Boomerang Attacks against ARX Hash Functions

Gaëtan Leurent & Arnab Roy

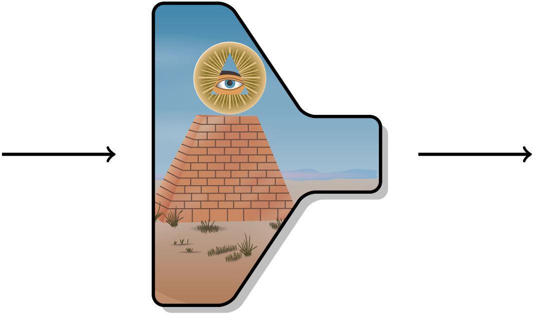
Gaëtan Leurent
University of Luxembourg

Session ID: CRYP-301
Session Classification: Advanced

RSACONFERENCE2012

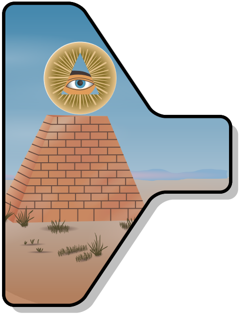
Introduction to Hash Functions

An Ideal Hash Function: the Random Oracle



- Public Random Oracle
- The output can be used as a fingerprint of the document

An Ideal Hash Function: the Random Oracle

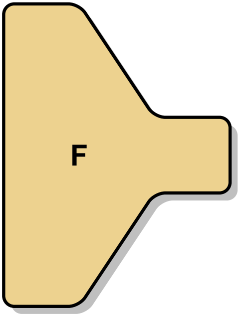


0x1d66ca77ab361c6f

- Public Random Oracle
- The output can be used as a fingerprint of the document

A Concrete Hash Function

- A public function with no structural property.
  - Should behave like a random function.
  - Cryptographic strength without any key!
- $F : \{0,1\}^* \to \{0,1\}^n$

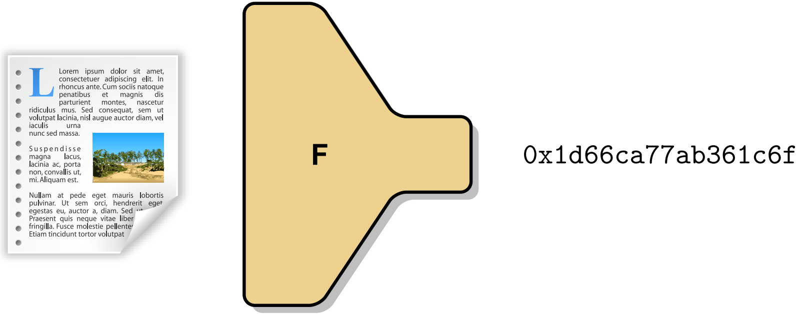


0x1d66ca77ab361c6f

A Concrete Hash Function

- A public function with no structural property.
  - Should behave like a random function.
  - Cryptographic strength without any key!

- $F : \{0,1\}^* \to \{0,1\}^n$



Using Hash Functions

Hash functions are used in many different contexts:

- To generate unique identifiers
  - Hash-and-sign signatures
  - Commitment schemes
- As a one-way function
  - One-Time-Passwords
  - Forward security
- To break the structure of the input
  - Entropy extractors
  - Key derivation
  - Pseudo-random number generator
- To build MACs
  - HMAC
  - Challenge/response authentication

The SHA-3 Competition

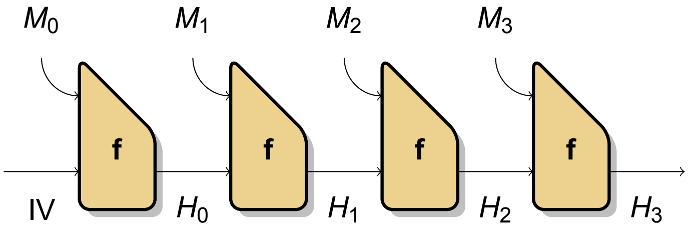
After Wang *et al.*'s attacks on the MD/SHA family,
we need new hash functions

The SHA-3 competition

- Organized by NIST
- Similar to the AES competition
- Submission deadline was October 2008: 64 candidiates
- 51 valid submissions
- 14 in the second round (July 2009)
- 5 finalists in December 2010:
  - Blake, Grøstl, JH, Keccak, Skein
- Winner in 2012?

Hash Function Design

- Build a small compression function, and iterate.
  - Cut the message in chunks $M_0, ...M_k$
  - $H_i = f(M_i, H_{-1})$
  - $F(M) = H_k$

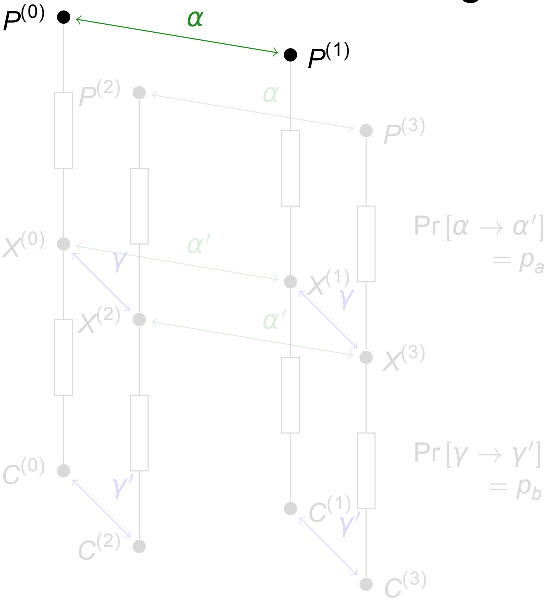


Boomerang Attacks

Boomerang Attacks

- Introduced by Wagner, many later improvements
- Combine two short differentials instead of using a long one.
  - $f = f_b \circ f_a$
  - for f_a, $\alpha \to \alpha'$ with probability p_a
  - for f_b, $\gamma \to \gamma'$ with probability p_b
  - Interesting when we don't know how to build iterative differentials.
- Uses an encryption oracle together with a decryption oracle
  - Adaptive attack

Boomerang Attacks

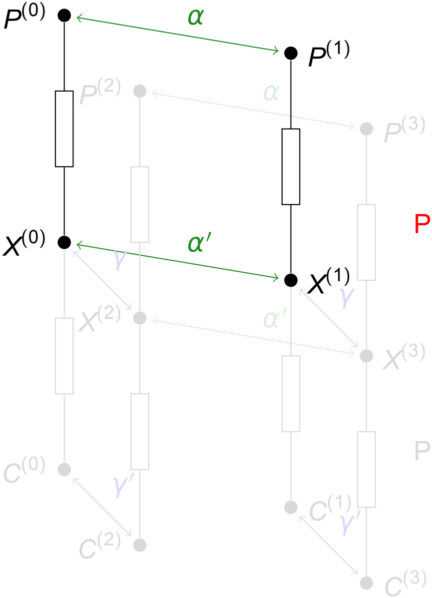


1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

$$C = \frac{1}{p_a}\frac{1}{p_b^2}\frac{1}{p_a}$$

$$P^{(0)} \oplus P^{(1)} = \alpha$$
$$P^{(2)} \oplus P^{(3)} = \alpha$$
$$C^{(0)} \oplus C^{(1)} = \gamma'$$
$$C^{(2)} \oplus C^{(3)} = \gamma'$$

Boomerang Attacks



$\Pr[\alpha \to \alpha'] = p_a$

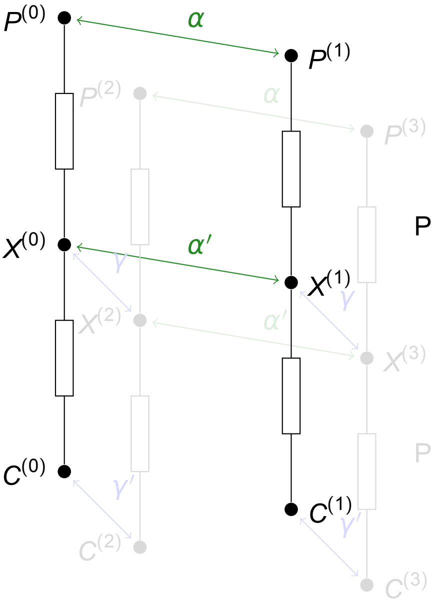
$\Pr[\gamma \to \gamma'] = p_b$

1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

$$C = \frac{1}{p_a} \frac{1}{p_b^2} \frac{1}{p_a}$$

$$P^{(0)} \oplus P^{(1)} = \alpha$$
$$P^{(2)} \oplus P^{(3)} = \alpha$$
$$C^{(0)} \oplus C^{(1)} = \gamma'$$
$$C^{(2)} \oplus C^{(3)} = \gamma'$$

Boomerang Attacks



1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

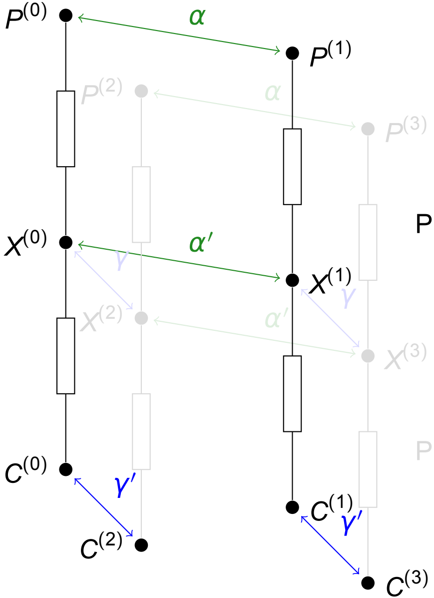
$$\Pr[\alpha \to \alpha'] = p_a$$

$$C = \frac{1}{p_a} \frac{1}{p_b^2} \frac{1}{p_a}$$

$$\Pr[\gamma \to \gamma'] = p_b$$

$$P^{(0)} \oplus P^{(1)} = \alpha$$
$$P^{(2)} \oplus P^{(3)} = \alpha$$
$$C^{(0)} \oplus C^{(1)} = \gamma'$$
$$C^{(2)} \oplus C^{(3)} = \gamma'$$

Boomerang Attacks



1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

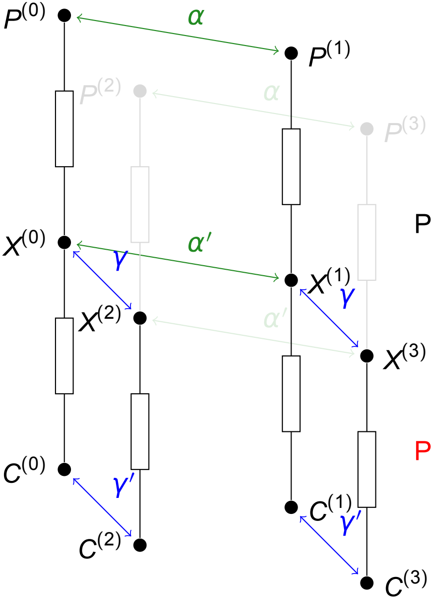
$$\Pr[\alpha \to \alpha'] = p_a$$

$$C = \frac{1}{p_a} \frac{1}{p_b^2} \frac{1}{p_a}$$

$$\Pr[\gamma \to \gamma'] = p_b$$

$$P^{(0)} \oplus P^{(1)} = \alpha$$
$$P^{(2)} \oplus P^{(3)} = \alpha$$
$$C^{(0)} \oplus C^{(1)} = \gamma'$$
$$C^{(2)} \oplus C^{(3)} = \gamma'$$

Boomerang Attacks



$\Pr[\alpha \to \alpha'] = p_a$

$\Pr[\gamma \to \gamma'] = p_b$

1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

$$C = \frac{1}{p_a} \frac{1}{p_b^2} \frac{1}{p_a}$$

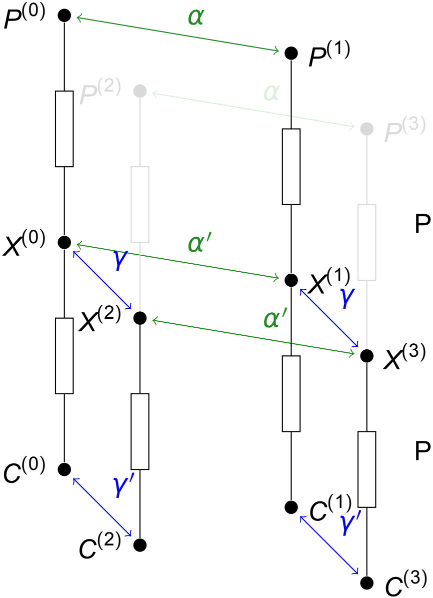
$P^{(0)} \oplus P^{(1)} = \alpha$

$P^{(2)} \oplus P^{(3)} = \alpha$

$C^{(0)} \oplus C^{(1)} = \gamma'$

$C^{(2)} \oplus C^{(3)} = \gamma'$

Boomerang Attacks



$\Pr[\alpha \to \alpha'] = p_a$

$\Pr[\gamma \to \gamma'] = p_b$

1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

$$C = \frac{1}{p_a} \frac{1}{p_b^2} \frac{1}{p_a}$$

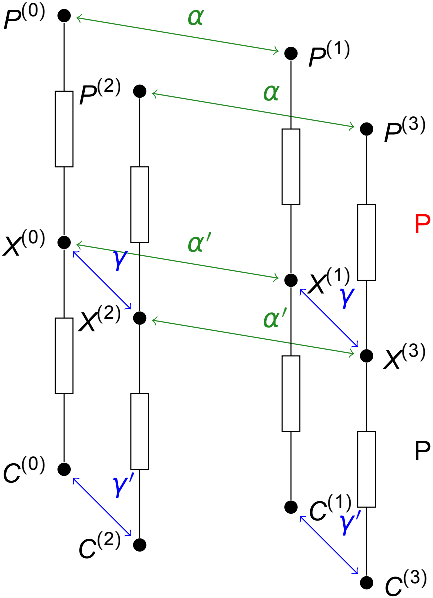
$P^{(0)} \oplus P^{(1)} = \alpha$

$P^{(2)} \oplus P^{(3)} = \alpha$

$C^{(0)} \oplus C^{(1)} = \gamma'$

$C^{(2)} \oplus C^{(3)} = \gamma'$

Boomerang Attacks



$\Pr[\alpha \to \alpha'] = p_a$

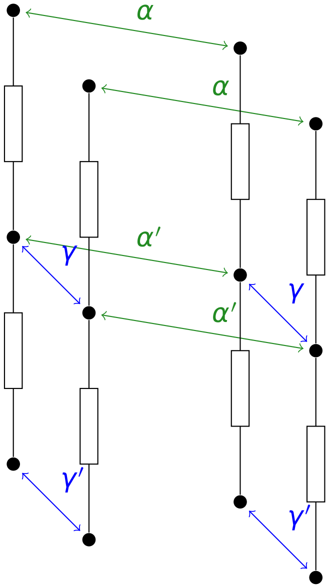
$\Pr[\gamma \to \gamma'] = p_b$

1. Start with $P^{(0)}, P^{(1)}$
2. Compute $C^{(0)}, C^{(1)}$
3. Build $C^{(2)}, C^{(3)}$
4. Compute $P^{(2)}, P^{(3)}$

$$C = \frac{1}{p_a} \frac{1}{p_b^2} \frac{1}{p_a}$$

$$P^{(0)} \oplus P^{(1)} = \alpha$$
$$P^{(2)} \oplus P^{(3)} = \alpha$$
$$C^{(0)} \oplus C^{(1)} = \gamma'$$
$$C^{(2)} \oplus C^{(3)} = \gamma'$$

Improvements to the Boomerang Attack



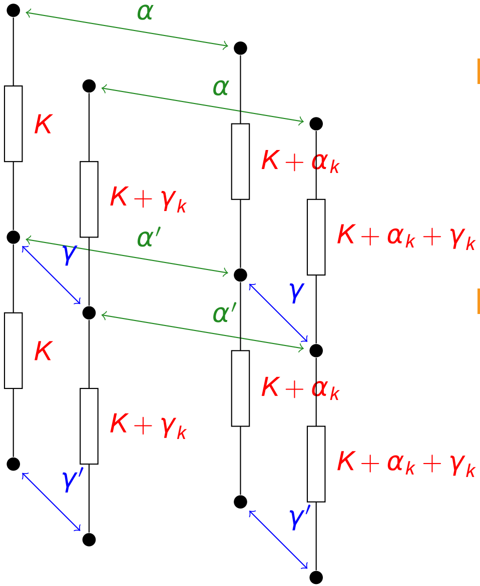
1. Amplified probabilities
   - Do not specify α' and γ
   - $\hat{p}_a = \sqrt{\sum_{\alpha'} \Pr[\alpha \to \alpha']}$

     $\hat{p}_b = \sqrt{\sum_{\gamma} \Pr[\gamma \to \gamma']}$

2. Related-key
   - $p_a = \Pr\left[\alpha \xrightarrow{\alpha_k} \alpha'\right]$

     $p_b = \Pr\left[\gamma \xrightarrow{\gamma_k} \gamma'\right]$

Improvements to the Boomerang Attack



1. Amplified probabilities
   - Do not specify α' and γ
   - $\hat{p}_a = \sqrt{\sum_{\alpha'} \Pr[\alpha \to \alpha']}$
     $\hat{p}_b = \sqrt{\sum_{\gamma} \Pr[\gamma \to \gamma']}$

2. Related-key
   - $p_a = \Pr\left[\alpha \xrightarrow{\alpha_k} \alpha'\right]$
     $p_b = \Pr\left[\gamma \xrightarrow{\gamma_k} \gamma'\right]$

Boomerang Attacks on Hash Functions

- Most hash functions are based on a block cipher:

  Davies-Meyer $f(h, m) = E_m(h) \oplus h$

  Matyas-Meyer-Oseas $f(h, m) = E_h(m) \oplus m$

- A (related-key) boomerang attack gives a quartet:

$$\sum P^{(i)} = 0 \qquad \sum C^{(i)} = 0 \qquad \sum K^{(i)} = 0$$

- This is a zero-sum for the compression function:

$$\sum h^{(i)} = 0 \qquad \sum m^{(i)} = 0 \qquad \sum f(h^{(i)}, m^{(i)}) = 0$$

- In general this is hard:
  - $\sum f(h, m) = 0$, best attack $2^{n/3}$, lower bound $2^{n/4}$
  - $\sum f(h, m) = \sum h = \sum m = 0$, best attack $2^{n/2}$, lower bound $2^{n/3}$

- With a known key, one can start from the middle
  - Message modification

New Technique:
Better Use of Degrees of Freedom
in a Hash Function Setting.

Using Auxiliary Paths

- Divide f in three sub-functions: $f = f_c \circ f_b \circ f_a$
  - for f_a, $\alpha \to \alpha'$ with probability p_a
  - for f_b, $\beta_j \to \beta_j'$ with probability p_b
  - for f_c, $\gamma \to \gamma'$ with probability p_c

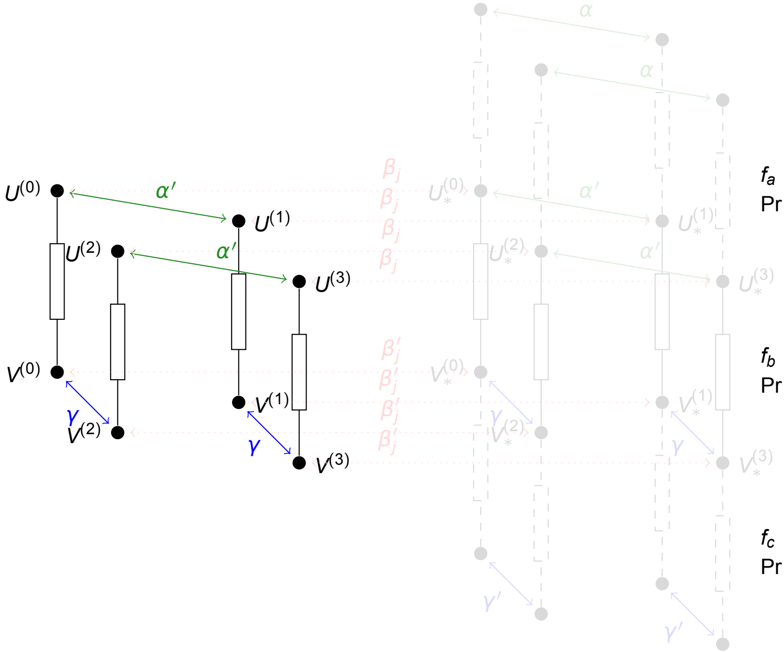
1. Start with a boomerang quartet for f_b:

$$U^{(1)} = U^{(0)} + \alpha' \qquad U^{(3)} = U^{(2)} + \alpha'$$
$$V^{(2)} = V^{(0)} + \gamma \qquad V^{(2)} = V^{(1)} + \gamma$$

2. For each auxiliary path, construct $U_*^{(i)} = U^{(i)} + \beta_j$.
   With probability p_b^4, $V_*^{(i)} = V^{(i)} + \beta_j'$, and we have a new quartet:

$$U_*^{(1)} = U_*^{(0)} + \alpha' \qquad U_*^{(3)} = U_*^{(2)} + \alpha'$$
$$V_*^{(2)} = V_*^{(0)} + \gamma \qquad V_*^{(2)} = V_*^{(1)} + \gamma$$

3. Check if the f_a and f_b paths are satisfied.



f_a

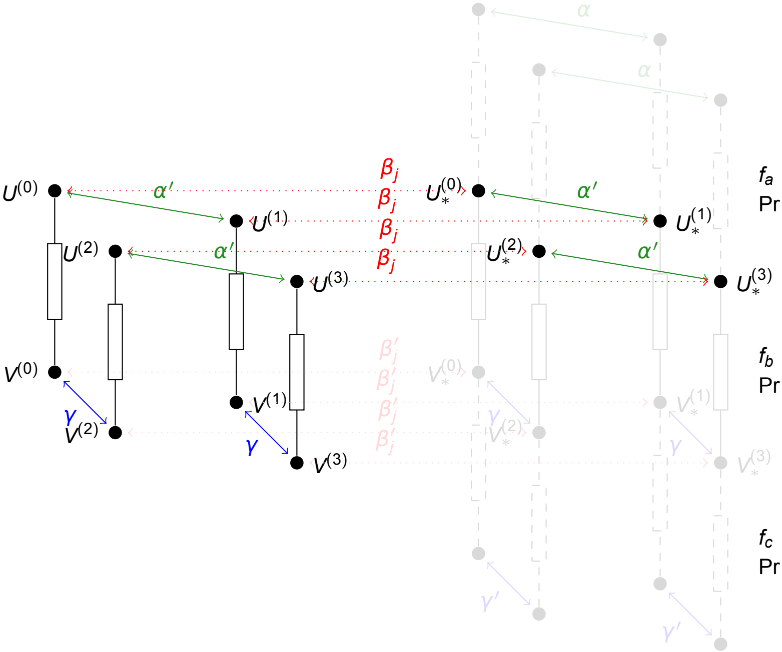
$\Pr[\alpha \to \alpha'] = p_a$

f_b

$\Pr\left[\beta_j \to \beta'_j\right] = p_b$

f_c

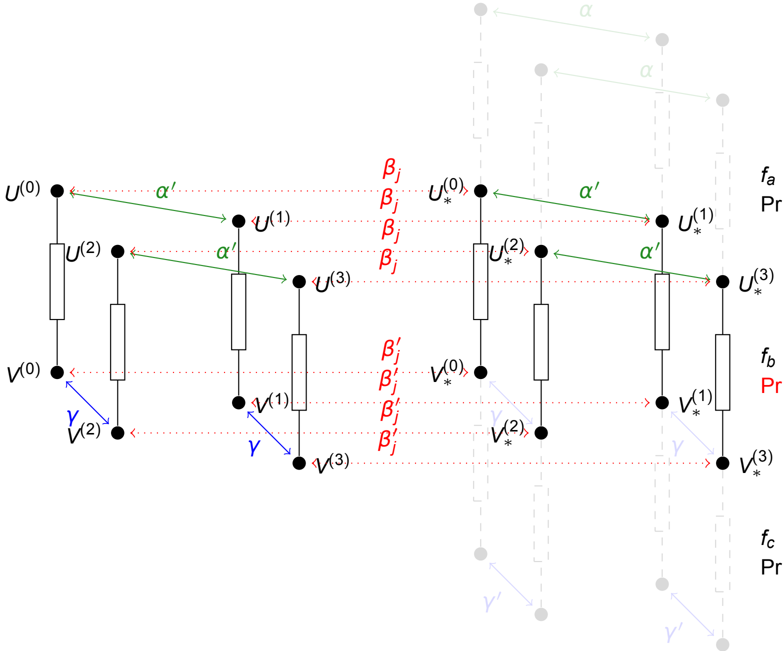
$\Pr[\gamma \to \gamma'] = p_c$



f_a
$\Pr[\alpha \to \alpha'] = p_a$

f_b
$\Pr\left[\beta_j \to \beta'_j\right] = p_b$

f_c
$\Pr[\gamma \to \gamma'] = p_c$



f_a

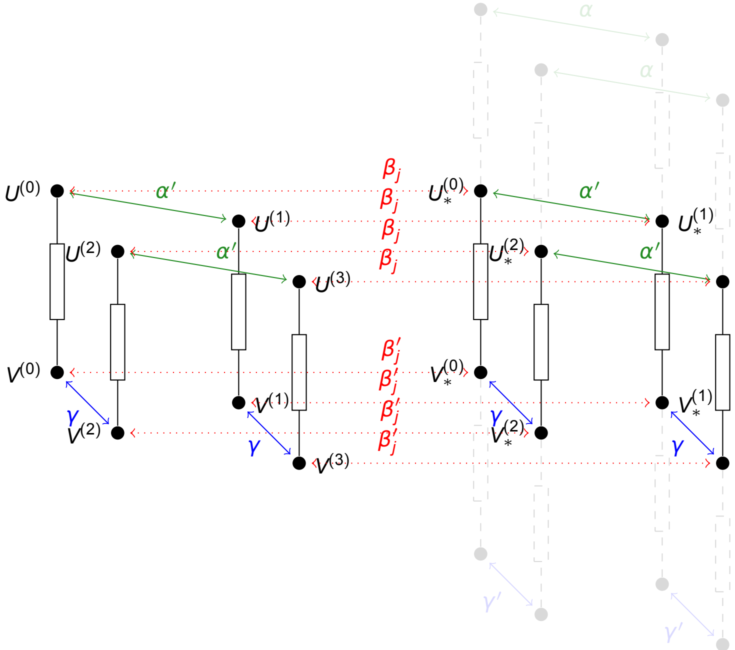
$\Pr[\alpha \to \alpha'] = p_a$

f_b

$\Pr\left[\beta_j \to \beta'_j\right] = p_b$

f_c

$\Pr[\gamma \to \gamma'] = p_c$



f_a

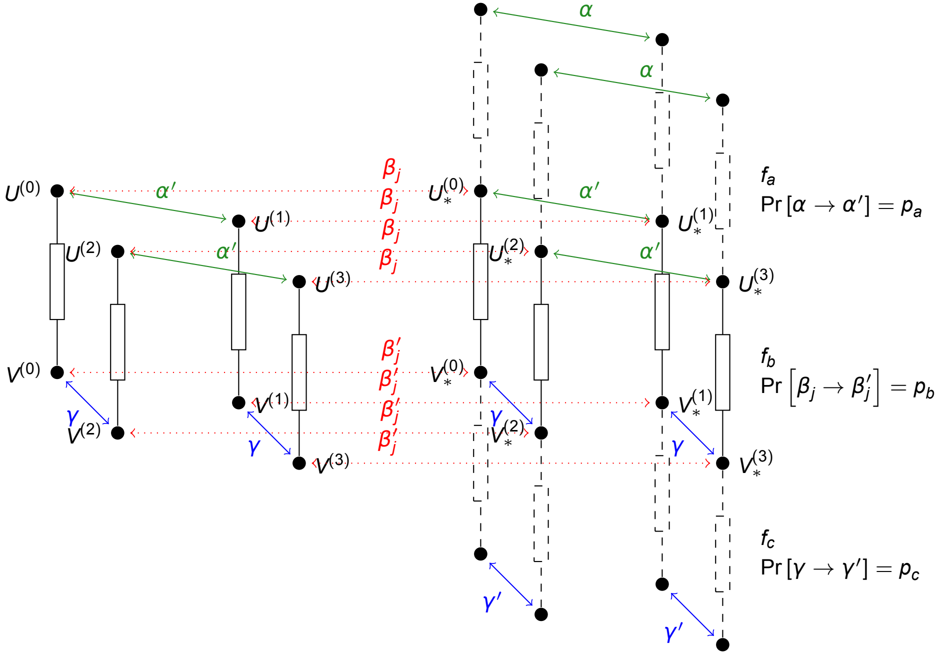
$\Pr[\alpha \to \alpha'] = p_a$

f_b

$\Pr\left[\beta_j \to \beta_j'\right] = p_b$

f_c

$\Pr[\gamma \to \gamma'] = p_c$



α
α
$U^{(0)}$
$U^{(1)}$
$U^{(2)}$
$U^{(3)}$
$U_*^{(0)}$
$U_*^{(1)}$
$U_*^{(2)}$
$U_*^{(3)}$
α′
β_j
β'_j
$V^{(0)}$
$V^{(1)}$
$V^{(2)}$
$V^{(3)}$
$V_*^{(0)}$
$V_*^{(1)}$
$V_*^{(2)}$
$V_*^{(3)}$
γ
γ′
f_a
$\Pr[\alpha \to \alpha'] = p_a$
f_b
$\Pr\left[\beta_j \to \beta'_j\right] = p_b$
f_c
$\Pr[\gamma \to \gamma'] = p_c$

Using Auxiliary Paths

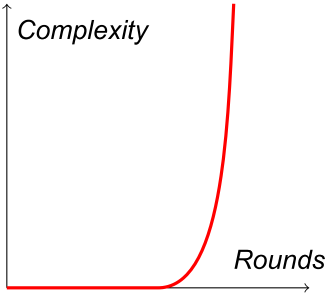
- Hash function setting allows to start from the middle and to build related quartets (instead of related pairs)
- Complexity: $$\frac{1}{p_a^2 p_c^2}\left(\frac{C}{b \cdot p_b^4} + 1\right)$$
  - Cost C to build an initial quartet
  - b paths with probability p_b for f_b
- Also works with related-key paths
  - New quartet with a different key
- Very efficient with a large family of probability 1 paths
  - We can combine three paths instead of two

Application

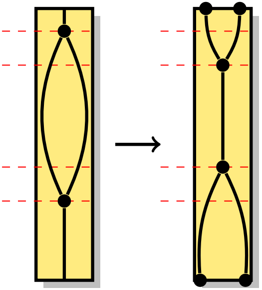
Application to ARX Designs

- Several recent design are based on the ARX design
  - Use only Addition, Rotation, Xor
  - Skein, Blake are SHA-3 finalists

- Short RK paths with high probability



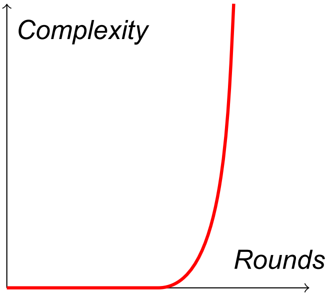
- Hard to build controlled characteristics



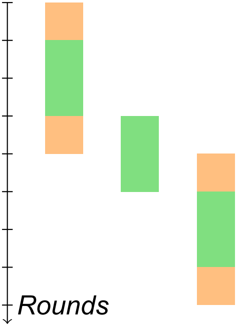
Application to ARX Designs

- Several recent design are based on the ARX design
  - Use only Addition, Rotation, Xor
  - Skein, Blake are SHA-3 finalists

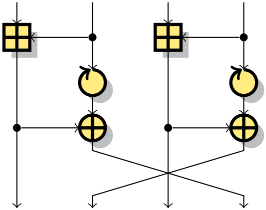
- Short RK paths with high probability



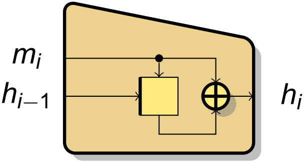
- Using auxiliary paths



Skein



Threefish-256 round



MMO mode

- SHA-3 finalist
- ARX design
  - 64-bit words
  - $\texttt{MIX}_r(a, b) := ((a \boxplus b), (b \lll r) \oplus c)$
  - Word permutations
  - Key addition every four rounds
- Threefish-256:
  - 256-bit key: K_0, K_1, K_2, K_3
  - 128-bit tweak: T_0, T_1
  - 256-bit text

Skein: Differential Trails

Key schedule (Threefish-256):

- 256-bit key: K_0, K_1, K_2, K_3
- 128-bit tweak: T_0, T_1
- $K_4 := K_0 \oplus K_1 \oplus K_2 \oplus K_3 \oplus C$
- $T_2 := T_0 \oplus T_1$

Round				
0	K_0	$K_1 + T_0$	$K_2 + T_1$	$K_3 + 0$
4	K_1	$K_2 + T_1$	$K_3 + T_2$	$K_4 + 1$
8	K_2	$K_3 + T_2$	$K_4 + T_0$	$K_0 + 2$
12	K_3	$K_4 + T_0$	$K_0 + T_1$	$K_1 + 3$
16	K_4	$K_0 + T_1$	$K_1 + T_2$	$K_2 + 4$

- Use a difference in the tweak and in the key so that they cancel out
- One key addition without any difference

Skein: Differential Trails

Key schedule (Threefish-256):

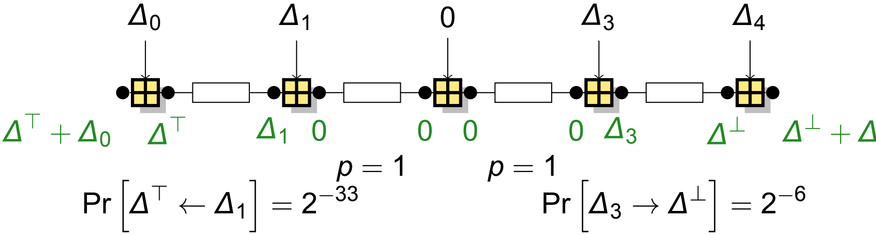
- 256-bit key: K_0, K_1, K_2, K_3
- 128-bit tweak: T_0, T_1
- $K_4 := K_0 \oplus K_1 \oplus K_2 \oplus K_3 \oplus C$
- $T_2 := T_0 \oplus T_1$

Round				
0	K_0	$K_1 + \color{red}{T_0}$	$K_2 + T_1$	$\color{red}{K_3} + 0$
4	K_1	$K_2 + T_1$	$\color{red}{K_3} + \color{red}{T_2}$	$\color{red}{K_4} + 1$
8	K_2	$\color{red}{K_3} + \color{red}{T_2}$	$\color{red}{K_4} + \color{red}{T_0}$	$K_0 + 2$
12	$\color{red}{K_3}$	$\color{red}{K_4} + \color{red}{T_0}$	$K_0 + T_1$	$K_1 + 3$
16	$\color{red}{K_4}$	$K_0 + T_1$	$K_1 + \color{red}{T_2}$	$K_2 + 4$

- Use a difference in the tweak and in the key so that they cancel out
- One key addition without any difference

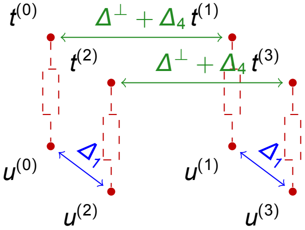
Skein: Differential Trails

- 16-round trail:



- Use a MSB difference for best probability
- Use any difference for auxiliary paths
  - 2^{64} 8-round paths with probability 1

Skein: Description of the Attack



Top path (0–12)

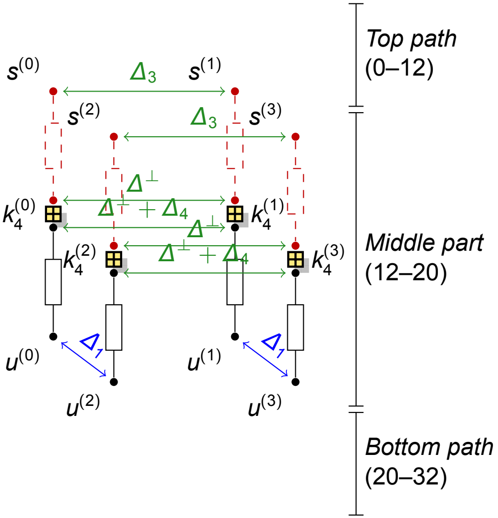
Middle part (12–20)

Bottom path (20–32)

1. Build a quartet for rounds 16—20.
   cost: 2^{18}
2. Extend to rounds 12—20 using random keys.
   cost: 2^{18}
3. Use auxiliary paths to generate quartets.

amortized cost: 2^{0}

Skein: Description of the Attack

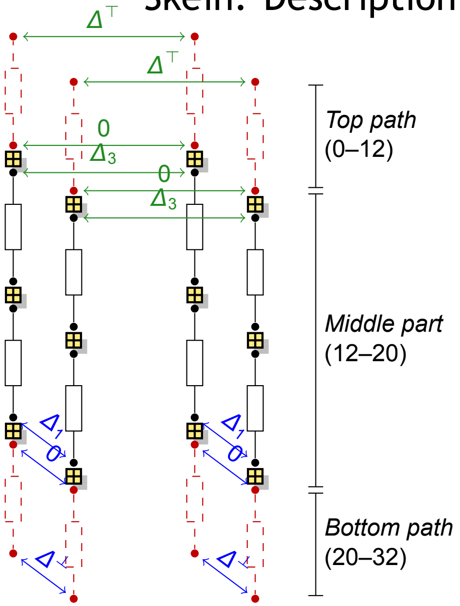


1. Build a quartet for rounds 16—20.
   cost: 2^{18}

2. Extend to rounds 12—20 using random keys.
   cost: 2^{18}

3. Use auxiliary paths to generate quartets.
   amortized cost: 2^{0}

Skein: Description of the Attack

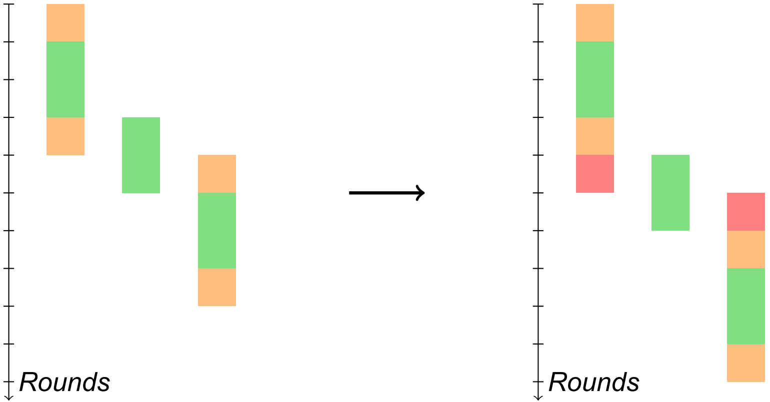


1. Build a quartet for rounds 16—20.
   cost: 2^{18}
2. Extend to rounds 12—20 using random keys.
   cost: 2^{18}
3. Use auxiliary paths to generate quartets.

amortized cost: 2^{0}

Limitations of the Technique

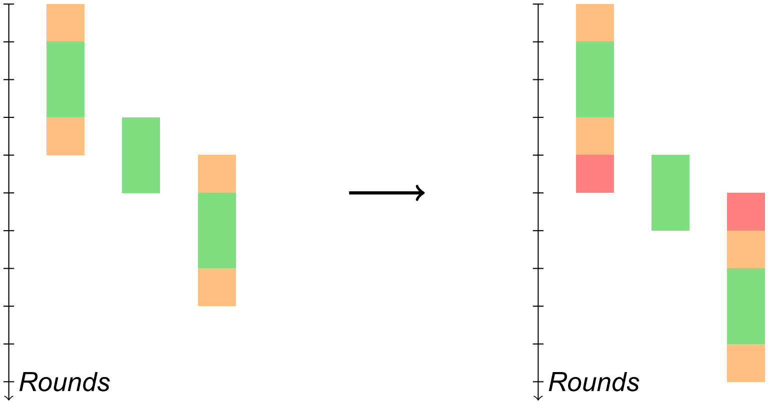
Why not attack more rounds?



Paths are incompatible!

Limitations of the Technique

Why not attack more rounds?



Paths are incompatible!

Incompatible Characteristics

Incompatibilities in Boomerang Paths

- For a Boomerang attack, we usually assume that the path are independent
- We are building a quartet $X^{(0)}, X^{(1)}, X^{(2)}, X^{(3)}$:

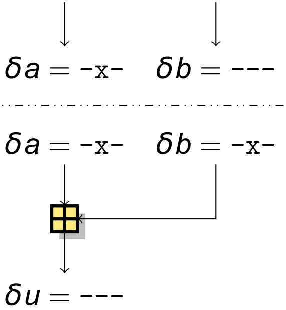
$$X^{(1)} = X^{(0)} + \alpha' \qquad X^{(3)} = X^{(2)} + \alpha'$$
$$X^{(2)} = X^{(0)} + \gamma \qquad X^{(2)} = X^{(1)} + \gamma$$

  We expect:

$$(X^{(0)}, X^{(1)}) \xleftarrow{f_a} \alpha \qquad (X^{(2)}, X^{(3)}) \xleftarrow{f_a} \alpha$$
$$(X^{(0)}, X^{(2)}) \xrightarrow{f_b} \gamma' \qquad (X^{(1)}, X^{(3)}) \xrightarrow{f_b} \gamma'$$

- But these events are not independent! [Murphy 2011]

Boomerang Incompatibility



Top path: $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

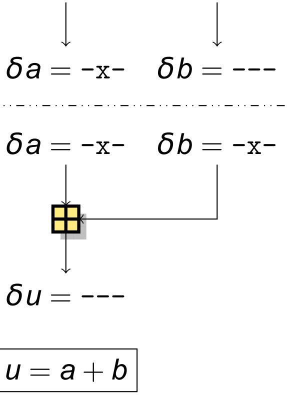
Bottom path: $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$

$$u = a + b$$

	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- Wlog, assume $a^{(0)} = 0$
- Compute $a^{(i)}$, deduce sign of b
- Contradiction for b!

Boomerang Incompatibility



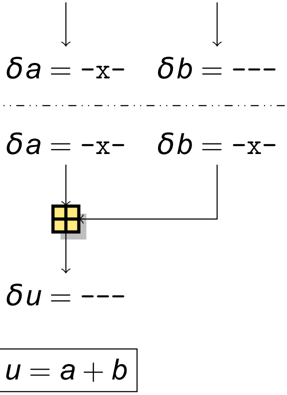
Top path: $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

Bottom path: $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$

	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- Wlog, assume $a^{(0)} = 0$
- Compute $a^{(i)}$, deduce sign of b
- Contradiction for b!

Boomerang Incompatibility



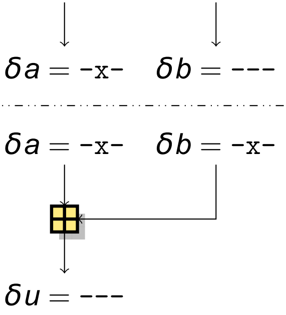
Top path: $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

Bottom path: $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$

	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- Wlog, assume $a^{(0)} = 0$
- Compute $a^{(i)}$, deduce sign of b
- Contradiction for b!

Boomerang Incompatibility



Top path: $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

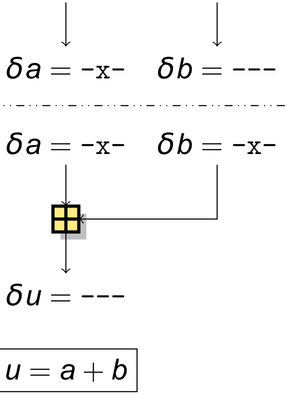
Bottom path: $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$

$u = a + b$

	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- Wlog, assume $a^{(0)} = 0$
- Compute $a^{(i)}$, deduce sign of b
- Contradiction for *b*!

Boomerang Incompatibility



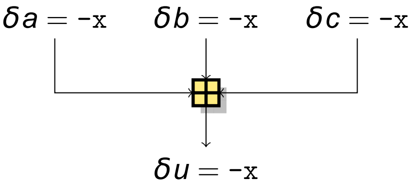
Top path: $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

Bottom path: $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$

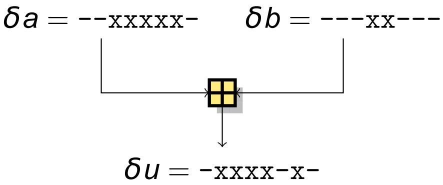
	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- Wlog, assume $a^{(0)} = 0$
- Compute $a^{(i)}$, deduce sign of b
- Contradiction for b!

Other Incompatible Paths



$u = a + b + c$



$u = a + b$

Many “natural” characteristics are in fact incompatible.

- Previous boomerang attacks on Skein-512 do not work
- Works on Skein-256

Results on Skein

Attack	CF/KP	Rounds	CF/KP calls	Ref.
Unknown Key				
Near collisions (Skein-256)	CF	24	2^{60}	[CANS '10]
~~Boomerang dist. (Threefish-512)~~	~~KP~~	~~32~~	~~2^{189}~~	[ISPEC '10]
~~Key Recovery (Threefish-512)~~	~~KP~~	~~34~~	~~$2^{474.4}$~~	[ISPEC '10]
Key Recovery (Threefish-512)	KP	32	2^{312}	[AC '09]
Open key				
Boomerang dist. (Threefish-512)	KP	35	2^{478}	[AC '09]
~~Near collisions (Skein-256)~~	~~CF~~	~~32~~	~~2^{105}~~	[ePrint '11]
Boomerang dist. (Skein-256)	CF and KP	24	2^{18}	
Boomerang dist. (Threefish-256)	KP	28	2^{21}	
Boomerang dist. (Skein-256)	CF	28	2^{24}	
Boomerang dist. (Threefish-256)	KP	32	2^{57}	
Boomerang dist. (Skein-256)	CF	32	2^{114}	

Conclusion

1. Boomerang attack on hash functions
   - Start from the middle
   - Use auxiliary path to avoid middle rounds
   - Significant improvement over previous results
   - New result: also works on Blake

see details

2. Analysis of differentials paths
   - Problems found in several previous works

Appendix

Related work

- Similar to “Boomerang” of Joux and Peyrin (auxiliary paths)
  - In the context of collision attacks
- Similar to message modifications for Boomerang attacks
  - Blake [BNR ’11]
  - SHA-2 [ML ’11]
  - HAVAL [Sasaki ’11]
  - Skein/Threefish [ACMPV ’09, Chen & Jia ’10]
  - Auxiliary paths allow to skip more rounds

New Result: Application to Blake

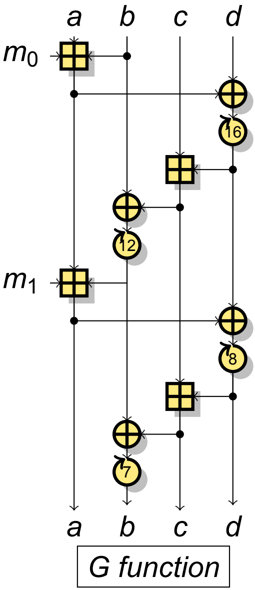
- The same technique can be applied to Blake
  - Another ARX SHA-3 finalist

- Significant improvement over previous results [FSE '11]

- Compression function attack:
  - 6.5 rounds: 2^{140} (vs. ~~2^{184}~~)
  - 7 rounds: 2^{183} (vs. ~~2^{232}~~)

- Keyed-permutation attacks (Open-key vs. Unknown-key)
  - 7 rounds: 2^{32} (vs. ~~2^{122}~~)
  - 8 rounds: 2^{1xx} (vs. ~~2^{242}~~)

Blake

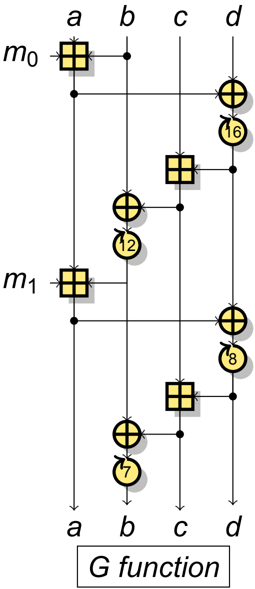


- State is 4×4 matrix:

a_0	a_1	a_2	a_3
b_0	b_1	b_2	b_3
c_0	c_1	c_2	c_3
d_0	d_1	d_2	d_3

- Column step:
  $G(a_0, b_0, c_0, d_0)$
  $G(a_1, b_1, c_1, d_1)$
  $G(a_2, b_2, c_2, d_2)$
  $G(a_3, b_3, c_3, d_3)$
- Diagonal step:
  $G(a_0, b_1, c_2, d_3)$
  $G(a_1, b_2, c_3, d_0)$
  $G(a_2, b_3, c_0, d_1)$
  $G(a_3, b_0, c_1, d_2)$

Blake

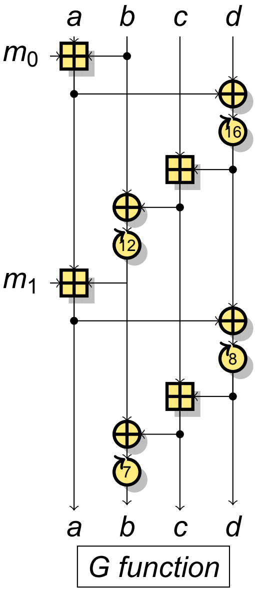


- State is 4×4 matrix:

a_0	a_1	a_2	a_3
b_0	b_1	b_2	b_3
c_0	c_1	c_2	c_3
d_0	d_1	d_2	d_3

- Column step:
  $G(a_0, b_0, c_0, d_0)$
  $G(a_1, b_1, c_1, d_1)$
  $G(a_2, b_2, c_2, d_2)$
  $G(a_3, b_3, c_3, d_3)$
- Diagonal step:
  $G(a_0, b_1, c_2, d_3)$
  $G(a_1, b_2, c_3, d_0)$
  $G(a_2, b_3, c_0, d_1)$
  $G(a_3, b_0, c_1, d_2)$

Blake



- State is 4×4 matrix:

a_0	a_1	a_2	a_3
b_0	b_1	b_2	b_3
c_0	c_1	c_2	c_3
d_0	d_1	d_2	d_3

- Column step:
  $G(a_0, b_0, c_0, d_0)$
  $G(a_1, b_1, c_1, d_1)$
  $G(a_2, b_2, c_2, d_2)$
  $G(a_3, b_3, c_3, d_3)$
- Diagonal step:
  $G(a_0, b_1, c_2, d_3)$
  $G(a_1, b_2, c_3, d_0)$
  $G(a_2, b_3, c_0, d_1)$
  $G(a_3, b_0, c_1, d_2)$

Blake: Differential Trails

- Key schedule: permutation based

σ_3 :	7	3	13	11	9	1	12	14	2	5	4	15	6	10	0	8
σ_4 :	9	5	2	10	0	7	4	15	14	11	6	3	1	12	8	13

- Choose a message word used
  - at the beginning of a round
  - at the end of the next round
- 4-round trail:

$\Pr\left[\Delta^\top \leftarrow \Delta_1\right] = 2^{-6} \cdot 2^{-42}$

$\Pr\left[\Delta_3 \rightarrow \Delta^\perp\right] = 2^{-24}$

Blake: Differential Trails

- Key schedule: permutation based

σ_3 :	7	3	13	11	9	1	12	14	2	5	4	15	6	10	0	8
σ_4 :	9	5	2	10	0	7	4	15	14	11	6	3	1	12	8	13

- Choose a message word used
  - at the beginning of a round
  - at the end of the next round

- 4-round trail:

$p = 1$ $p = 1/2$

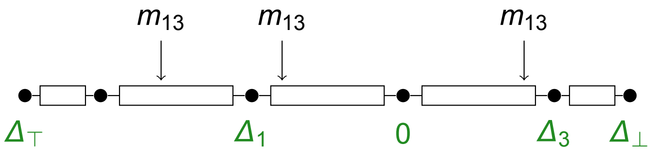
$\Pr\left[\Delta^\top \leftarrow \Delta_1\right] = 2^{-6} \cdot 2^{-42}$ $\Pr\left[\Delta_3 \rightarrow \Delta^\perp\right] = 2^{-24}$

Blake: Differential Trails

- Key schedule: permutation based

σ_3 :	7	3	13	11	9	1	12	14	2	5	4	15	6	10	0	8
σ_4 :	9	5	2	10	0	7	4	15	14	11	6	3	1	12	8	13

- Choose a message word used
  - at the beginning of a round
  - at the end of the next round
- 4-round trail:



$p = 1$ $\quad$ $p = 1/2$

$\Pr\left[\Delta^\top \leftarrow \Delta_1\right] = 2^{-6} \cdot 2^{-42}$ $\quad$ $\Pr\left[\Delta_3 \rightarrow \Delta^\bot\right] = 2^{-24}$

Blake: Description of the Attack

The hard part is the middle round

- Column step is part of the top path
- Diagonal step is part of the bottom path

1. Find (state, message) candidates for each diagonal G function
   - Start with middle quartets with all differences fixed

2. Look for combinations of candidates that follow the first part of the diagonal step
   - Use the message to randomize

3. Look for candidates that follow the full diagonal step
   - Use the message to randomize

Blake-256: Results

Attack	CF/KP	Rounds	CF/KP calls	Ref.
Unknown Key				
Boomerang dist.	KP	7	2^{122}	[FSE '11]
~~Boomerang dist.~~	~~KP~~	~~8~~	~~2^{242}~~	[FSE '11]
Open Key				
~~Boomerang dist.~~	~~CF w/ Init~~	~~7~~	~~2^{232}~~	[FSE '11]
Boomerang dist.	CF w/ Init	7	2^{183}	
Boomerang dist.	KP	7	2^{32}	
Boomerang dist.	KP	8	2^{1xx}	