Tools in Cryptanalysis of Hash Functions

Application to SHA-256

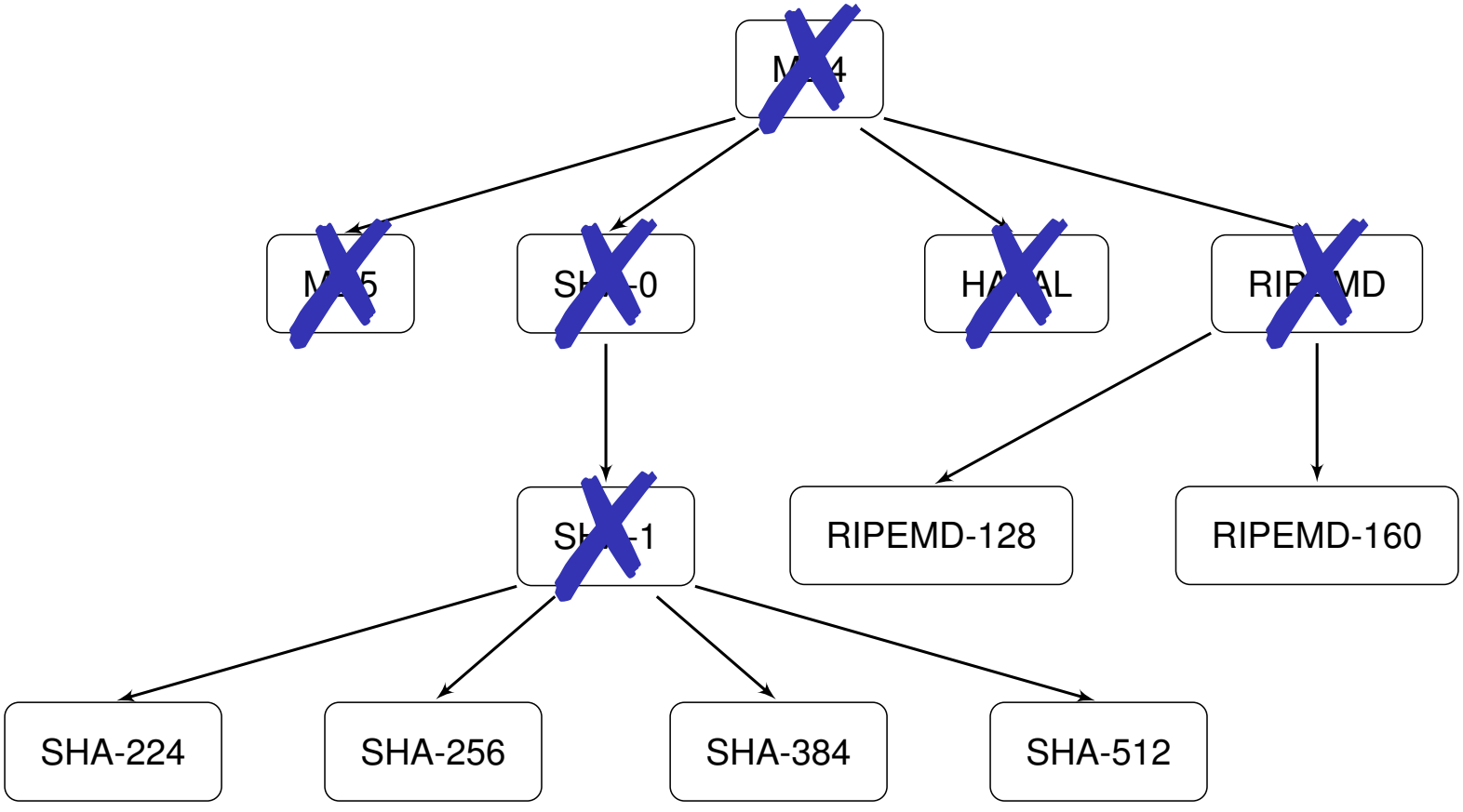
Florian Mendel

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology
Inffeldgasse 16a, A-8010 Graz, Austria

Outline

Outline

Attacks on the MD4-family



Consequences of the Attacks

Transition from SHA-1 to SHA-2

- NIST proposed the transition from SHA-1 to the SHA-2 family
- Companies and organization are expected to migrate to SHA-2

SHA-3 initiative

- Researchers were evaluating alternative hash functions in the SHA-3 initiative organized by NIST
- NIST selected Keccak as SHA-3

Results for SHA-256

Preimage Attack

- Aoki et al. [AGM+09]
  - 43 out of 64 steps (complexity: $2^{254.9}$)
- Khovratovich et al. [KRS12]
  - 45 out of 64 steps (complexity: $2^{255.5}$)

Collision Attack

- Nikolić and Biryukov [NB08]
  - 21 out of 64 steps (example)
- Indesteege et al. [IMPR08]; Sanadhya and Sarkar [SS08]
  - 24 out of 64 steps (example)

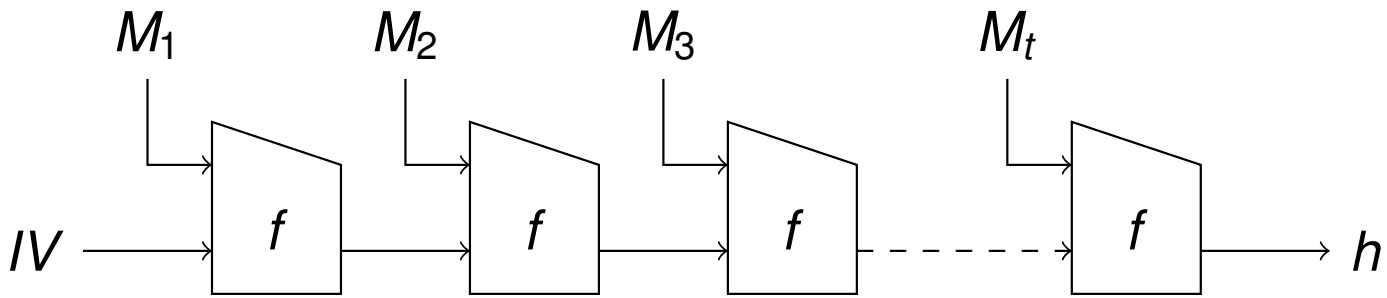
Outline

The SHA-2 Family

- Designed by NSA and issued by NIST in 2002.
- Defined in the Federal Information Processing Standard (FIPS-180-3)
- Part of several international standards
- Often recommended as an alternative to SHA-1
- Consists of 4 hash functions, i.e. SHA-224, SHA-256, SHA-384, SHA-512

Description of SHA-256

- Iterated hash function processing message blocks of 512 bits and producing a hash value of 256 bits.
- Compression function f consists of 2 parts:
  - Message Expansion
  - State Update (64 steps)



Message Expansion

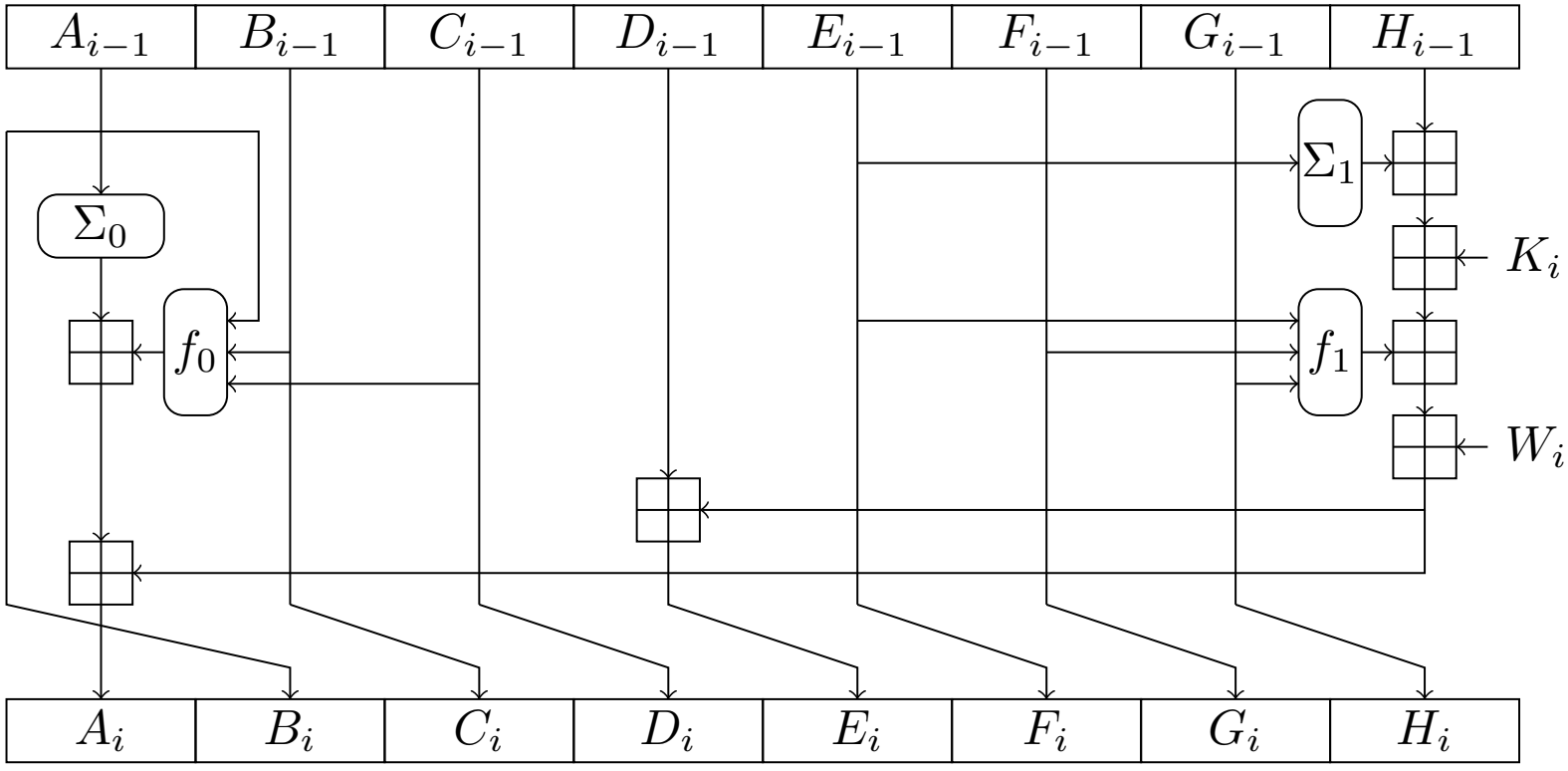
The message expansion of SHA-256 splits the 512-bit message block into 16 words M_i, $i = 0, \ldots, 15$, and expands them into 64 expanded message words W_i as follows:

$$W_i = \begin{cases} M_i & 0 \leq i < 16 \\ \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16} & 16 \leq i < 64 \end{cases}$$

The functions $\sigma_0(X)$ and $\sigma_1(X)$ are given by

$$\begin{aligned} \sigma_0(X) &= (X \ggg 7) \oplus (X \ggg 18) \oplus (X \gg 3) \\ \sigma_1(X) &= (X \ggg 17) \oplus (X \ggg 19) \oplus (X \gg 10) \end{aligned}$$

Step Function of SHA-256



Step Function of SHA-256

- The bitwise Boolean functions f_0 and f_1 used in each step are defined as follows:

$$\begin{aligned} f_0(X, Y, Z) &= X \wedge Y \oplus Y \wedge Z \oplus X \wedge Z \\ f_1(X, Y, Z) &= X \wedge Y \oplus \neg X \wedge Z \end{aligned}$$

- The linear functions Σ_0 and Σ_1 are defined as follows:

$$\begin{aligned} \Sigma_0(X) &= (X \ggg 2) \oplus (X \ggg 13) \oplus (X \ggg 22) \\ \Sigma_1(X) &= (X \ggg 6) \oplus (X \ggg 11) \oplus (X \ggg 25) \end{aligned}$$

Outline

Our Contribution

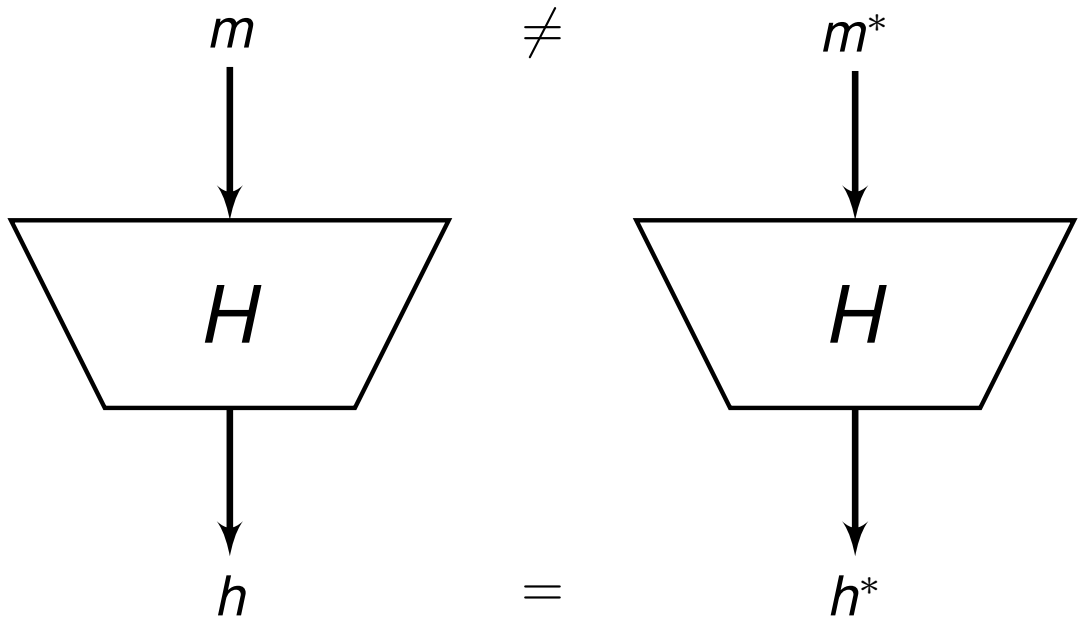
Advanced Automatic Search Tool

- Finding complex differential characteristics for SHA-2 automatically
- Similar to the one for SHA-1 by De Cannière and Rechberger [DR06]

Collisions Attacks on SHA-256

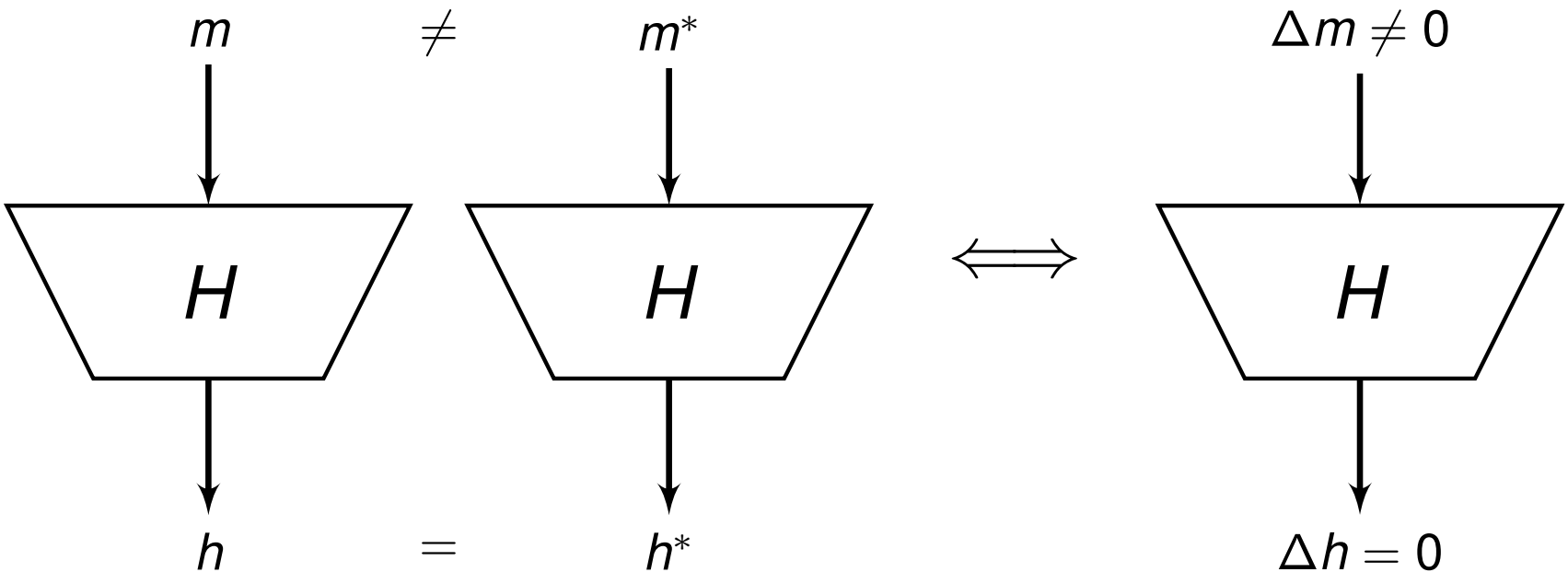
- Collisions for up to 38 steps of the compression function
- Collisions for up to 31 steps of the hash function

Collision Attacks



- Birthday Attack: $2^{n/2}$

Collision Attacks (Differential View)



- Find a differential characteristic which results in a collision with a good probability
- Find a message m following the differential characteristic to get a colliding message pair (m, m^*)

Collision Attacks on SHA-256

- All collisions attacks so far are of practical complexity
- They are all based on the same basic idea: extending a local collision over 9 steps to more steps
- The best collision attack so far is for 24 steps based on the 9-step differential characteristic of Nikolić and Biryukov

Basic Attack Strategy

- To find collisions for more than 24 steps, we need differential characteristics spanning over $t > 9$ steps
- To find these characteristics we proceed as follows:
  (1) Fix the value of t
  (2) Identify those message words which need to have differences to result in a valid differential characteristic for the message expansion
  (3) Consider only the candidates that may result in a collision for more than 24 steps
  (4) Use an automatic search tool to construct a valid differential characteristic for both the state update transformation and the message expansion

Candidate for 27 Steps

- For $t = 10$ we already find a candidate which may result in a collision for 27 steps

step	ΔA	ΔB	ΔC	ΔD	ΔE	ΔF	ΔG	ΔH	ΔW
4									?
5	?				?				
6	?	?			?	?			
7		?	?		?	?	?		
8			?	?	?	?	?	?	
9				?	?	?	?	?	
10					?	?	?	?	
11						?	?	?	
12							?	?	?
13								?	?
14									

W	4	12	13
0			
1			
2			
3			
4	x		
5			
6			
7			
8			
9			
10			
11			
12		x	
13			x
14			
15			
16			
17			
18			
19	x	x	
20	x		x
21			
22			
23			
24			
25			
26			

Finding Differential Characteristics

- These characteristics can not be constructed manually
- A sophisticated automatic search tool is needed to construct these characteristics
  - Gröbner Basis, SAT solvers, . . .
  - Dedicated Approach [DR06] (Guess-and-Determine)

Guess-and-Determine Attack

On a high level, a guess-and-determine attack can be described as a repetition of the following two steps

- guess the value of some unknowns
- determine the value of as many unknowns as is possible

until all unknowns have been determined

Guess-and-Determine Attack

A guess-and-determine attack works specially well if there are

- many sparse equations
- the set of equations can be split into a number of subsets with very few variables occurring in more than one subset

$\Rightarrow$ A successful attack employs a strategy to convert the complex and dense equations into a form that is more amenable to attack

De Cannière and Rechberger Approach for SHA-1

Generalized Conditions

- All 16 possible conditions on a pair of bits are taken into account.

(X_i, X_i^*)	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
?	✓	✓	✓	✓
–	✓	-	-	✓
x	-	✓	✓	-
0	✓	-	-	-
u	-	✓	-	-
n	-	-	✓	-
1	-	-	-	✓
#	-	-	-	-

(X_i, X_i^*)	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
3	✓	✓	-	-
5	✓	-	✓	-
7	✓	✓	✓	-
A	-	✓	-	✓
B	✓	✓	-	✓
C	-	-	✓	✓
D	✓	-	✓	✓
E	-	✓	✓	✓

De Cannière and Rechberger Approach for SHA-1

Search Algorithm

(1) Start with an unrestricted characteristic (only '?')

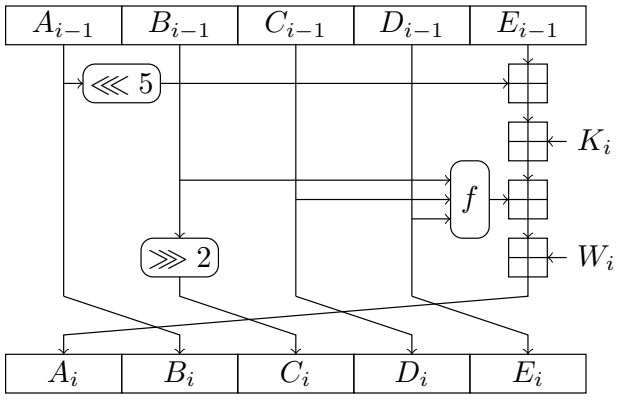
(2) Successively impose new conditions on the characteristic (replace '?' by '−' and 'x' by 'n' or 'u')

(3) Propagate the conditions in a bitslice manner and check for consistency
   - If a contradiction occurs then backtrack
   - else proceed with step 2

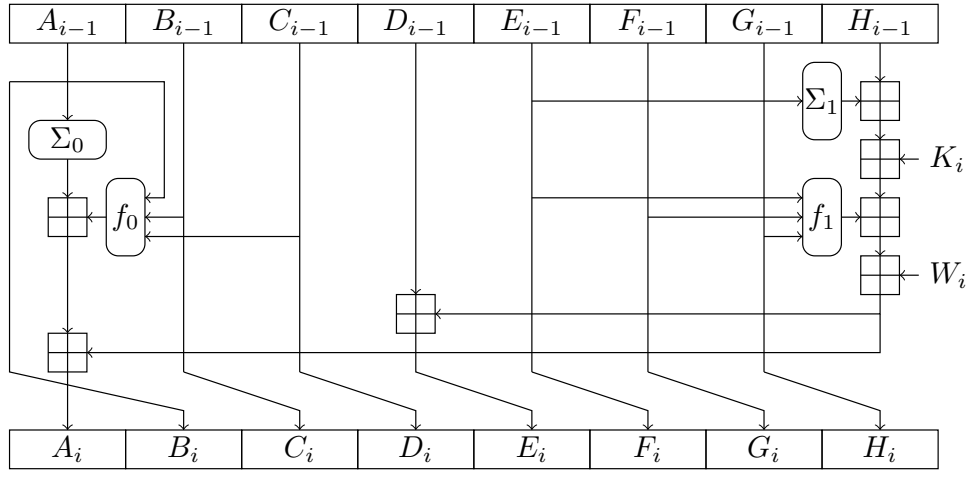
(4) Repeat steps 2 and 3 until all bits of the characteristic are determined

Increased Complexity of SHA-2

SHA-1



SHA-2



Design Complexity

How to overcome the problems?

- Use alternative description of state update
- Identify more complex conditions involving several bits
- Use modified search algorithm
  - Combine search for differential characteristic and message pair
  - Apply sophisticated tests to detect contradictions earlier

Example

Collision for 27 Steps of SHA-256 Compression Function

Candidate for 27 Steps

- For $t = 10$ we already find a candidate which may result in a collision for 27 steps

step	ΔA	ΔB	ΔC	ΔD	ΔE	ΔF	ΔG	ΔH	ΔW
4									?
5	?				?				
6	?	?			?	?			
7		?	?		?	?	?		
8			?	?	?	?	?	?	
9				?	?	?	?	?	
10					?	?	?	?	
11						?	?	?	
12							?	?	?
13								?	?
14									

W	4	12	13
0			
1			
2			
3			
4	x		
5			
6			
7			
8			
9			
10			
11			
12		x	
13			x
14			
15			
16			
17			
18			
19	x	x	
20	x		x
21			
22			
23			
24			
25			
26			

i	∇A_i	∇E_i	∇W_i
-4	----------	----------	
-3	----------	----------	
-2	----------	----------	
-1	----------	----------	
0	----------	----------	----------
1	----------	----------	----------
2	----------	----------	----------
3	----------	----------	----------
4	??????????	??????????	??????????
5	??????????	??????????	----------
6	----------	??????????	----------
7	----------	??????????	----------
8	----------	??????????	----------
9	----------	??????????	----------
10	----------	----------	----------
11	----------	----------	----------
12	----------	----------	??????????
13	----------	----------	??????????
14	----------	----------	----------
15	----------	----------	----------
16	----------	----------	----------
17	----------	----------	----------
18	----------	----------	----------
19	----------	----------	----------
20	----------	----------	----------
21	----------	----------	----------
22	----------	----------	----------
23	----------	----------	----------
24	----------	----------	----------
25	----------	----------	----------
26	----------	----------	----------

freebits: 352
backtrack: 0

Results for the Compression Function

- Example of a collision for 27 steps of the compression function (seconds on a standard PC)

h_0	4d031285	26b1c18f	c8c014f2	3cca74bd	58481e1b	c7dd5a1e	0ae3c962	e01f0e96
m	e4e9607f 97121d5e	b6fb6b22 35214e08	01597e95 174b5fbb	5265b614 1dc549e5	d4dbb9af 7e5b858a	8a228a75 c966e506	3c660afd faac3dbc	55b668bc 9df96855
m^*	e4e9607f 97121d5e	b6fb6b22 35214e08	01597e95 174b5fbb	5265b614 1dc549e5	d543baaf 7e0bb58a	8a228a75 c8fee406	3c660afd faac3dbc	55b668bc 9df96855
Δm	00000000 00000000	00000000 00000000	00000000 00000000	00000000 00000000	01980300 00503000	00000000 01980100	00000000 00000000	00000000 00000000
h_1	9f9579f2	2737d03f	20263c5b	1b802daf	0b5e24ad	9eed0964	6bb8f239	2a4c60f7

Extending the Attack to more Steps

- Example of a collision for 38 steps of the compression function (about 8 hours on a standard PC)

h_0	ba75b4ac	c3c9fd45	fce04f3a	6d620fdb	42559d01	b0a0cd10	729ca9bc	b284a572
m	4f5267f8	8f8ec13b	22371c61	56836f2b	459501d1	8078899e	98947e61	4015ef31
	06e98ffc	4babda4a	27809447	3bf9f3be	7b3b74e1	065f711d	6c6ead5e	a1781d54
m^*	4f5267f8	8f8ec13b	22371c61	56836f2b	459501d1	8078899e	98947e61	7e73f1f1
	06e99000	4babda4a	277f1447	3bf9f3be	7b3b74e1	065f711d	6c6ead5e	a1781d50
Δm	00000000	00000000	00000000	00000000	00000000	00000000	00000000	3e661ec0
	00001ffc	00000000	00ff8000	00000000	00000000	00000000	00000000	00000004
h_1	baa8df17	9f9f64dd	d57d5c2c	7b232c81	1f3916e6	7a03a2be	7afb1d86	6b0eced6

Results for the Hash Function

Extending the Attack to the Hash Function

- Approach of Indesteege et al. [IMPR08]
  - Construct a collision for the compression function (with no differences in the first message words)
  - Use the freedom in the first message words to turn it into a collision for the hash function

$\Rightarrow$ Collision attack on 28 steps of SHA-256

W	8	9	13	16	18
0					
1					
2					
3					
4					
5					
6					
7					
8	x				
9		x			
10					
11					
12					
13			x		
14					
15					
16		x		x	
17					
18				x	x
19					
20			x		x
21					
22					
23	x			x	
24	x	x			
25		x			x
26					
27					

Results for the Hash Function

- Example of a collision for 28 steps of the hash function

h_0	6a09e667	bb67ae85	3c6ef372	a54ff53a	510e527f	9b05688c	1f83d9ab	5be0cd19
m	14c48440	b3c3277f	ad69812d	c3d4dffa	7eae690b	7f9fe027	832aece8	9a489458
	1607a45c	db81bdc8	8786e031	d8f22801	72b6be5e	45a2652f	f3fbb17a	2ce70f52
m^*	14c48440	b3c3277f	ad69812d	c3d4dffa	7eae690b	7f9fe027	832aece8	9a489458
	e6b2f4fc	d759b930	8786e031	d8f22801	72b6be5e	47e26dbf	f3fbb17a	2ce70f52
Δm	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	f0b550a0	0cd804f8	00000000	00000000	00000000	02400890	00000000	00000000
h_1	01470131	cd0062bc	7e8f8c21	98938652	3d49075a	327f38e8	11f0d36d	58601725

- Theoretical collision attack on SHA-256 reduced to 31 steps (complexity $2^{65.5}$)

Outline

RIPEMD-128/160

- Designed by Dobbertin, Bosselaers and Preneel in 1996
- ISO/IEC 10118-3 standard on dedicated hash function
- Similar design principle as MD5 and SHA-1
- Results:

	component	attack	steps	complexity
RIPEMD-128	compression	collision	48	example
	hash	collision	38	example
	hash	near-collision	44	example
	hash	non-randomness	48	2^{70}
RIPEMD-160	compression	collision	48	example

Other

HAS-160

- Standardized by the Korean government
- Similar design principle as SHA-1
- Results:

component	attack	steps	complexity
compression	collision	65	example

SM3

- Standardized by the Chinese government
- Similar design principle as SHA-256
- Results:

component	attack	steps	complexity
compression	collision	24	example
hash	collision	20	example

Outline

Summary and Future Work

Summary

- Sophisticated tool to construct complex differential characteristics
- Attacks on several popular hash functions
  - SHA-256, RIPEMD-128/160, SM3, ...

Current/Future Work

- Application to other hash functions
- Analysis of keyed functions, e.g. HMAC

$\Rightarrow$ Still lots of work to be done!

Thank you for your attention!

References I


Kazumaro Aoki, Jian Guo, Krystian Matusiewicz, Yu Sasaki, and Lei Wang.
Preimages for Step-Reduced SHA-2.
In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 578–597. Springer, 2009.


Alex Biryukov, Mario Lamberger, Florian Mendel, and Ivica Nikolic.
Second-Order Differential Collisions for Reduced SHA-256.
In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *LNCS*, pages 270–287. Springer, 2011.


Christophe De Cannière and Christian Rechberger.
Finding SHA-1 Characteristics: General Results and Applications.
In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *LNCS*, pages 1–20. Springer, 2006.


Sebastiaan Indesteege, Florian Mendel, Bart Preneel, and Christian Rechberger.
Collisions and Other Non-random Properties for Step-Reduced SHA-256.
In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *LNCS*, pages 276–293. Springer, 2008.


Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva.
Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family.
In Anne Canteaut, editor, *FSE*, volume 7549 of *LNCS*, pages 244–263. Springer, 2012.


Florian Mendel, Tomislav Nad, and Martin Schläffer.
Cryptanalysis of Round-Reduced HAS-160.
In Howon Kim, editor, *ICISC*, volume 7259 of *LNCS*, pages 33–47. Springer, 2011.

References II


Florian Mendel, Tomislav Nad, and Martin Schläffer.
Finding SHA-2 Characteristics: Searching through a Minefield of Contradictions.
In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *LNCS*, pages 288–307. Springer, 2011.


Florian Mendel, Tomislav Nad, and Martin Schläffer.
Collision Attacks on the Reduced Dual-Stream Hash Function RIPEMD-128.
In Anne Canteaut, editor, *FSE*, volume 7549 of *LNCS*, pages 226–243. Springer, 2012.


Florian Mendel, Tomislav Nad, and Martin Schläffer.
Finding Collisions for Round-Reduced SM3.
In Ed Dawson, editor, *Topics in Cryptology - CT-RSA 2013*, volume 7779 of *LNCS*, pages 174 – 188. Springer, 2013.


Florian Mendel, Tomislav Nad, and Martin Schläffer.
Improving Local Collisions: New Attacks on Reduced SHA-256.
In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, LNCS. Springer, 2013. To appear.


Florian Mendel, Tomislav Nad, Stefan Scherz, and Martin Schläffer.
Differential Attacks on Reduced RIPEMD-160.
In Dieter Gollmann and Felix C. Freiling, editors, *ISC*, volume 7483 of *LNCS*, pages 23–38. Springer, 2012.


Ivica Nikolić and Alex Biryukov.
Collisions for Step-Reduced SHA-256.
In Kaisa Nyberg, editor, *FSE*, volume 5086 of *LNCS*, pages 1–15. Springer, 2008.

References III


Somitra Kumar Sanadhya and Palash Sarkar.
New Collision Attacks against Up to 24-Step SHA-2.
In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT*, volume 5365 of *LNCS*, pages 91–103. Springer, 2008.