Practical Near-Collisions for Reduced Round Blake, Fugue, Hamsi and JH

Meltem Sönmez Turan, Erdener Uyan

Computer Security Division, National Institute of Standards and Technology, USA

Institute of Applied Mathematics, Middle East Technical University, Turkey

Near-Collisions

Near-Collision Resistance

A hash function H is near-collision resistant, if it is 'hard' to find two messages with hash values that differ in only small number of bits.

Compression Function h(M, CV) CV

An ϵ/n-bit near-collision on h is obtained whenever two message blocks M_1 and M_2 satisfying

$$weight(h(M_1, CV) \oplus h(M_2, CV)) = n - \epsilon \quad (1)$$

are found, where n is the output size.
$\epsilon = n$ corresponds to a collision on the compression function.

Complexity of Finding Near-Collisions

A Generic Approach to Find ϵ/n-bit Near-collisions

Given M_1 and M_2, finding a CV such that

$$weight(h(M_1, CV) \oplus h(M_2, CV)) = n - \epsilon \tag{2}$$

requires

- $\approx \sqrt{2^n / \binom{n}{\epsilon}}$ CF evaluations and $\sqrt{2^n / \binom{n}{\epsilon}}$ memory or
- $\approx 2^n / \binom{n}{\epsilon}$ CF evaluations

ϵ/n	Complexity ($\approx$)
151/256, 287/512, 553/1024	2^{10}
166/256, 308/512, 585/1024	2^{20}
176/256, 323/512, 606/1024	2^{30}
184/256, 335/512, 623/1024	2^{40}
191/256, 345/512, 638/1024	2^{50}
197/256, 354/512, 651/1024	2^{60}

Hill Climbing Methods

- Heuristic approach
- "Good" solutions to "hard" optimization problems in short running times

Failure in Cryptographic Problems

- Only one global optimal, no local optimal solutions
- Discontinuity of cryptographic functions

Hill Climbing Methods

The aim of our hill climbing method is to minimize the function

$$f_{M_1,M_2}(x) = weight(h(M_1, x) \oplus h(M_2, x)) \tag{3}$$

where $x \in \{0,1\}^n$, for given message blocks M_1 and M_2.
Let CV be a randomly chosen chaining value. We define the set of k-bit neighbors of CV as

$$S^k_{CV} = \{x \in \{0,1\}^n | weight(CV \oplus x) \leq k\}. \tag{4}$$

Clearly, the size of S^k_{CV} is equal to $\sum_{i=0}^{k} \binom{n}{i}$.
For message blocks M_1 and M_2, a chaining value CV is defined to be k-opt, if

$$f_{M_1,M_2}(CV) = \min_{x \in S^k_{CV}} f_{M_1,M_2}(x). \tag{5}$$

The Algorithm

Algorithm 2.1: HillClimbing(M_1, M_2, k)

Randomly select CV;
$f_{best} = f_{M_1,M_2}(CV)$;
while (CV is not k-opt)
    $CV = x$ such that $x \in S^k_{CV}$ with $f(x) < f_{best}$;
    $f_{best} = f_{M_1,M_2}(CV)$;
return (CV, f_{best})

Selection of the next chaining value

- Greedy Gradient ascent ✓
- Steepest ascent

Experimental Results

The algorithm is repeated approximately 2^{25} times and we consider the method successful, whenever we obtained an ϵ/n-bit near-collision with

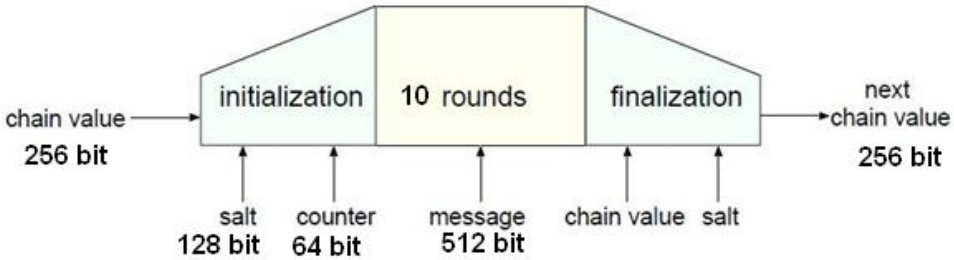
- $\epsilon \geq 184$ for $n = 256$ bits
- $\epsilon \geq 335$ for $n = 512$ bits
- $\epsilon \geq 623$ for $n = 1024$ bits

The generic approach requires 2^{40} compression function evaluations for these bounds.

Blake-32

Blake-32

- Designed by J.-P. Aumasson, L. Henzen, W. Meier, R. C.-W. Phan
- 10 rounds



Our setting

- 1 bit difference to the message blocks
- Counter and Salt are fixed to zero.

Comparison of Results

Paper	Rounds	Complexity	Type	Difference
✓	1	2^1	(256-4)/256-bit NC	Message
✓	1.5	$< 2^{26}$	(256-47)/256-bit NC	Message
✓	2	$< 2^{26}$	(256-72)/256-bit NC	Message
Su et al.	4 (4-7)	2^{21}	(256-104)/256-bit NC	Message, CV
✓	4	$2^{37.39}$	(256-74)/256-bit NC	Message
Aumasson et al.	4 (3-6)	2^{56}	(256-24)/256-bit NC	Message, CV, Salt, Counter

Fugue

Fugue

- Designed by S. Halevi and W. E. Hall and C. S. Jutla
- Sponge-like design, based on the F-256 function

Algorithm 3.1: $\text{F-256}(M_1, \ldots, M_m, IV_0, \ldots, IV_7, k, r, t)$

for $i \leftarrow 0$ **to** 21
  $S_i = 0$;
for $i \leftarrow 22$ **to** 29
  $S_i = IV_{i-22}$;
for $i \leftarrow 1$ **to** m
  $TIX(M_i)$;
  for $j \leftarrow 1$ **to** k
    $ROR3$; $CMIX$; $SMIX$;
for $i \leftarrow 1$ **to** r
  $ROR3$; $CMIX$; $SMIX$;
for $i \leftarrow 1$ **to** t
  $S_4 += S_0$; $S_{15} += S_0$; $ROR15$; $SMIX$;
  $S_4 += S_0$; $S_{16} += S_0$; $ROR14$; $SMIX$;
return $(S_1, S_2, S_3, S_4, S_{15}, S_{16}, S_{17}, S_{18}.)$

Fugue

Fugue

- Designed by S. Halevi and W. E. Hall and C. S. Jutla
- Sponge-like design, based on the F-256 function

Algorithm 3.2: $\text{F-256}(M_1, \ldots, M_m, IV_0, \ldots, IV_7, 2, 10, 13)$

for $i \leftarrow 0$ **to** 21
  $S_i = 0$;
for $i \leftarrow 22$ **to** 29
  $S_i = IV_{i-22}$;
for $i \leftarrow 1$ **to** m
  $TIX(M_i)$;
  for $j \leftarrow 1$ **to** 2
    $ROR3$; $CMIX$; $SMIX$;
for $i \leftarrow 1$ **to** 10
  $ROR3$; $CMIX$; $SMIX$;
for $i \leftarrow 1$ **to** 13
  $S_4 += S_0$; $S_{15} += S_0$; $ROR15$; $SMIX$;
  $S_4 += S_0$; $S_{16} += S_0$; $ROR14$; $SMIX$;
return $(S_1, S_2, S_3, S_4, S_{15}, S_{16}, S_{17}, S_{18}.)$

Result on the reduced version of F-256

(k, r, t)	Best result (with $\approx 2^{10} CF$)
(1,1,1),(1,1,2),(1,2,1),(1,2,2), (1,2,3),(1,2,4),(1,2,5),(2,1,1), (2,1,2),(2,1,3),(2,1,4),(2,1,5)	Collision
(1,1,3),(1,2,6),(1,3,1),(1,3,2), (1,3,3),(1,3,4),(1,3,5),(1,3,6), (1,3,7),(1,3,8),(2,1,6),(2,2,1), (2,2,2),(2,2,3),(2,2,4),(2,2,5), (2,2,6),(2,2,7),(2,2,8)	$\geq$ 231/256-bit near-collision
(1,1,4),(1,1,5),(1,2,7),(1,2,8), (1,3,9),(1,3,10),(2,1,7),(2,1,8), (2,2,9),(2,2,10)	$\geq$ 184/256-bit near-collision

TIX operation is parametrized by k, which is ignored in the implementation. Results should be updated.

Hamsi-256

Hamsi

- Designed by Ö. Küçük
- Input:256 bit CV, 32 bit Message block and Output:256 bit CV
- 3 Rounds

Our setting

- Random message blocks. After the linear expansion, $|\delta_M| \geq 70$ bits.
- No difference in CVs.

Comparison of Results

Paper by	Rounds	$\lvert\delta_{CV}\rvert$	$\lvert\delta_{M}\rvert$	Result
Nikolic	3	14	0	(256-25)/256-bit NC
Wang et al.	3	16	0	(256-23)/256-bit NC
Aumasson et al.	3	6	0	(256-25)/256-bit NC
Yun-qiang et al.	3	4	0	(256-20)/256-bit NC
✓	1	0	$\geq$ 70 bits	(256-24)/256-bit NC
✓	2	0	$\geq$ 70 bits	(256-64)/256-bit NC

JH

JH

- Designed by Wu
- Input:1024-bit CV, 512-bit Message block, Output:1024-bit CV
- 35.5 Rounds

Our setting

- 1-byte input difference to message blocks
- No difference in input CVs

JH Results

Rounds	Near-collision	Complexity
1	1023/1024	$2^{20.31}$
2	1020/1024	$2^{18.57}$
3	1019/1024	$2^{19.20}$
4	1013/1024	$2^{19.80}$
5	1005/1024	$2^{25.01}$
6	991/1024	$2^{27.57}$
7	942/1024	$2^{20.71}$
8	907/1024	$2^{24.24}$
9	816/1024	$2^{19.77}$
10	820/1024	$2^{23.24}$

Summary

We practically obtained

- 184/256-bit near-collision for the 2-round compression function of Blake-32
- 192/256-bit near-collision for the 2-round compression function of Hamsi-256
- 820/1024-bit near-collisions for 10-round compression function of JH

Thank you. Any Questions?