

Cryptanalysis of Dynamic SHA*

Sebastiaan Indestege

COSIC, ESAT, Katholieke Universiteit Leuven, Belgium

First SHA-3 Candidate Conference
Rump Session

Dynamic SHA

- ▶ SHA-3 round 1 candidate
- ▶ Designer: Zijie Xu
- ▶ SHA-256-like structure
- ▶ 48 rounds
- ▶ Trivial message expansion (repetition)
- ▶ Modular additions, 3-input boolean functions, **data-dependent rotations**

Dynamic SHA

$a = H_0; b = H_1; c = H_2; d = H_3;$
 $e = H_4; f = H_5; g = H_6; h = H_7;$

for $t = 0$ to 47 **do**

$T = \mathbf{R}(a, b, c, d, e, f, g, h);$

$h = g; g = f; f = e; e = d;$

$d = \mathbf{G}_{t \bmod 4}(a, b, c) \boxplus W_{t \bmod 16} \boxplus TT_{\lfloor t/16 \rfloor}$

$c = b; b = a; a = T;$

end for

$H_0 \boxplus = a; H_1 \boxplus = b; H_2 \boxplus = c; H_3 \boxplus = d;$

$H \boxplus = H \boxplus = f \quad H \boxplus = H \boxplus = h$

Dynamic SHA

$$G_i(a, b, c) = \begin{cases} a \oplus b \oplus c & i = 0 \\ (a \wedge b) \oplus c & i = 1 \\ (\neg(a \vee c)) \vee (a \wedge (b \oplus c)) & i = 2 \\ (\neg(a \vee (b \oplus c))) \vee (a \wedge \neg c) & i = 3 \end{cases}$$

function $R(a, b, c, d, e, f, g, h)^\dagger$

$t = (((((a \boxplus b) \oplus c) \boxplus d) \oplus e) \boxplus f) \oplus g;$

$t = ((t \gg 17) \oplus t) \&(2^{17} - 1);$

$t = ((t \gg 10) \oplus t) \&(2^{10} - 1);$

$t = ((t \gg 5) \oplus t) \&(2^5 - 1);$

t **h** \ggg **t**;

Collision Attack

Observations on Dynamic SHA 256

$$= \left\{ \begin{array}{l} = \\ = \\ = \\ = \end{array} \right.$$

$G(\cdot)$ -functions

- ▶ Each $G(\cdot)$ -function is **linear** in c
- ▶ Each $G(\cdot)$ -function can either pass or absorb differences in a and/or b ($\text{Pr} = 1/2$)

Observations on Dynamic SHA 256

function $R(a, b, c, d, e, f, g, h)$

$$t = ((((((a \boxplus b) \oplus c) \boxplus d) \oplus e) \boxplus f) \oplus g);$$

=

=

=

R-function

- ▶ Linear in MSB of a, \dots, g
- ▶ MSB of a, \dots, g only influences MSB of t^\ddagger

Idea

- ▶ Stick to MSB differences only (modular additions: $\text{Pr} = 1$)
- ▶ Absorb or pass differences in a , b entering the $G(\cdot)$ -functions, as desired ($\text{Pr} = 2^{-1}$)
- ▶ If $\Delta t \neq 0$, require $h = h \lll 16$ (16-bit rotation invariant, $\text{Pr} = 2^{-16}$)[§]
- ▶ If $\Delta h \neq 0$, require $t = 0$ (no rotation, $\text{Pr} = 2^{-5}$)
- ▶ Search for good one-block collision differentials (future work: multi-block!)
- ▶ Use message modification (many things come for free in the beginning)

=

Collision Attack on Dynamic SHA

0: 0 - - -	16: ..1..1.. 0 - - -	32: 1..1..1.. 0 - - 0
1: 1 - - -	17: .1..1... 1 - - -	33: ..1.11.1 1 - G -
2:1... 1 R - -	18: 1..11... 1 - - 0	34: .1.11.1. 1 - G -
3: ...11... 1 - - -	19: ..111... 1 - G -	35: 1.1111.. 1 - - 0
4: ..111... 0 R - -	20: .111..1. 0 - - -	36: .111...1 0 - - -
5: .111.... 0 R - -	21: 111.11.. 0 - - 0	37: 111.1.1. 0 - G 0
6: 111.... 0 - - 0	22: 11.11..1 0 - G 0	38: 11.1.1.1 0 - G 0
7: 11....1 0 - G 0	23: 1.11..11 0 - G 0	39: 1.1.1.11 0 - G 0
8: 1....11 1 - - 0	24: .11.1111 1 - - -	40: .1.11111 1 - - -
9:1111 0 - G -	25: 11.1.11. 0 - G 0	41: 1.11.11. 0 - G 0
10: ...1.11. 0 R G -	26: 1.1.11.1 0 - G 0	42: .11..1.1 0 - G -
11: ..1..1.. 1 - - -	27: .1.1..11 1 - G -	43: 11....1. 1 - G 0
12: .1..... 0 R - -	28: 1.1.111. 0 - - 0	44: 1....1.1 0 - - 0
13: 1..... 1 - - 0	29: .1.1.1.1 1 - G -	45:11 1 - G -
14:1..1 0 - G -	30: 1.1...1. 0 - G 0	46:11. 0 - G -
15: ...1..1. 1 - G -	31: .1..11.1 1 - G -	47:1.. 1 R - -
		48:

- ▶ Same differential for both digest lengths
- ▶ Dynamic SHA-256: 2^{114} (incl. message modification)
- ▶ - 2^{170}

Preimage Attack

Preimage Attack

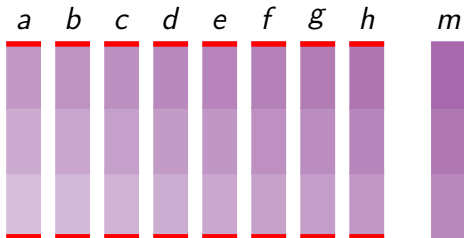
- ▶ Preimage attack on the compression function
- ▶ Trivial extension to second preimage attack on the hash function
- ▶ Idea somewhat similar to



Christophe De Cannière, Christian Rechberger
Preimages for Reduced SHA-0 and SHA-1
CRYPTO 2008

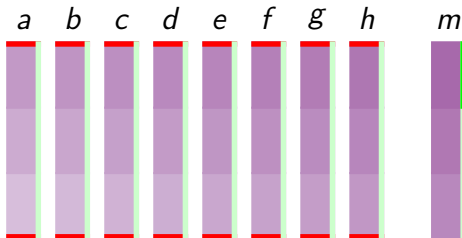
Idea

- ▶ Assume that **all rotations are by 0 bits**
- ▶ (there is enough freedom to do this)
- ▶ Now every bit slice depends only on less significant bitslices!



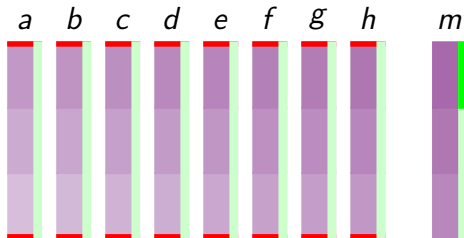
Idea

- ▶ Assume that **all rotations are by 0 bits**
- ▶ (there is enough freedom to do this)
- ▶ Now every bit slice depends only on less significant bitslices!



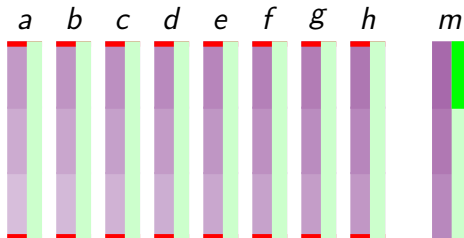
Idea

- ▶ Assume that **all rotations are by 0 bits**
- ▶ (there is enough freedom to do this)
- ▶ Now every bit slice depends only on less significant bitslices!



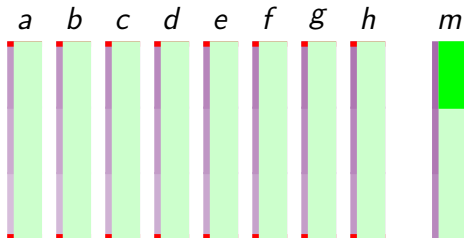
Idea

- ▶ Assume that **all rotations are by 0 bits**
- ▶ (there is enough freedom to do this)
- ▶ Now every bit slice depends only on less significant bitslices!



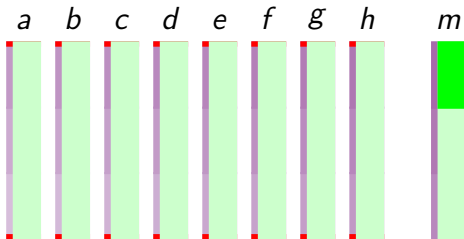
Idea

- ▶ Assume that **all rotations are by 0 bits**
- ▶ (there is enough freedom to do this)
- ▶ Now every bit slice depends only on less significant bitslices!



Idea

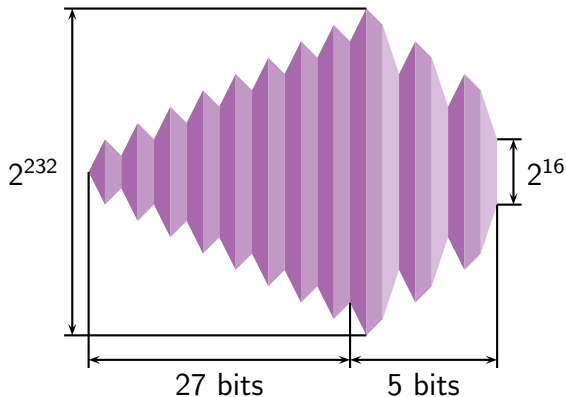
- ▶ Assume that **all rotations are by 0 bits**
- ▶ (there is enough freedom to do this)
- ▶ Now every bit slice depends only on less significant bitslices!



- ▶ 16
- ▶

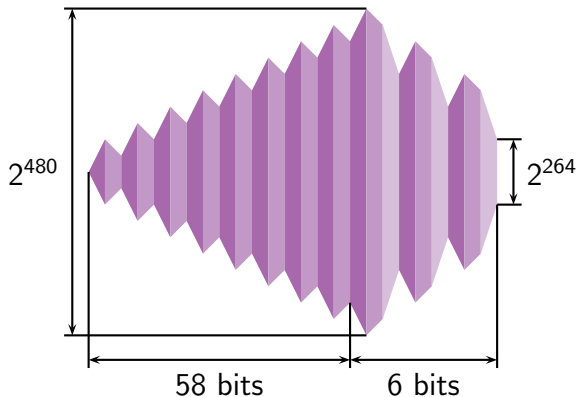
–8

Attack Complexity



$$- \quad 2^{27 \cdot 8 + 16} = 2^{216}$$

Attack Complexity



- ▶ - $2^{27 \cdot 8 + 16} = 2^{216}$
- ▶ - $\dots = 2^{256}$

-
-

Compression function preimage / Second preimage

- ▶ Dynamic SHA-256: 2^{216}
- ▶ Dynamic SHA-512: 2^{256}