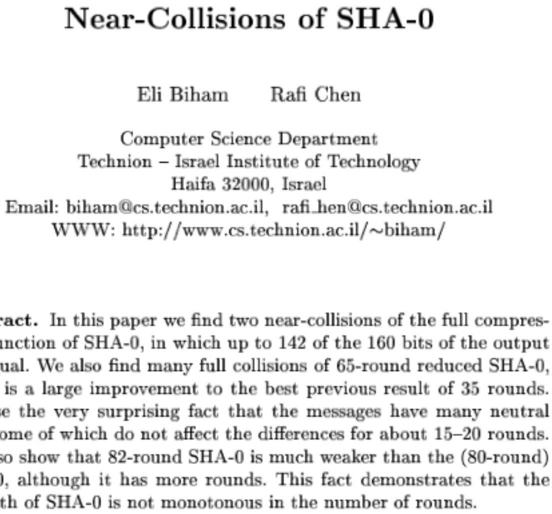
Differences with respect to modular addition

Advanced message modification to fulfill conditions

Then most advanced application of differential cryptanalysis

Many subsequent improvements. . . (by Klima, Stevens, et al.)

CRYPTO 2004



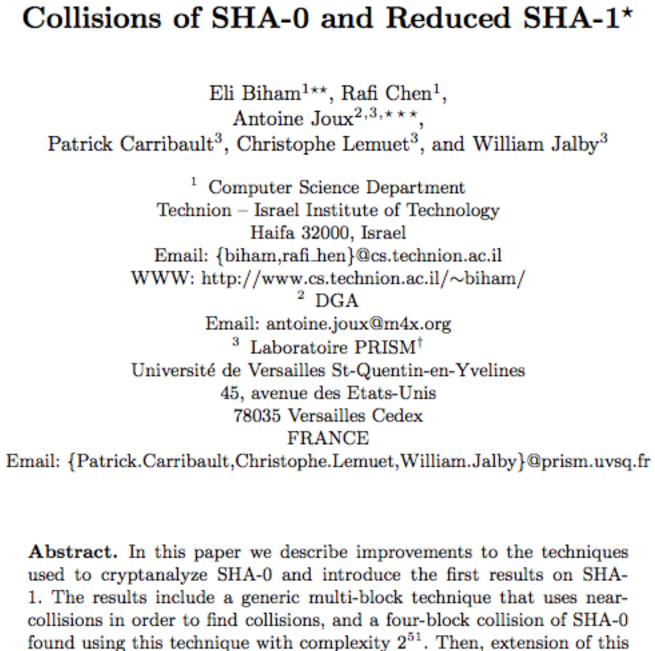
Near-Collisions of SHA-0

Eli Biham Rafi Chen

Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel
Email: biham@cs.technion.ac.il, rafi_hen@cs.technion.ac.il
WWW: http://www.cs.technion.ac.il/~biham/

Abstract. In this paper we find two near-collisions of the full compression function of SHA-0, in which up to 142 of the 160 bits of the output are equal. We also find many full collisions of 65-round reduced SHA-0, which is a large improvement to the best previous result of 35 rounds. We use the very surprising fact that the messages have many neutral bits, some of which do not affect the differences for about 15–20 rounds. We also show that 82-round SHA-0 is much weaker than the (80-round) SHA-0, although it has more rounds. This fact demonstrates that the strength of SHA-0 is not monotonous in the number of rounds.

EUROCRYPT 2005



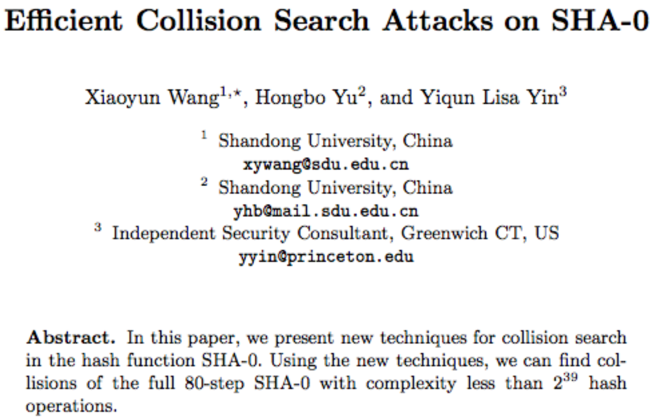
Collisions of SHA-0 and Reduced SHA-1[*]

Eli Biham[1,**], Rafi Chen[1],
Antoine Joux[2,3,***],
Patrick Carribault[3], Christophe Lemuet[3], and William Jalby[3]

[1] Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel
Email: {biham,rafi_hen}@cs.technion.ac.il
WWW: http://www.cs.technion.ac.il/~biham/
[2] DGA
Email: antoine.joux@m4x.org
[3] Laboratoire PRISM[†]
Université de Versailles St-Quentin-en-Yvelines
45, avenue des Etats-Unis
78035 Versailles Cedex
FRANCE
Email: {Patrick.Carribault,Christophe.Lemuet,William.Jalby}@prism.uvsq.fr

Abstract. In this paper we describe improvements to the techniques used to cryptanalyze SHA-0 and introduce the first results on SHA-1. The results include a generic multi-block technique that uses near-collisions in order to find collisions, and a four-block collision of SHA-0 found using this technique with complexity 2^{51}. Then, extension of this

CRYPTO 2005



Efficient Collision Search Attacks on SHA-0

Xiaoyun Wang[1,⋆], Hongbo Yu[2], and Yiqun Lisa Yin[3]

[1] Shandong University, China
xywang@sdu.edu.cn
[2] Shandong University, China
yhb@mail.sdu.edu.cn
[3] Independent Security Consultant, Greenwich CT, US
yyin@princeton.edu

Abstract. In this paper, we present new techniques for collision search in the hash function SHA-0. Using the new techniques, we can find collisions of the full 80-step SHA-0 with complexity less than 2^{39} hash operations.

Collisions for SHA-0

Main results:

- 2^{51} collision attack for SHA-0 [Biham et al.-05]
- 2^{39} collision attack for SHA-0 [Wang-Yu-Yin-05]

Based on earlier results [Chabaud-Joux-98] [Wang-Yin-97]

Introduction of the notion of neutral bits [Biham-Chen-04]

XOR differences [Biham et al.-05] vs. modular differences [Wang-Yu-Yin-05]

2^{51} attack implemented on a 256-CPU supercomputer

Exploit of the simple (linear, bitsliced) message expansion

$$m_i = m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}$$

Exploit sequences of local collisions

Joux's SHA-0 collision

Thursday 12th, August 2004

We are glad to announce that we found a collision for SHA-0.

First message (2048 bits represented in hex):

```
a766a602 b65cffe7 73bcf258 26b322b3 d01b1a97 2684ef53 3e3b4b7f 53fe3762
24c08e47 e959b2bc 3b519880 b9286568 247d110f 70f5c5e2 b4590ca3 f55f52fe
effd4c8f e68de835 329e603c c51e7f02 545410d1 671d108d f5a4000d cf20a439
4949d72c d14fbb03 45cf3a29 5dcda89f 998f8755 2c9a58b1 bdc38483 5e477185
f96e68be bb0025d2 d2b69edf 21724198 f688b41d eb9b4913 fbe696b5 457ab399
21e1d759 1f89de84 57e8613c 6c9e3b24 2879d4d8 783b2d9c a9935ea5 26a729c0
6edfc501 37e69330 be976012 cc5dfe1c 14c4c68b d1db3ecb 24438a59 a09b5db4
35563e0d 8bdf572f 77b53065 cef31f32 dc9dbaa0 4146261e 9994bd5c d0758e3d
```

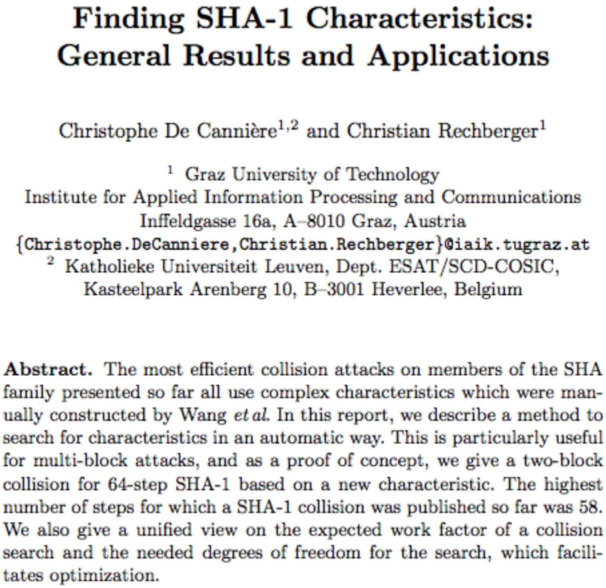
Second message:

```
a766a602 b65cffe7 73bcf258 26b322b1 d01b1ad7 2684ef51 be3b4b7f d3fe3762
a4c08e45 e959b2fc 3b519880 39286528 a47d110d 70f5c5e0 34590ce3 755f52fc
6ffd4c8d 668de875 329e603e 451e7f02 d45410d1 e71d108d f5a4000d cf20a439
4949d72c d14fbb01 45cf3a69 5dcda89d 198f8755 ac9a58b1 3dc38481 5e4771c5
796e68fe bb0025d0 52b69edd a17241d8 7688b41f 6b9b4911 7be696f5 c57ab399
a1e1d719 9f89de86 57e8613c ec9e3b26 a879d498 783b2d9e 29935ea7 a6a72980
6edfc503 37e69330 3e976010 4c5dfe5c 14c4c689 51db3ecb a4438a59 209b5db4
35563e0d 8bdf572f 77b53065 cef31f30 dc9dbae0 4146261c 1994bd5c 50758e3d
```

Common hash value (can be found using for example "openssl sha file.bin"
after creating a binary file containing any of the messages)

```
c9f160777d4086fe8095fba58b7e20c228a4006b
```

ASIACRYPT 2006



Finding SHA-1 Characteristics: General Results and Applications

Christophe De Cannière[1,2] and Christian Rechberger[1]

[1] Graz University of Technology
Institute for Applied Information Processing and Communications
Inffeldgasse 16a, A–8010 Graz, Austria
{Christophe.DeCanniere,Christian.Rechberger}@iaik.tugraz.at
[2] Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC,
Kasteelpark Arenberg 10, B–3001 Heverlee, Belgium

Abstract. The most efficient collision attacks on members of the SHA family presented so far all use complex characteristics which were manually constructed by Wang *et al.* In this report, we describe a method to search for characteristics in an automatic way. This is particularly useful for multi-block attacks, and as a proof of concept, we give a two-block collision for 64-step SHA-1 based on a new characteristic. The highest number of steps for which a SHA-1 collision was published so far was 58. We also give a unified view on the expected work factor of a collision search and the needed degrees of freedom for the search, which facilitates optimization.

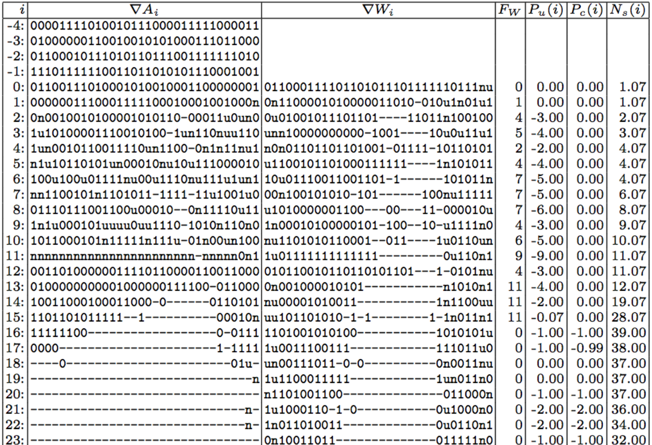
SHA-1 cryptanalysis

Main result:

- Automated method for finding (NL) characteristics

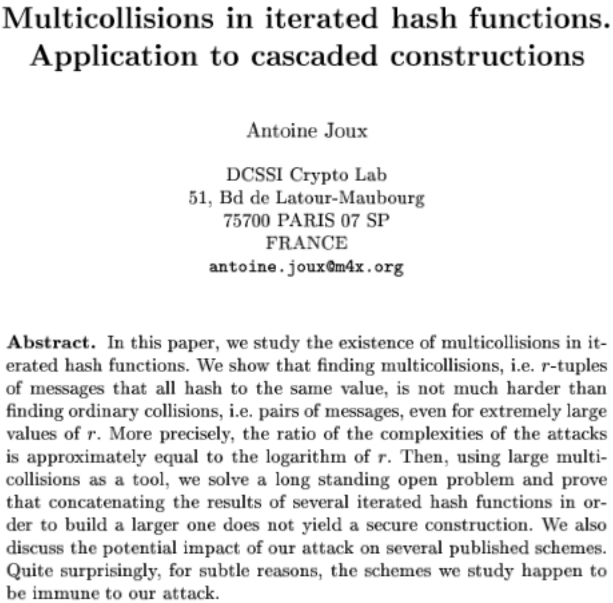
Follow-up to Wang et al.'s manually found characteristics

Example given for collisions of 64-step SHA-1:



i	∇A_i	∇W_i	F_W	$P_u(i)$	$P_c(i)$	$N_s(i)$
-4:	00001111010010111000011111000011					
-3:	01000000110010010101000111011000					
-2:	01100010111010110111001111111010					
-1:	11101111110011011010101110001001					
0:	01100111010001010010001100000001	01100011110110101110111111 0111nu	0	0.00	0.00	1.07
1:	0000001110001111100010001001000n	0n110000101000001101 0-010u1n01u1	1	0.00	0.00	1.07
2:	0n00100101000010101100-00011u0un0	0u01001011101101----11011n100100	4	-3.00	0.00	2.07
3:	1u10100001110010100-1un110nuu110	unn10000000000-1001----10u0u11u1	5	-4.00	0.00	3.07
4:	1un0010110011110un1100-0n1n11nu1	n0n01101101101001-01111-10110101	2	-2.00	0.00	4.07
5:	n1u10110101un00010nu10u111000010	u1100101101000111111----1n101011	4	-4.00	0.00	4.07
6:	100u100u01111nu00u1110nu111u1un1	10u01110011001101-1------101011n	7	-5.00	0.00	4.07
7:	nn1100101n1101011-1111-11u1001u0	00n100101010-101------100nu11111	7	-5.00	0.00	6.07
8:	01110111001100u00010--0n11110u11	u1010000001100---00---11-000010u	7	-6.00	0.00	8.07
9:	1n1u000101uuuu0uu1110-1010n110n0	1n00010100000101-100--10-u1111n0	4	-3.00	0.00	9.07
10:	1011000101n11111n111u-01n00un100	nu1101010110001--011----1u0110un	6	-5.00	0.00	10.07
11:	nnnnnnnnnnnnnnnnnnnnnnn-nnnnn0n1	1u01111111111111---------0u110n1	9	-9.00	0.00	11.07
12:	00110100000011110110000110011000	010110010110110101101---1-0101nu	4	-3.00	0.00	11.07
13:	0100000000001000000111100-011000	0n001000010101-----------n1010n1	11	-4.00	0.00	12.07
14:	10011000100011000-0------0110101	nu00001010011-----------1n1100uu	11	-2.00	0.00	19.07
15:	1101101011111--1----------00010n	uu101101010-1-1--------1-1n011n1	11	-0.07	0.00	28.07
16:	11111100------------------0-0111	1101001010100-----------1010101u	0	-1.00	-1.00	39.00
17:	0000----------------------1-1111	1u0011100111------------111011u0	0	-1.00	-0.99	38.00
18:	----0-----------------------01u-	un00111011-0-0----------0n0011nu	0	0.00	0.00	37.00
19:	-------------------------------n	1u1100011111------------1un011n0	0	0.00	0.00	37.00
20:	--------------------------------	n1101001100--------------011000n	0	-1.00	-1.00	37.00
21:	------------------------------n-	1u1000110-1-0-----------0u1000n0	0	-2.00	-2.00	36.00
22:	------------------------------n-	1n011010011-------------0u0110n1	0	-2.00	-2.00	34.00
23:	--------------------------------	0n10011011--------------011111n0	0	-1.00	-1.00	32.00

CRYPTO 2004



Multicollisions in iterated hash functions. Application to cascaded constructions

Antoine Joux

DCSSI Crypto Lab
51, Bd de Latour-Maubourg
75700 PARIS 07 SP
FRANCE
antoine.joux@m4x.org

Abstract. In this paper, we study the existence of multicollisions in iterated hash functions. We show that finding multicollisions, i.e. r-tuples of messages that all hash to the same value, is not much harder than finding ordinary collisions, i.e. pairs of messages, even for extremely large values of r. More precisely, the ratio of the complexities of the attacks is approximately equal to the logarithm of r. Then, using large multicollisions as a tool, we solve a long standing open problem and prove that concatenating the results of several iterated hash functions in order to build a larger one does not yield a secure construction. We also discuss the potential impact of our attack on several published schemes. Quite surprisingly, for subtle reasons, the schemes we study happen to be immune to our attack.

Multicollisions

Main result:

- Algorithm to find k-collisions for DM hashes in $\log_2 k \cdot 2^{n/2}$

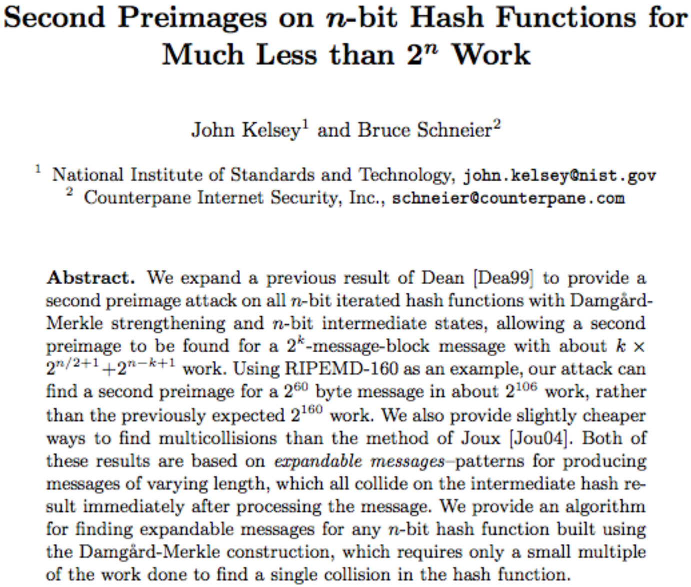
Improves on the folklore $k!^{1/k} \cdot 2^{n(k-1)/k}$ method

Application to concatenated hashes ($2^{n/4}$ collision attack)

Used in the “Nostradamus attack”

Used as a cryptanalysis tool (e.g., to break AURORA)

EUROCRYPT 2005



Second Preimages on n-bit Hash Functions for Much Less than 2^n Work

John Kelsey[1] and Bruce Schneier[2]

[1] National Institute of Standards and Technology, john.kelsey@nist.gov
[2] Counterpane Internet Security, Inc., schneier@counterpane.com

Abstract. We expand a previous result of Dean [Dea99] to provide a second preimage attack on all n-bit iterated hash functions with Damgård-Merkle strengthening and n-bit intermediate states, allowing a second preimage to be found for a 2^k-message-block message with about $k \times 2^{n/2+1} + 2^{n-k+1}$ work. Using RIPEMD-160 as an example, our attack can find a second preimage for a 2^{60} byte message in about 2^{106} work, rather than the previously expected 2^{160} work. We also provide slightly cheaper ways to find multicollisions than the method of Joux [Jou04]. Both of these results are based on *expandable messages*–patterns for producing messages of varying length, which all collide on the intermediate hash result immediately after processing the message. We provide an algorithm for finding expandable messages for any n-bit hash function built using the Damgård-Merkle construction, which requires only a small multiple of the work done to find a single collision in the hash function.

Long-message second preimage attack

Main result:

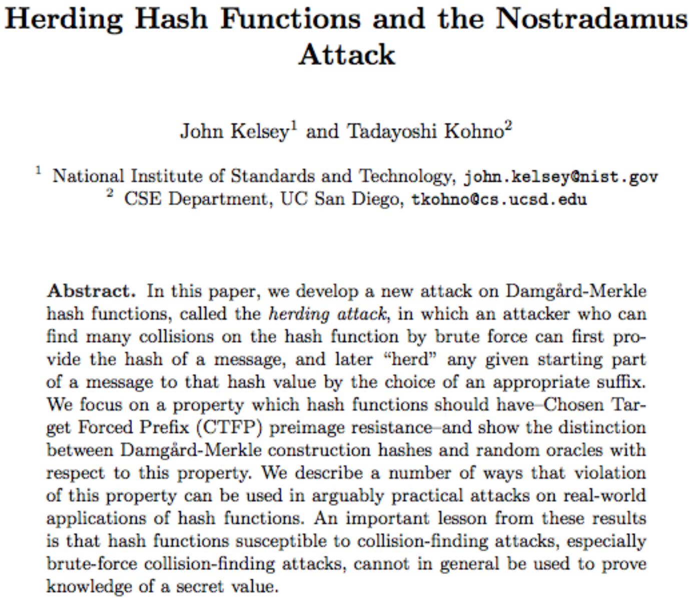
- Second preimage attack for DM hashes and 2^k-block messages in 2^{n-k}

Based on previous attack using easily found fixed points [Dean-99]

Introduction of the notion of “expandable message” $=$ multicollision with messages of different lengths

Also describe a multicollision attack in time $3 \cdot 2^{n/2}$ (long colliding messages)

EUROCRYPT 2006



Herding Hash Functions and the Nostradamus Attack

John Kelsey[1] and Tadayoshi Kohno[2]

[1] National Institute of Standards and Technology, john.kelsey@nist.gov
[2] CSE Department, UC San Diego, tkohno@cs.ucsd.edu

Abstract. In this paper, we develop a new attack on Damgård-Merkle hash functions, called the *herding attack*, in which an attacker who can find many collisions on the hash function by brute force can first provide the hash of a message, and later "herd" any given starting part of a message to that hash value by the choice of an appropriate suffix. We focus on a property which hash functions should have–Chosen Target Forced Prefix (CTFP) preimage resistance–and show the distinction between Damgård-Merkle construction hashes and random oracles with respect to this property. We describe a number of ways that violation of this property can be used in arguably practical attacks on real-world applications of hash functions. An important lesson from these results is that hash functions susceptible to collision-finding attacks, especially brute-force collision-finding attacks, cannot in general be used to prove knowledge of a secret value.

Herding hash functions and the Nostradamus attack

Main result:

- Herding attack and applications

Alice precomputes digest h, Bob chooses m_1, Alice finds m_2 such that $H(m_1 \| m_2) = h$

Commit to digest before I know full string I'm hashing! Can "predict" future events…

Suffix m_2 can be made meaningful, using multicollision techniques

Herding hash functions and the Nostradamus attack

Main result:

- Herding attack and applications

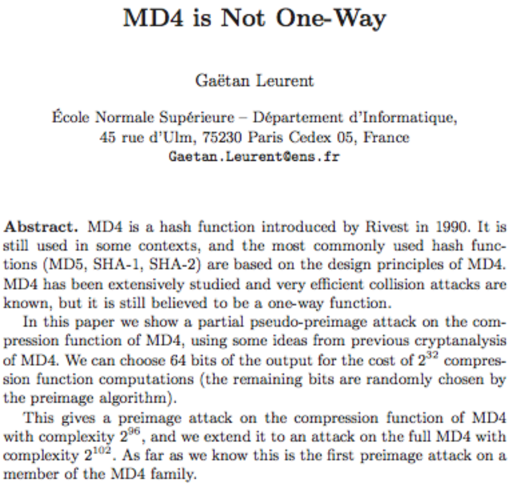
Alice precomputes digest h, Bob chooses m_1, Alice finds m_2 such that $H(m_1 \| m_2) = h$

Commit to digest before I know full string I'm hashing! Can "predict" future events...

Suffix m_2 can be made meaningful, using multicollision techniques

Last minute: from this morning's ePrint update:
(`eprint.iacr.org/2010/030`, by Stinson and Upadhyay)
"In this paper, we analyze the complexity of the construction of the 2^k-diamond structure proposed by Kelsey and Kohno. We point out a flaw in their analysis and show that their construction may not produce the desired diamond structure."

FSE 2008



MD4 is Not One-Way

Gaëtan Leurent

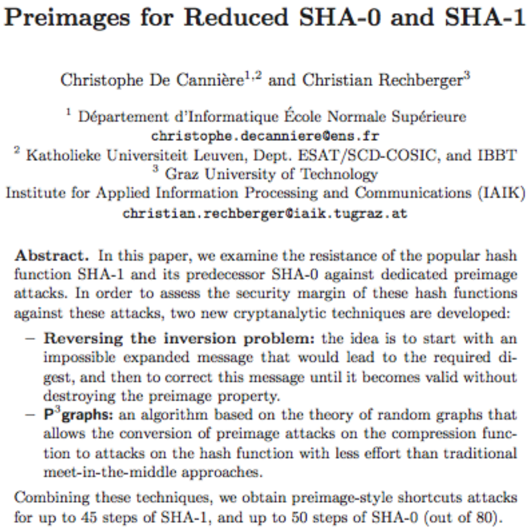
École Normale Supérieure – Département d'Informatique,
45 rue d'Ulm, 75230 Paris Cedex 05, France
Gaetan.Leurent@ens.fr

Abstract. MD4 is a hash function introduced by Rivest in 1990. It is still used in some contexts, and the most commonly used hash functions (MD5, SHA-1, SHA-2) are based on the design principles of MD4. MD4 has been extensively studied and very efficient collision attacks are known, but it is still believed to be a one-way function.

In this paper we show a partial pseudo-preimage attack on the compression function of MD4, using some ideas from previous cryptanalysis of MD4. We can choose 64 bits of the output for the cost of 2^{32} compression function computations (the remaining bits are randomly chosen by the preimage algorithm).

This gives a preimage attack on the compression function of MD4 with complexity 2^{96}, and we extend it to an attack on the full MD4 with complexity 2^{102}. As far as we know this is the first preimage attack on a member of the MD4 family.

CRYPTO 2008



Preimages for Reduced SHA-0 and SHA-1

Christophe De Cannière[1,2] and Christian Rechberger[3]

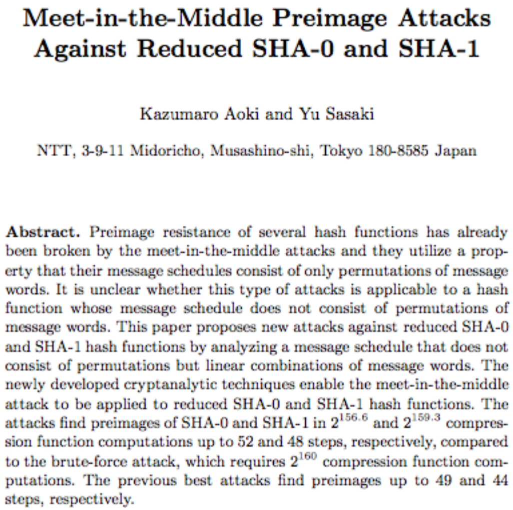
[1] Département d'Informatique École Normale Supérieure
christophe.decanniere@ens.fr
[2] Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC, and IBBT
[3] Graz University of Technology
Institute for Applied Information Processing and Communications (IAIK)
christian.rechberger@iaik.tugraz.at

Abstract. In this paper, we examine the resistance of the popular hash function SHA-1 and its predecessor SHA-0 against dedicated preimage attacks. In order to assess the security margin of these hash functions against these attacks, two new cryptanalytic techniques are developed:

- **Reversing the inversion problem:** the idea is to start with an impossible expanded message that would lead to the required digest, and then to correct this message until it becomes valid without destroying the preimage property.
- **P^{3}graphs:** an algorithm based on the theory of random graphs that allows the conversion of preimage attacks on the compression function to attacks on the hash function with less effort than traditional meet-in-the-middle approaches.

Combining these techniques, we obtain preimage-style shortcuts attacks for up to 45 steps of SHA-1, and up to 50 steps of SHA-0 (out of 80).

CRYPTO 2009



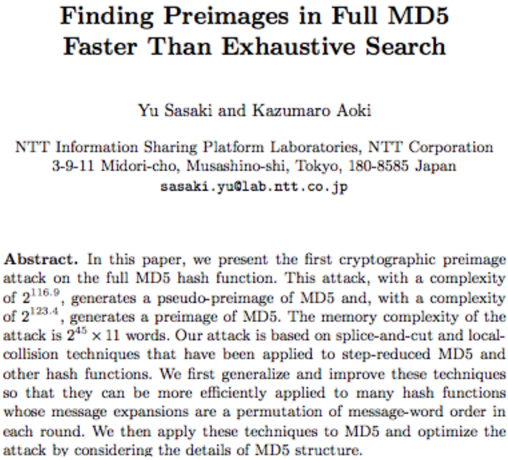
Meet-in-the-Middle Preimage Attacks Against Reduced SHA-0 and SHA-1

Kazumaro Aoki and Yu Sasaki

NTT, 3-9-11 Midoricho, Musashino-shi, Tokyo 180-8585 Japan

Abstract. Preimage resistance of several hash functions has already been broken by the meet-in-the-middle attacks and they utilize a property that their message schedules consist of only permutations of message words. It is unclear whether this type of attacks is applicable to a hash function whose message schedule does not consist of permutations of message words. This paper proposes new attacks against reduced SHA-0 and SHA-1 hash functions by analyzing a message schedule that does not consist of permutations but linear combinations of message words. The newly developed cryptanalytic techniques enable the meet-in-the-middle attack to be applied to reduced SHA-0 and SHA-1 hash functions. The attacks find preimages of SHA-0 and SHA-1 in $2^{156.6}$ and $2^{159.3}$ compression function computations up to 52 and 48 steps, respectively, compared to the brute-force attack, which requires 2^{160} compression function computations. The previous best attacks find preimages up to 49 and 44 steps, respectively.

EUROCRYPT 2009



Finding Preimages in Full MD5
Faster Than Exhaustive Search

Yu Sasaki and Kazumaro Aoki

NTT Information Sharing Platform Laboratories, NTT Corporation
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
`sasaki.yu@lab.ntt.co.jp`

Abstract. In this paper, we present the first cryptographic preimage attack on the full MD5 hash function. This attack, with a complexity of $2^{116.9}$, generates a pseudo-preimage of MD5 and, with a complexity of $2^{123.4}$, generates a preimage of MD5. The memory complexity of the attack is $2^{45} \times 11$ words. Our attack is based on splice-and-cut and local-collision techniques that have been applied to step-reduced MD5 and other hash functions. We first generalize and improve these techniques so that they can be more efficiently applied to many hash functions whose message expansions are a permutation of message-word order in each round. We then apply these techniques to MD5 and optimize the attack by considering the details of MD5 structure.

Preimages for MD4 et al.

Main results:

- Preimage attack for MD5 [Sasaki-Aoki-09]
- Preimage attack for reduced SHA-0/1 (50/45 steps) [De Cannière-Rechberger-08] [Aoki-Sasaki-09]

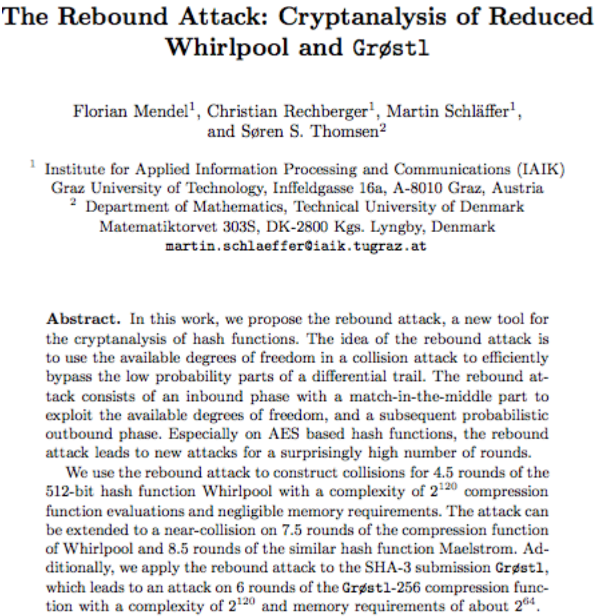
Series of papers introducing new techniques for finding preimages on MD4-like schemes: “neutral words”, “partial matching”, etc.

Often non-negligible memory requirements

Other preimage attacks in 2007/8 on (reduced): HAS-V, Tiger, GOST, Snefru, HAVAL, SHA-2, etc.

Recent results on reduced SHA-256 [Aoki et al-09]

FSE 2009



The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl

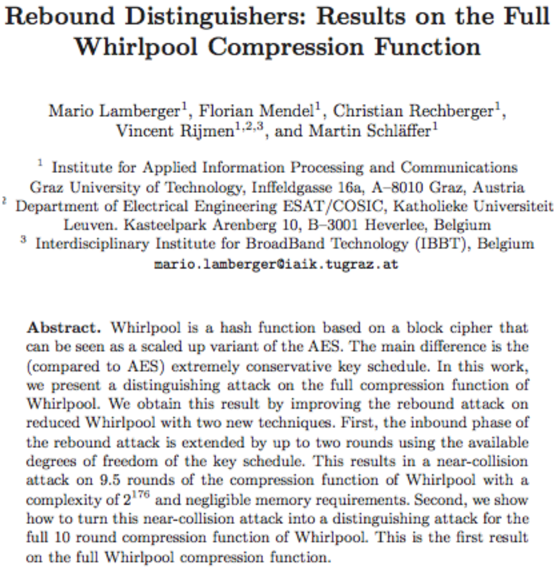
Florian Mendel[1], Christian Rechberger[1], Martin Schläffer[1],
and Søren S. Thomsen[2]

[1] Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria
[2] Department of Mathematics, Technical University of Denmark
Matematiktorvet 303S, DK-2800 Kgs. Lyngby, Denmark
martin.schlaeffer@iaik.tugraz.at

Abstract. In this work, we propose the rebound attack, a new tool for the cryptanalysis of hash functions. The idea of the rebound attack is to use the available degrees of freedom in a collision attack to efficiently bypass the low probability parts of a differential trail. The rebound attack consists of an inbound phase with a match-in-the-middle part to exploit the available degrees of freedom, and a subsequent probabilistic outbound phase. Especially on AES based hash functions, the rebound attack leads to new attacks for a surprisingly high number of rounds.

We use the rebound attack to construct collisions for 4.5 rounds of the 512-bit hash function Whirlpool with a complexity of 2^{120} compression function evaluations and negligible memory requirements. The attack can be extended to a near-collision on 7.5 rounds of the compression function of Whirlpool and 8.5 rounds of the similar hash function Maelstrom. Additionally, we apply the rebound attack to the SHA-3 submission `Grøstl`, which leads to an attack on 6 rounds of the `Grøstl`-256 compression function with a complexity of 2^{120} and memory requirements of about 2^{64}.

ASIACRYPT 2009



Rebound Distinguishers: Results on the Full Whirlpool Compression Function

Mario Lamberger[1], Florian Mendel[1], Christian Rechberger[1],
Vincent Rijmen[1,2,3], and Martin Schläffer[1]

[1] Institute for Applied Information Processing and Communications
Graz University of Technology, Inffeldgasse 16a, A–8010 Graz, Austria
[2] Department of Electrical Engineering ESAT/COSIC, Katholieke Universiteit
Leuven. Kasteelpark Arenberg 10, B–3001 Heverlee, Belgium
[3] Interdisciplinary Institute for BroadBand Technology (IBBT), Belgium
mario.lamberger@iaik.tugraz.at

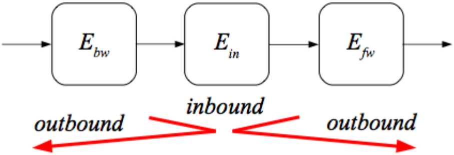
Abstract. Whirlpool is a hash function based on a block cipher that can be seen as a scaled up variant of the AES. The main difference is the (compared to AES) extremely conservative key schedule. In this work, we present a distinguishing attack on the full compression function of Whirlpool. We obtain this result by improving the rebound attack on reduced Whirlpool with two new techniques. First, the inbound phase of the rebound attack is extended by up to two rounds using the available degrees of freedom of the key schedule. This results in a near-collision attack on 9.5 rounds of the compression function of Whirlpool with a complexity of 2^{176} and negligible memory requirements. Second, we show how to turn this near-collision attack into a distinguishing attack for the full 10 round compression function of Whirlpool. This is the first result on the full Whirlpool compression function.

The rebound attack

Main result:

- The rebound attack

Directly exploit degrees of freedom in the “middle” to satisfy low-probability characteristics (match-in-the-middle)

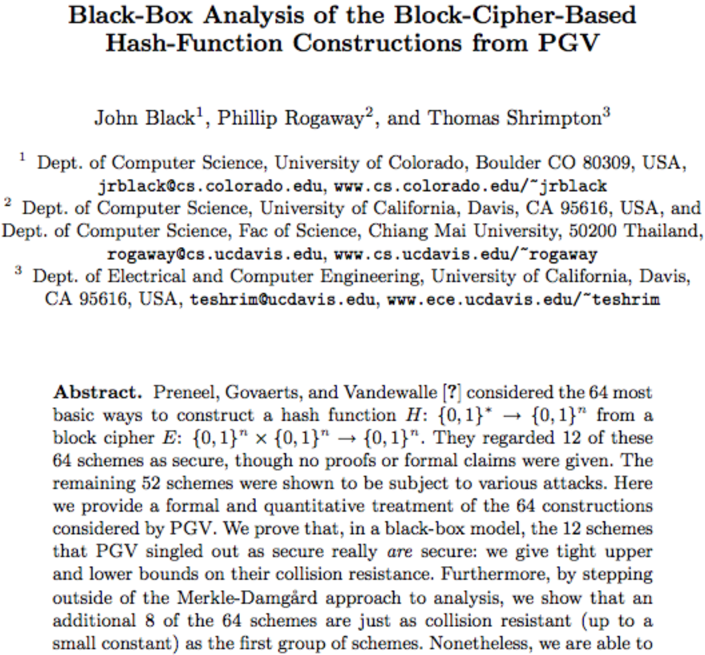


Applied to Whirlpool, and to the SHA-3 candidates ECHO, JH, Groestl, LANE, and Twister

Part Two

New paradigms

CRYPTO 2002



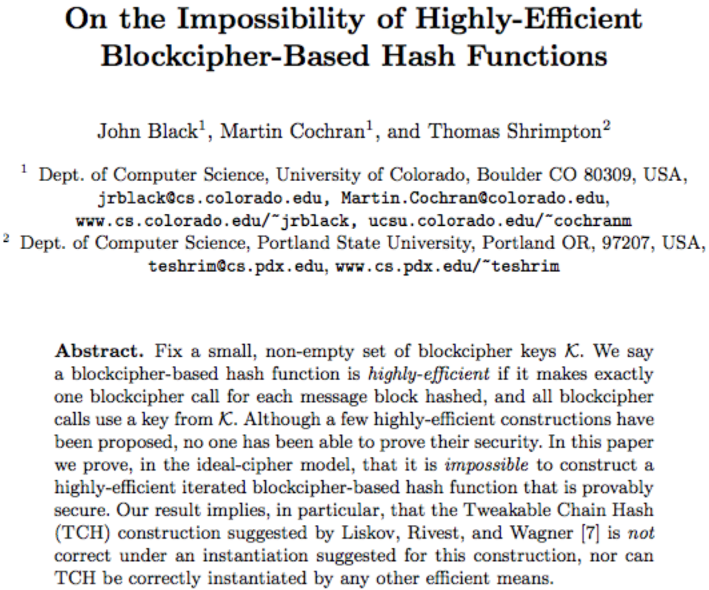
Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV

John Black[1], Phillip Rogaway[2], and Thomas Shrimpton[3]

[1] Dept. of Computer Science, University of Colorado, Boulder CO 80309, USA,
jrblack@cs.colorado.edu, www.cs.colorado.edu/~jrblack
[2] Dept. of Computer Science, University of California, Davis, CA 95616, USA, and
Dept. of Computer Science, Fac of Science, Chiang Mai University, 50200 Thailand,
rogaway@cs.ucdavis.edu, www.cs.ucdavis.edu/~rogaway
[3] Dept. of Electrical and Computer Engineering, University of California, Davis,
CA 95616, USA, teshrim@ucdavis.edu, www.ece.ucdavis.edu/~teshrim

Abstract. Preneel, Govaerts, and Vandewalle [?] considered the 64 most basic ways to construct a hash function $H: \{0,1\}^* \to \{0,1\}^n$ from a block cipher $E: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. They regarded 12 of these 64 schemes as secure, though no proofs or formal claims were given. The remaining 52 schemes were shown to be subject to various attacks. Here we provide a formal and quantitative treatment of the 64 constructions considered by PGV. We prove that, in a black-box model, the 12 schemes that PGV singled out as secure really *are* secure: we give tight upper and lower bounds on their collision resistance. Furthermore, by stepping outside of the Merkle-Damgård approach to analysis, we show that an additional 8 of the 64 schemes are just as collision resistant (up to a small constant) as the first group of schemes. Nonetheless, we are able to

EUROCRYPT 2005



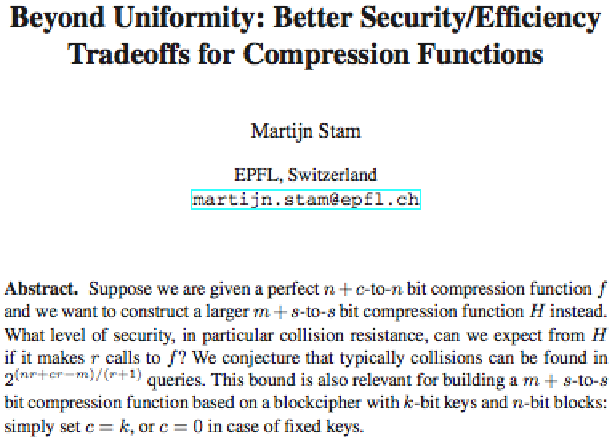
On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions

John Black[1], Martin Cochran[1], and Thomas Shrimpton[2]

[1] Dept. of Computer Science, University of Colorado, Boulder CO 80309, USA,
jrblack@cs.colorado.edu, Martin.Cochran@colorado.edu,
www.cs.colorado.edu/~jrblack, ucsu.colorado.edu/~cochranm
[2] Dept. of Computer Science, Portland State University, Portland OR, 97207, USA,
teshrim@cs.pdx.edu, www.cs.pdx.edu/~teshrim

Abstract. Fix a small, non-empty set of blockcipher keys $\mathcal{K}$. We say a blockcipher-based hash function is *highly-efficient* if it makes exactly one blockcipher call for each message block hashed, and all blockcipher calls use a key from $\mathcal{K}$. Although a few highly-efficient constructions have been proposed, no one has been able to prove their security. In this paper we prove, in the ideal-cipher model, that it is *impossible* to construct a highly-efficient iterated blockcipher-based hash function that is provably secure. Our result implies, in particular, that the Tweakable Chain Hash (TCH) construction suggested by Liskov, Rivest, and Wagner [7] is *not* correct under an instantiation suggested for this construction, nor can TCH be correctly instantiated by any other efficient means.

CRYPTO 2008



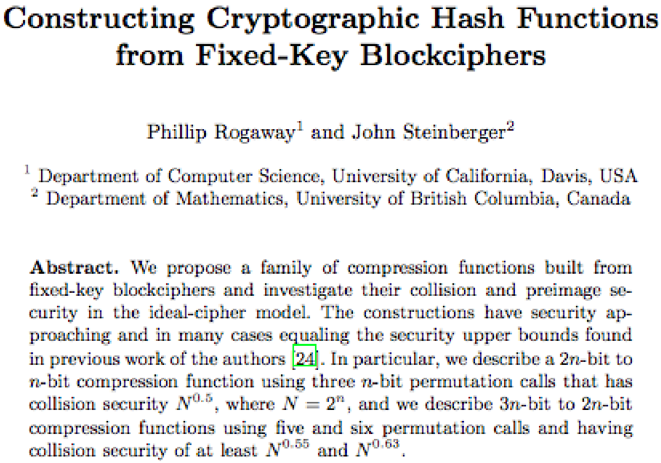
Beyond Uniformity: Better Security/Efficiency Tradeoffs for Compression Functions

Martijn Stam

EPFL, Switzerland
martijn.stam@epfl.ch

Abstract. Suppose we are given a perfect $n + c$-to-n bit compression function f and we want to construct a larger $m + s$-to-s bit compression function H instead. What level of security, in particular collision resistance, can we expect from H if it makes r calls to f? We conjecture that typically collisions can be found in $2^{(nr+cr-m)/(r+1)}$ queries. This bound is also relevant for building a $m + s$-to-s bit compression function based on a blockcipher with k-bit keys and n-bit blocks: simply set $c = k$, or $c = 0$ in case of fixed keys.

CRYPTO 2008



Constructing Cryptographic Hash Functions from Fixed-Key Blockciphers

Phillip Rogaway[1] and John Steinberger[2]

[1] Department of Computer Science, University of California, Davis, USA
[2] Department of Mathematics, University of British Columbia, Canada

Abstract. We propose a family of compression functions built from fixed-key blockciphers and investigate their collision and preimage security in the ideal-cipher model. The constructions have security approaching and in many cases equaling the security upper bounds found in previous work of the authors [24]. In particular, we describe a $2n$-bit to n-bit compression function using three n-bit permutation calls that has collision security $N^{0.5}$, where $N = 2^n$, and we describe $3n$-bit to $2n$-bit compression functions using five and six permutation calls and having collision security of at least $N^{0.55}$ and $N^{0.63}$.

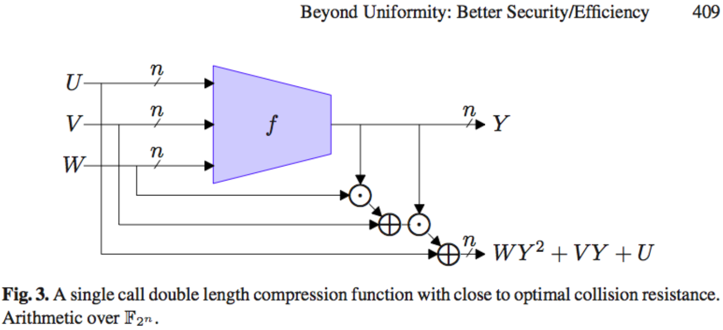
Results on compression function constructions

Main results

- Better understanding of blockcipher-based hashing
- Results for (e.g.) compression function combiners

Example of impossibility: compression function with one call to a fixed key block cipher [Black-Cochran-Shrimpton-05]

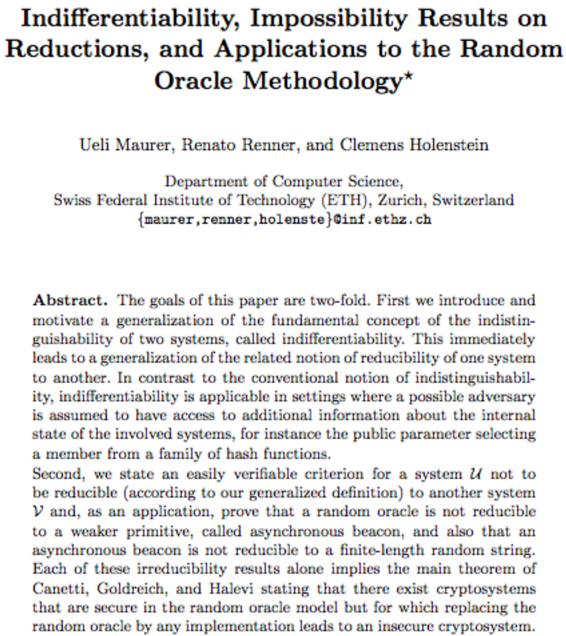
Many new constructions proposed, e.g.



Beyond Uniformity: Better Security/Efficiency 409

Fig. 3. A single call double length compression function with close to optimal collision resistance. Arithmetic over $\mathbb{F}_{2^n}$.

TCC 2004



Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology*

Ueli Maurer, Renato Renner, and Clemens Holenstein

Department of Computer Science,
Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
{maurer,renner,holenste}@inf.ethz.ch

Abstract. The goals of this paper are two-fold. First we introduce and motivate a generalization of the fundamental concept of the indistinguishability of two systems, called indifferentiability. This immediately leads to a generalization of the related notion of reducibility of one system to another. In contrast to the conventional notion of indistinguishability, indifferentiability is applicable in settings where a possible adversary is assumed to have access to additional information about the internal state of the involved systems, for instance the public parameter selecting a member from a family of hash functions.

Second, we state an easily verifiable criterion for a system $\mathcal{U}$ not to be reducible (according to our generalized definition) to another system $\mathcal{V}$ and, as an application, prove that a random oracle is not reducible to a weaker primitive, called asynchronous beacon, and also that an asynchronous beacon is not reducible to a finite-length random string. Each of these irreducibility results alone implies the main theorem of Canetti, Goldreich, and Halevi stating that there exist cryptosystems that are secure in the random oracle model but for which replacing the random oracle by any implementation leads to an insecure cryptosystem.

Indifferentiability

Main result:

- Notion of indifferentiability and proof strategies

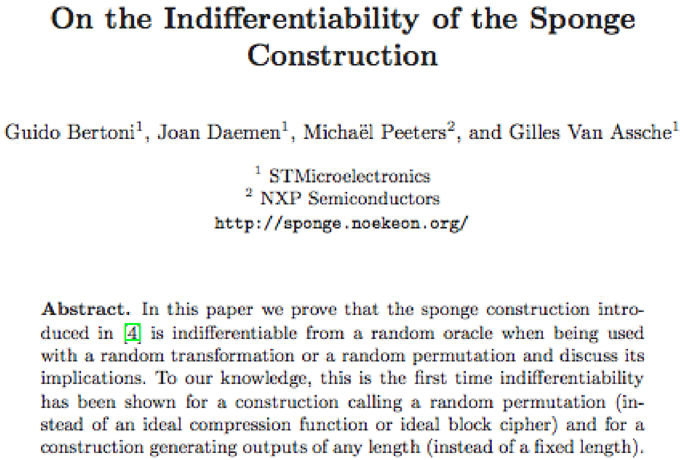
"Ultimate" notion of security for operation modes

For a hash, says that if the compression function has no structural flaw, then the hash function resists any attack

Useful for cryptanalysis (any flaw must be in the compression algorithm)

Proofs sometimes difficult to verify...

EUROCRYPT 2008



On the Indifferentiability of the Sponge Construction

Guido Bertoni[1], Joan Daemen[1], Michaël Peeters[2], and Gilles Van Assche[1]

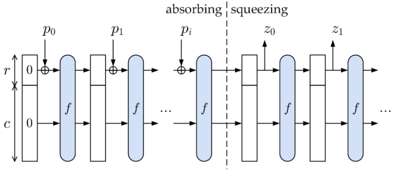
[1] STMicroelectronics
[2] NXP Semiconductors
http://sponge.noekeon.org/

Abstract. In this paper we prove that the sponge construction introduced in [4] is indifferentiable from a random oracle when being used with a random transformation or a random permutation and discuss its implications. To our knowledge, this is the first time indifferentiability has been shown for a construction calling a random permutation (instead of an ideal compression function or ideal block cipher) and for a construction generating outputs of any length (instead of a fixed length).

Sponge functions

Main results:

- Definition of the sponge construction for hash function
- Proof of indifferentiability



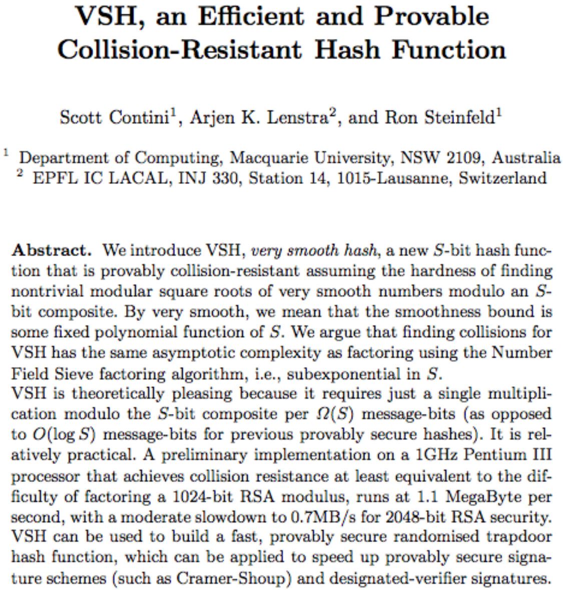
First real alternative to the DM operation mode

First distinction of security (“capacity”) and digest length

High flexibility (block length, digest length, security)

Needs larger state, but no “feedforward” is needed

EUROCRYPT 2006



VSH, an Efficient and Provable Collision-Resistant Hash Function

Scott Contini[1], Arjen K. Lenstra[2], and Ron Steinfeld[1]

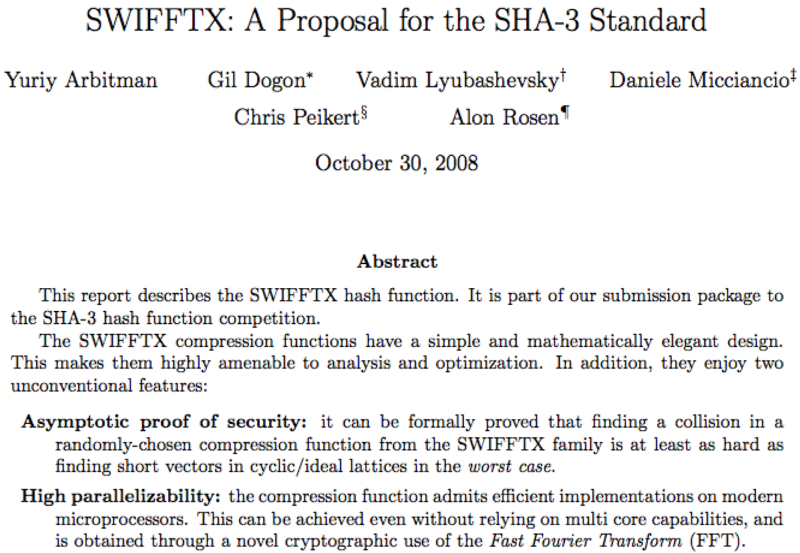
[1] Department of Computing, Macquarie University, NSW 2109, Australia
[2] EPFL IC LACAL, INJ 330, Station 14, 1015-Lausanne, Switzerland

Abstract. We introduce VSH, *very smooth hash*, a new S-bit hash function that is provably collision-resistant assuming the hardness of finding nontrivial modular square roots of very smooth numbers modulo an S-bit composite. By very smooth, we mean that the smoothness bound is some fixed polynomial function of S. We argue that finding collisions for VSH has the same asymptotic complexity as factoring using the Number Field Sieve factoring algorithm, i.e., subexponential in S.

VSH is theoretically pleasing because it requires just a single multiplication modulo the S-bit composite per $\Omega(S)$ message-bits (as opposed to $O(\log S)$ message-bits for previous provably secure hashes). It is relatively practical. A preliminary implementation on a 1GHz Pentium III processor that achieves collision resistance at least equivalent to the difficulty of factoring a 1024-bit RSA modulus, runs at 1.1 MegaByte per second, with a moderate slowdown to 0.7MB/s for 2048-bit RSA security.

VSH can be used to build a fast, provably secure randomised trapdoor hash function, which can be applied to speed up provably secure signature schemes (such as Cramer-Shoup) and designated-verifier signatures.

SHA-3 submission



SWIFFTX: A Proposal for the SHA-3 Standard

Yuriy Arbitman Gil Dogon[*] Vadim Lyubashevsky[†] Daniele Micciancio[‡]
Chris Peikert[§] Alon Rosen[¶]

October 30, 2008

Abstract

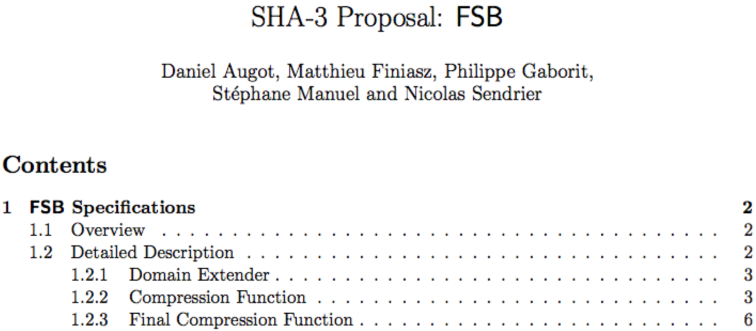
This report describes the SWIFFTX hash function. It is part of our submission package to the SHA-3 hash function competition.

The SWIFFTX compression functions have a simple and mathematically elegant design. This makes them highly amenable to analysis and optimization. In addition, they enjoy two unconventional features:

Asymptotic proof of security: it can be formally proved that finding a collision in a randomly-chosen compression function from the SWIFFTX family is at least as hard as finding short vectors in cyclic/ideal lattices in the *worst case*.

High parallelizability: the compression function admits efficient implementations on modern microprocessors. This can be achieved even without relying on multi core capabilities, and is obtained through a novel cryptographic use of the *Fast Fourier Transform* (FFT).

SHA-3 submission



SHA-3 Proposal: FSB

Daniel Augot, Matthieu Finiasz, Philippe Gaborit,
Stéphane Manuel and Nicolas Sendrier

Contents

Provably secure hashes

Main results:

- Reductions of factoring, SVP, decoding
- Significant efforts to improve efficiency

Significant progress compared to previous (broken) approaches

Simple designs (e.g., FSB is essentially XORs)

Current limitations: security against non-proved notions, efficiency

PART THREE

From SHA-2 to SHA-3

The NIST Hash Competition

Oct 2008	deadline for submissions, 64 received
Feb 2009	First SHA-3 Conference (Leuven, Belgium)
Jul 2009	14 second round candidates selected
Aug 2010	Second SHA-3 Conference (Santa Barbara, USA)
fall 2010	selection of $\approx$ 5 finalists
early 2012	Final SHA-3 Conference

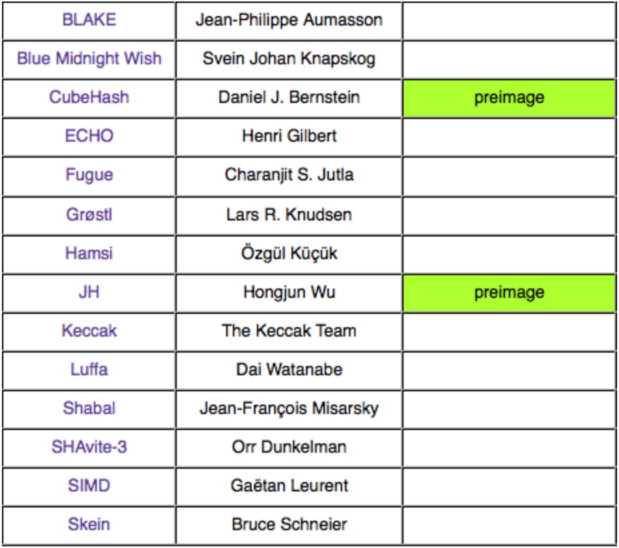
SHA-3 must support 224, 256, 384, and 512-bit digests

Most submission from academia, a few from industry (Sony, IBM, Intel, Hitachi, etc.)

Specification, attacks, etc. published on ECRYPT's SHA-3 Zoo

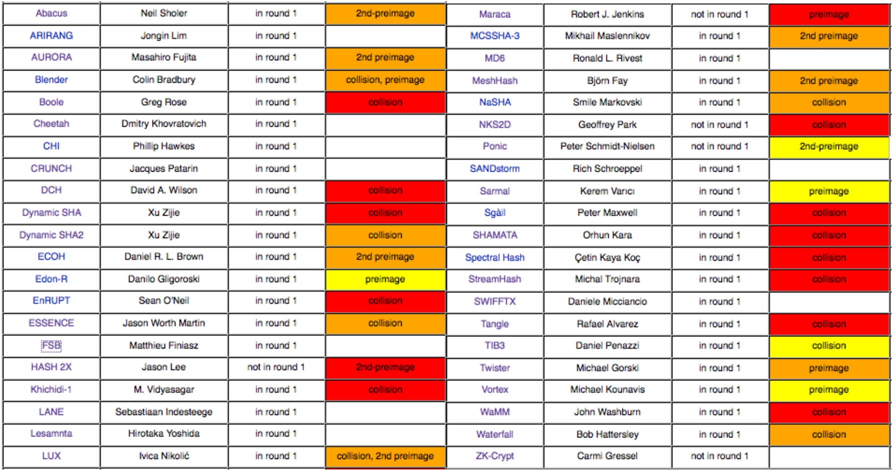
`http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`

The 14 second round candidates



BLAKE	Jean-Philippe Aumasson	
Blue Midnight Wish	Svein Johan Knapskog	
CubeHash	Daniel J. Bernstein	preimage
ECHO	Henri Gilbert	
Fugue	Charanjit S. Jutla	
Grøstl	Lars R. Knudsen	
Hamsi	Özgül Küçük	
JH	Hongjun Wu	preimage
Keccak	The Keccak Team	
Luffa	Dai Watanabe	
Shabal	Jean-François Misarsky	
SHAvite-3	Orr Dunkelman	
SIMD	Gaëtan Leurent	
Skein	Bruce Schneier	

The 42+8 NOT second round candidates



Abacus	Neil Sholer	in round 1	2nd-preimage	Maraca	Robert J. Jenkins	not in round 1	preimage
ARIRANG	Jongin Lim	in round 1		MCSSHA-3	Mikhail Maslennikov	in round 1	2nd preimage
AURORA	Masahiro Fujita	in round 1	2nd preimage	MD6	Ronald L. Rivest	in round 1	
Blender	Colin Bradbury	in round 1	collision, preimage	MeshHash	Björn Fay	in round 1	2nd preimage
Boole	Greg Rose	in round 1	collision	NaSHA	Smile Markovski	in round 1	collision
Cheetah	Dmitry Khovratovich	in round 1		NKS2D	Geoffrey Park	not in round 1	collision
CHI	Phillip Hawkes	in round 1		Ponic	Peter Schmidt-Nielsen	not in round 1	2nd-preimage
CRUNCH	Jacques Patarin	in round 1		SANDstorm	Rich Schroeppel	in round 1	
DCH	David A. Wilson	in round 1	collision	Sarmal	Kerem Varıcı	in round 1	preimage
Dynamic SHA	Xu Zijie	in round 1	collision	Sgàil	Peter Maxwell	in round 1	collision
Dynamic SHA2	Xu Zijie	in round 1	collision	SHAMATA	Orhun Kara	in round 1	collision
ECOH	Daniel R. L. Brown	in round 1	2nd preimage	Spectral Hash	Çetin Kaya Koç	in round 1	collision
Edon-R	Danilo Gligoroski	in round 1	preimage	StreamHash	Michal Trojnara	in round 1	collision
EnRUPT	Sean O'Neil	in round 1	collision	SWIFFTX	Daniele Micciancio	in round 1	
ESSENCE	Jason Worth Martin	in round 1	collision	Tangle	Rafael Alvarez	in round 1	collision
FSB	Matthieu Finiasz	in round 1		TIB3	Daniel Penazzi	in round 1	collision
HASH 2X	Jason Lee	not in round 1	2nd-preimage	Twister	Michael Gorski	in round 1	preimage
Khichidi-1	M. Vidyasagar	in round 1	collision	Vortex	Michael Kounavis	in round 1	preimage
LANE	Sebastiaan Indesteege	in round 1		WaMM	John Washburn	in round 1	collision
Lesamnta	Hirotaka Yoshida	in round 1		Waterfall	Bob Hattersley	in round 1	collision
LUX	Ivica Nikolić	in round 1	collision, 2nd preimage	ZK-Crypt	Carmi Gressel	not in round 1	

Observations so far

Great diversity of designs:

- HAIFA, sponge, variants thereof, tree-based, etc.
- AES-based, AXR, AND/XOR, Serpent-like, etc.

Proofs do not help much to survive

New attack proposed (rebound, linearization, zero-sum,)

Designers have no right to err (any “flaw” can be fatal)

No single candidate stands out as the favorite

Epilogue

A golden decade

Much more results in the last 10 years than in $[-\infty; 2000]$

Very rich decade for hash functions, both for the “theory” and “applied” sides

With a good message expansion in SHA-0 and 128 rounds in MD5 from the beginning, we wouldn’t have needed to worry (and there would probably be no SHA-3 competition)

Next expected breakthrough: collision for SHA-1?

10 years of cryptographic hashing

Jean-Philippe Aumasson

HASH, x. There is no definition for this word—nobody knows what hash is.

Ambrose Bierce, *The Devil's Dictionary*