Cryptanalysis of Dynamic SHA*

Sebastiaan Indesteege

COSIC, ESAT, Katholieke Universiteit Leuven, Belgium

First SHA-3 Candidate Conference
Rump Session

*Work in progress. Thanks to Jean-Philippe Aumasson and Orr Dunkelman for discussions and ideas.

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

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

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

$$T = \mathbf{R(a, b, c, d, e, f, g, h)};$$
$$h = g; \quad g = f; \quad f = e; \quad e = d;$$
$$d = \mathbf{G_{t \bmod 4}(a, b, c)} \boxplus W_{t \bmod 16} \boxplus TT_{\lfloor t/16 \rfloor}$$
$$c = b; \quad b = a; \quad a = T;$$

end for

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

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

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

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

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

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

return $\mathbf{h} \ggg \mathbf{t};$

end function

---

†For Dynamic SHA-256

Part I

Collision Attack

Observations on Dynamic SHA-256

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

$G(\cdot)$-functions

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

Observations on Dynamic SHA-256

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

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

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

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

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

return $\mathbf{h} \ggg \mathbf{t}$;

end function

R-function

- Linear in MSB of $a, \cdots, g$
- MSB of $a, \cdots, g$ only influences MSB of $t^{\ddagger}$

‡For Dynamic SHA-512, it influences $t^{(3)}$

Idea

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

[§] For Dynamic SHA-512, we require invariance under $8k$-bit rotation, so $\Pr = 2^{-56}$

Collision Attack on Dynamic SHA

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

- Same differential for both digest lengths
- Dynamic SHA-256: 2^{114} (incl. message modification)
- Dynamic SHA-512: 2^{170} (incl. message modification)

Part II

Preimage Attack

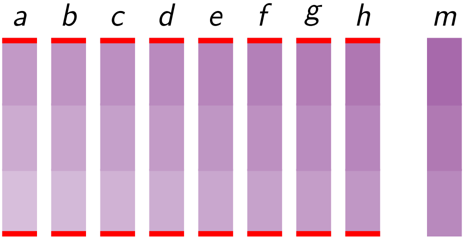
Preimage Attack

- 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

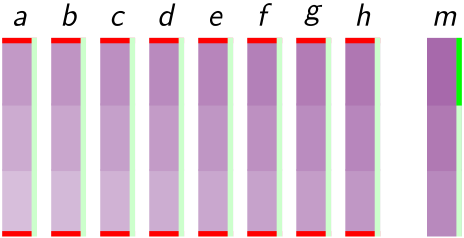
Idea

- 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!



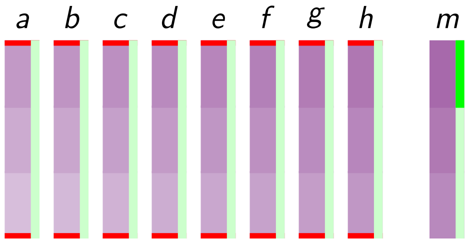
Idea

- 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!



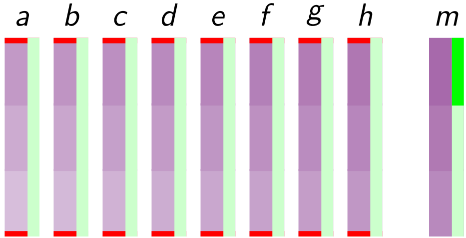
Idea

- 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!



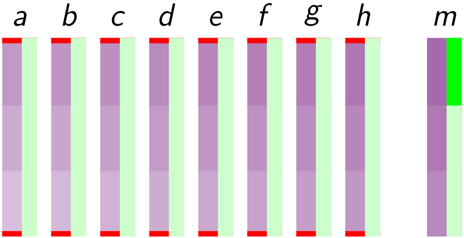
Idea

- 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!



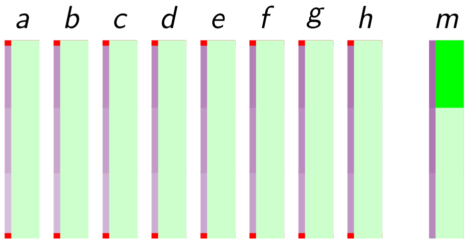
Idea

- 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!



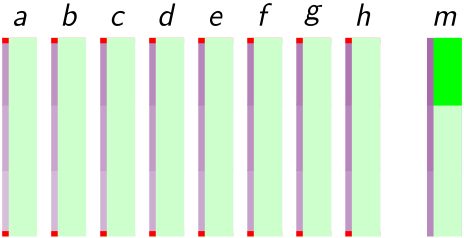
Idea

- 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!



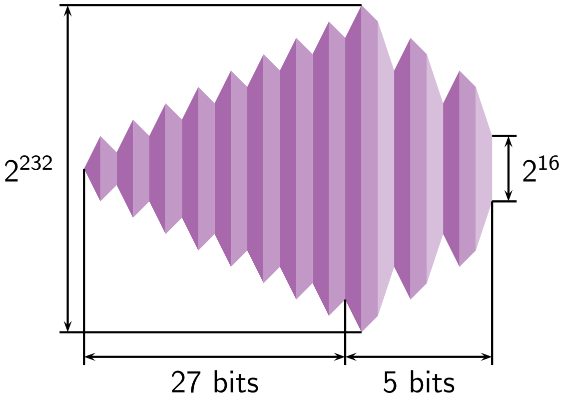
Idea

- 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!



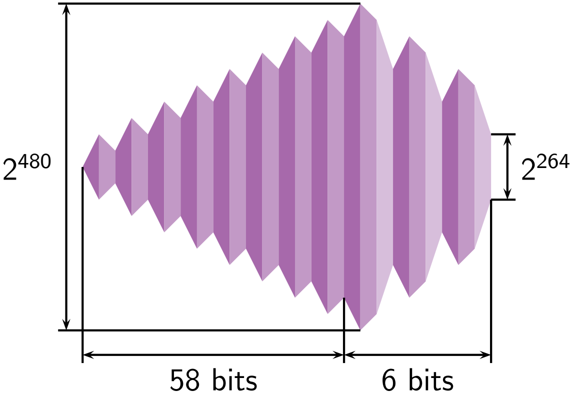
- 2^{16} freedom per bit slice; Pr 2^{-8} for match at output
- Compute 28 resp. 59 bit slices; then one bit of each t is known; filtering

Attack Complexity



- Dynamic SHA-256: $\frac{2^{27\cdot 8+16}}{2^{32\cdot 8-5\cdot 48}} = \mathbf{2^{216}}$

Attack Complexity



- Dynamic SHA-256: $\frac{2^{27\cdot 8+16}}{2^{32\cdot 8-5\cdot 48}} = \mathbf{2^{216}}$
- Dynamic SHA-512: $\frac{2^{58\cdot 8+16}}{2^{64\cdot 8-6\cdot 48}} = \mathbf{2^{256}}$

Conclusion

- Cryptanalysis of Dynamic SHA[¶]

Collision

- Dynamic SHA-256: 2^{114}
- Dynamic SHA-512: 2^{170}

Compression function preimage / Second preimage

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

[¶] Ongoing; work in progress