Boomerang and Slide-Rotational Analysis of the SM3 Hash Function

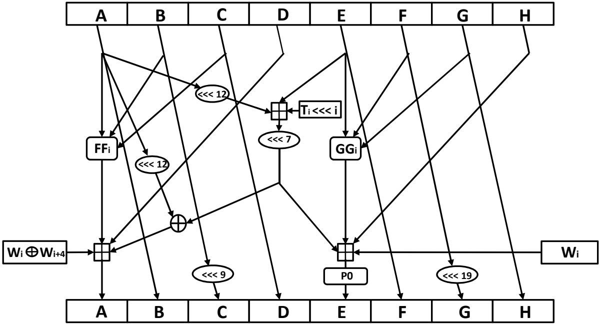
Aleksandar Kircanski[1], Yanzhao Shen[2],
Gaoli Wang[2†], Amr Youssef[1]

[1]: Concordia University, Montréal, Canada
[2]: Donghua University, Shanghai, China
[†]: State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China

Selected Areas in Cryptography (SAC) 2012
Windsor, Canada

August, 2012

Motivation



- SM3: a new hash function standardized in China
- Design: Xiaoyun Wang *et al.*
- Belongs to the SHA family

Overview

- SM3 hash specification
- A boomerang distnguisher for step-reduced SM3
- Slide-rotational property of SM3-XOR
- Future work and conclusions

SM3 hash: context

December 2007:

- Chinese National Cryptographic Administration Bureau releases a TCM
- To be used within the Trusted Computing framework in China
- Specified:
  - SMS4 block cipher
  - SM2 assymetric algorithm
  - SM3: a new cryptographic hash function

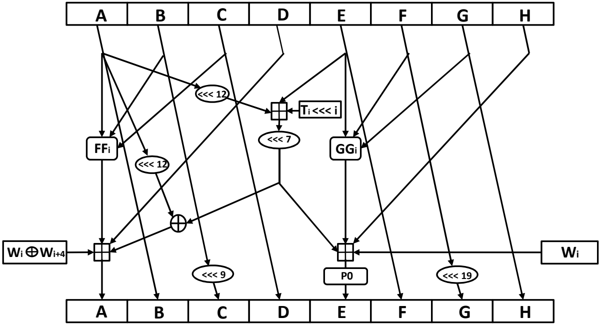
October 2011

- IETF RFC is published detailing SM3

SM3 hash: specification

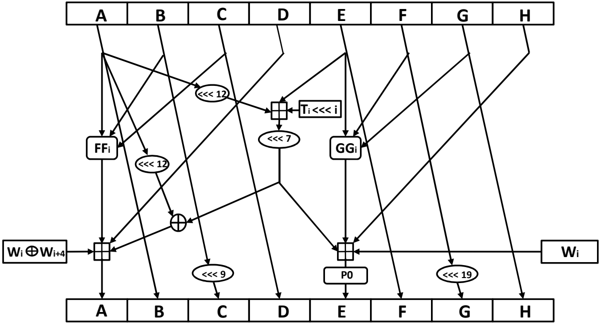
- Merkle-Damgård design
- 256-bit state and 512-bit message block are compressed to 256 bits.
- Belongs to the SHA family of hash functions (comparable to SHA-2).
- Compression function: 64 steps

Previous work: Zou *et al.*, ICISC 2011: Preimage for 30 step of SM3: computational complexity $\approx 2^{249}$ compression function calls, memory 2^{16}



Overview of the step function:

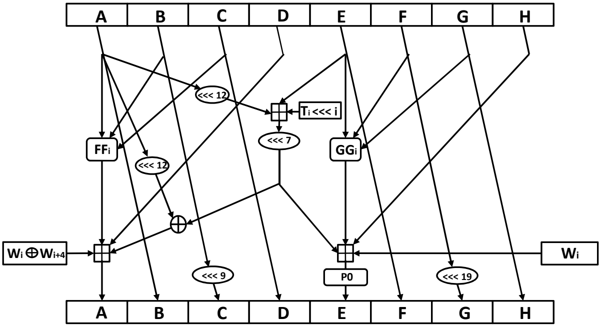
- Two words updated: A and E
- Operations: $+ \bmod 2^{32}$, $\oplus$, rotation, logical functions
- Two expanded message words fed to the step function



$$FF(X,Y,Z) = \begin{cases} X \oplus Y \oplus Z, & 0 \leq i \leq 15, \\ MAJ(X,Y,Z) & 16 \leq i \leq 63, \end{cases}$$

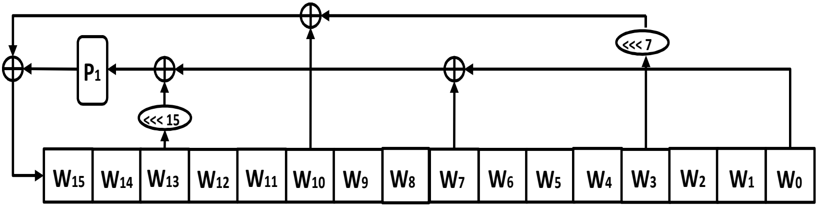
$$GG(X,Y,Z) = \begin{cases} X \oplus Y \oplus Z, & 0 \leq i \leq 15, \\ IF(X,Y,Z) & 16 \leq i \leq 63. \end{cases}$$

P_0 is defined as: $P_0(X) = X \oplus (X \lll 9) \oplus (X \lll 17)$



- Constant used in step i: $T_i \lll i$
- However, T_i is fixed in steps $j \in \{0, ..., 15\}$ and also in $j \in \{16, ..., 63\}$

Only two hard-coded constants used.



Operations: only $\oplus$, $\lll$. Maximal tap distance: 4. Here,

$$P_1(X) = X \oplus (X \lll 15) \oplus (X \lll 23).$$

The starting message $w_i = m_i$, $i = 0, \ldots 15$ is expanded to

$$w_i, i = 0, \ldots 67$$

and then

$$w'_i = w_i \oplus w_{i+4}, i = 0, \ldots 63$$

Comparison with SHA-2

- SM3: 2 instead of 1 message words are fed to the step function
- Maximal distances between taps in the message expansion, SM3: 4, SHA-2: 8
- In message expansion, SM3 uses only $+$ in F_2^{32} (whereas SHA-2 uses $+$ both in $Z_{2^{32}}$ and F_2^{32})
- SM3 step function: 8 mod 2^{32} additions, as opposed to 7 such additions in the case of SHA-2.

Boomerang distinguishers for hash functions

Goal: distinguish the compression function from a random function.

Definition: zero-sum

A 4-zero-sum for f is a quartet x_0, x_1, x_2, x_3 s.t.

$$
\begin{aligned}
x_0 \oplus x_1 \oplus x_2 \oplus x_3 &= 0 \\
f(x_0) \oplus f(x_1) \oplus f(x_2) \oplus f(x_3) &= 0
\end{aligned}
$$

- Used to distinguish Keccak-f permutation (Aumasson, Meier) CHES 2009
- Goal: find $\{x_0, x_1, x_2, x_3\}$ faster than generically

Best known generic algorithm: $2^{n/2}$, n is the f output size
Query complexity: $2^{n/3}$

Boomerang distinguishers for hash functions

Using boomerang attack to generate zero-sums was proposed in 2011 independently by:

- Biryukov and Nikolić in the context of BLAKE (2011)
- Mendel and Lamberger in the context of SHA-256 (2011)

Zero-sums can be seen as second-order collisions.

Definition

A second-order collision for f is a pair (a_1, a_2) together with x such that

$$f(x \oplus a_1 \oplus a_2) \oplus f(x \oplus a_1) \oplus f(x \oplus a_2) \oplus f(x) = 0$$

Definition: zero-sum

Def: A 4-zero-sum for is a quartet x_0, x_1, x_2, x_3 s.t.

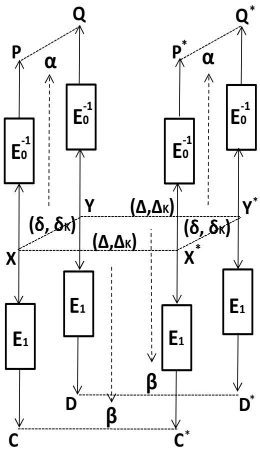
$$x_0 \oplus x_1 \oplus x_2 \oplus x_3 = 0$$
$$f(x_0) \oplus f(x_1) \oplus f(x_2) \oplus f(x_3) = 0$$

Definition: second-order collision

A second-order collision for f is a pair (a_1, a_2) together with x such that

$$f(x \oplus a_1 \oplus a_2) \oplus f(x \oplus a_1) \oplus f(x \oplus a_2) \oplus f(x) = 0$$

Equivalent notions, e.g., set $x = x_0$, $a_1 = x_0 \oplus x_1$, $a_2 = x_0 \oplus x_2$.

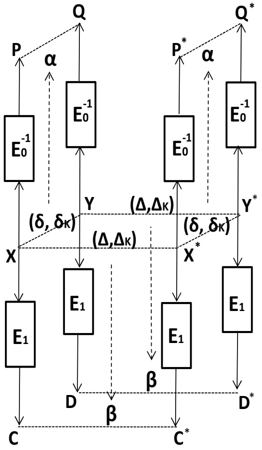


Start-from-the-middle boomerang approach:

- Represent E as $E = E_1 \circ E_0$
- Fix related-key differentials

$$(\delta, \delta_K) \to \alpha$$
$$(\Delta, \Delta_K) \to \beta$$

  for E_0^{-1} and E_1.
- Set up:
  - a quartet of keys/messages
  - a quartet of middle states
- Starting from (X, X^*, Y, Y^*)
- Compute backward: obtain (P, P^*, Q, Q^*)
- Compute forward: obtain (C, C^*, D, D^*)



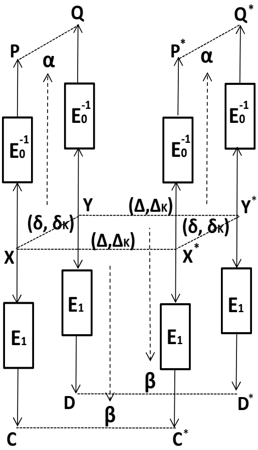
- Verify whether

$$C \oplus C^* = D \oplus D^*$$
$$P \oplus Q = P^* \oplus Q^*.$$

- If yes, a zero-sum for the encryption in Davis Meyer mode is found:

$$P \oplus Q \oplus P^* \oplus Q^* = 0$$
$$(C \oplus P) \oplus (C^* \oplus P^*) \oplus$$
$$(D \oplus Q) \oplus (D^* \oplus Q^*) = 0$$

If a zero-sum can be found in less than $2^{n/2}$, the compression function can be distinguished from random.



Two main steps:

(1) Get a zero-sum property for the middle steps

(2) Add steps at the top and the bottom

(1) and (2) can sometimes be done independently.

- Step (1)
  - Use message modification to find one zero-sum for middle steps
  - Augment the result using auxiliary differentials (Leurent and Roy, CT-RSA 2012)
- Step (2): satisfy randomly

33-step boomerang distinguisher

- The backward direction from step 16 to step 1 holds with probability 2^{-69}
- The forward direction from step 16 to step 33 holds with probability 2^{-70}
- Previously set A_{16} to H_{16}, 33-step boomerang distinguisher holds with probability 2^{-82}
- Using the message modification, 33-step boomerang distinguisher holds with probability 2^{-41}
- Using the amplified differential characteristics, 33-step boomerang distinguisher holds with probability $2^{-32.4}$

A 33-step SM3 zero-sum example

Message								
M_X	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
$M_{X'}$	00000000	00000000	80000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
M_Y	04001c00	02800080	08582838	5000a050	80858283	00008000	68000800	00000800
	00000000	7010b050	08080010	00000000	00008000	28000800	00000000	00000000
$M_{Y'}$	04001c00	02800080	88582838	5000a050	80858283	00008000	68000800	00000800
	00000000	7010b050	08080010	00000000	00008000	28000800	00000000	00000000
Chaining Value								
IV_X	274e6355	3333edb0	14f1b3d9	7be58154	d969d138	bb60c21a	ff5909df	e92dce5d
$IV_{X'}$	274e6355	3373edb0	94f1b3d9	fba58154	d969d138	bb60d21a	7f5909df	692dde5d
IV_Y	28b7b4d8	fe5f1155	93973138	c10d3808	32d4319b	dc8de94e	ef594319	8ef80fe1
$IV_{Y'}$	28b7b4d8	fe1f1155	13973138	414d3808	32d4319b	dc8df94e	6f594319	0ef81fe1
H_X	52793642	8017615c	fbf548ba	8b05cf67	dcb79a73	e1035e10	2caefeae	701d22d9
$H_{X'}$	772427a1	b2064c80	0dd79a89	2a809122	8bc2413f	8dd6b954	bad8867b	06c59c18
H_Y	987f3286	c017e19c	fbf548ba	8b05cf67	dabd9677	e1035e10	2caefeae	701d22d9
$H_{Y'}$	bd222365	f206cc40	0dd79a89	2a809122	8dc84d3b	8dd6b954	bad8867b	06c59c18

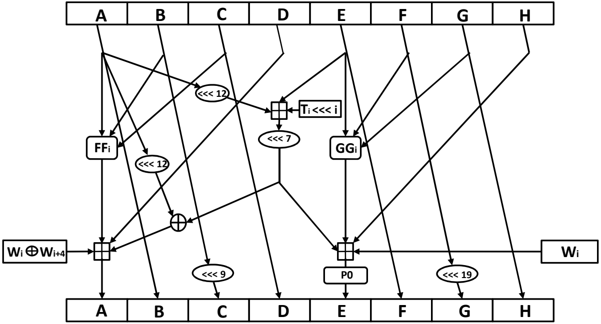
34/35-step boomerang distinguisher

- Add 1 step after 33-step, we can get a 34-step boomerang distinguisher with probability $2^{-(32.4+20.7)} = 2^{-53.1}$
- Add 2 steps after 33-step, we can get a 35-step boomerang distinguisher with probability $2^{-(32.4+20.7+2\times 32)} = 2^{-117.1}$

Comparison to the SHA-256 boomerang distinguisher

- A similar method for SHA-256: 47 steps (Asiacrypt 2011)
- SM3 allows passing less steps mainly due to:
  - Maximal distance between taps in the message exp., SM3: 4, SHA-2: 8
  - SM3: Two messages on distance 4 fed to the registers in each step in SM3

Slide-Rotational Property of SM3-XOR



Constants used in step i:

- $i \in \{0, ..., 15\}$: $0x79cc4519 \lll i$
- $i \in \{16, ..., 63\}$: $0x7a879d8a \lll i$,

Does this introduce some "regularity" in the SM3 compression function?

Observation 1

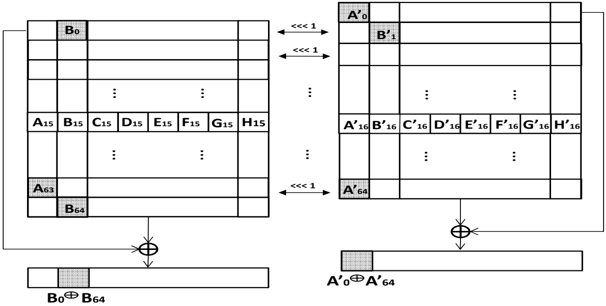
Constants used in steps j and $j+1$ are *rotational*, for all steps except for step $j = 15$.

Observation 2

All the operations except modular addition in the SM3 step function preserve rotational property with probability 1.

Instead of SM3, we look into SM3-XOR:

- addition mod 2^{32} replaced by $\oplus$
- FF_i and GG_i are left as is.



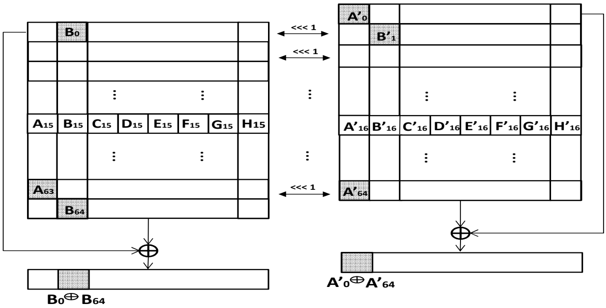
Since constants used in steps j and $j+1$ are rotational, it makes sense to introduce *sliding*. Setup a slide-rotational pair of messages (w, w^*)

$$w^*_{i+1} = w_i \lll 1, w'^*_{i+1} = w'_i \lll 1$$

Also, a slide-rotational pair of registers $(A, \ldots, H), (A^*, \ldots, H^*)$:

$$A^*_{i+1} = A_i \lll 1, B^*_{i+1} = B_i \lll 1, \ldots, H^*_{i+1} = H_i \lll 1 \qquad (1)$$

For every $i \neq 15$, (1) will be preserved for $i+1$ with probability 1.



In steps $i = 0, ..14$ and $16, \ldots 62$, the rotational property is satisfied with probability 1.

To bypass the middle step problem, one starts from step 15, constructs a rotational pair for this step and then propagates forward and backward.

Result

Instant generation of "rotational" input-output pairs for SM3-XOR.

$A^1, B^1, \ldots, H^1$	0x565060b7 0x125d5655 0x285c7653 0xeaf5fe1e 0xda8bd7dd 0xb8bb1904 0x43bcaf18 0x7cf88895
$W^1_0, \ldots, W^1_{15}$	0x8f450bbd 0x4a0c9922 0x73dd44f8 0x9eceaaf8 0x33b13e20 0xb59d9c33 0x6b5a5f23 0xc0d2b468 0x7a9a1e16 0xaff62878 0x3fbb01f4 0x75278787 0xac0b849e 0x498f3045 0x62687c15 0xd3498eb
$A^2, B^2, \ldots, H^2$	0x24baacaa 0x53285c76 0xd5ebfc3d 0xdf1ee2a6 0x71763209 0x2bc610ef 0xf9f1112a 0xffeb86a4
$W^2_0, \ldots, W^2_{15}$	0x7efa7542 0x1e8a177b 0x94193244 0xe7ba89f0 0x3d9d55f1 0x67627c40 0x6b3b3867 0xd6b4be46 0x81a568d1 0xf5343c2c 0x5fec50f1 0x7f7603e8 0xea4f0f0e 0x5817093d 0x931e608a 0xc4d0f82a

Figure: SM3-XOR slide-rotational pair example

If instead of SM3-XOR, the SM3 compression function is considered:

- a probabilistic slide-rotational property
- one step preserves the rotational property with $\approx (p_1)^8 = 2^{-11.320}$.

Similar property does not exist for the SHA-2-XOR

Yoshida *et al.*, SAC 2005: 31-step SHA-2-XOR was shown to exhibit non-randomness $\Rightarrow$ attack on 32-step SHACAL-2-XOR)

Conclusions

- SM3 appears to be more resistant to boomerang distinguishers than SHA-2
- Unlike SHA2-XOR, SM3-XOR admits a simple slide-rotational property
- No real impact on the SM3 security

Future work

- Extend the boomerang distinguisher to more steps by adding steps in the middle
- Explore the slide-rotational property present in SM3

Thank you