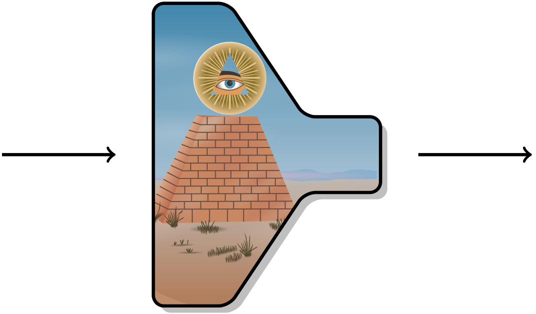
Design and Analysis of Hash Functions

Gaëtan Leurent

École normale supérieure
Paris, France

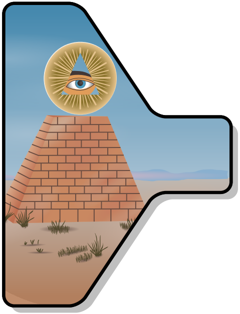
Ph.D. Defense
September 30, 2010

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

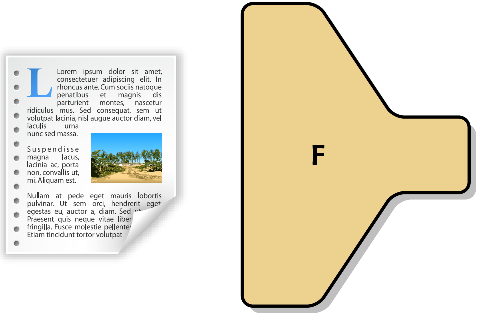


`0x1d66ca77ab361c6f`

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

A Concrete Hash Function

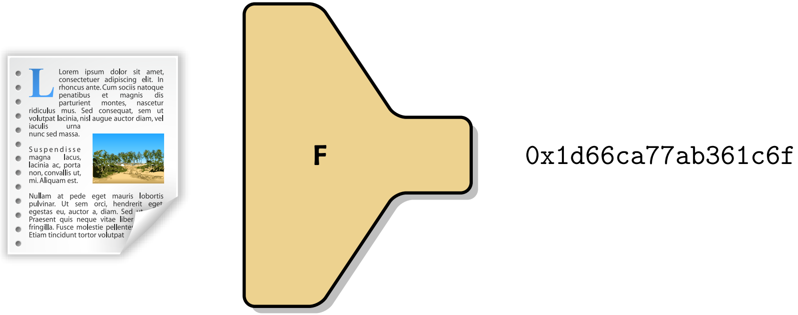
- A public function with no structural property.
  - Cryptographic strength without any key!
- $F : \{0,1\}^* \to \{0,1\}^n$



0x1d66ca77ab361c6f

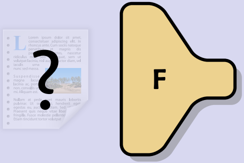
A Concrete Hash Function

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



Security goals

Preimage attack

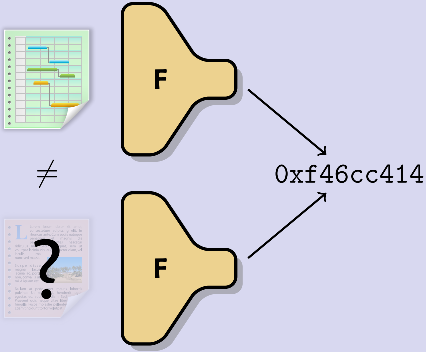


Given F and $\overline{H}$, find M s.t. $F(M) = \overline{H}$.

Ideal security: 2^n.

Security goals

Second-preimage attack

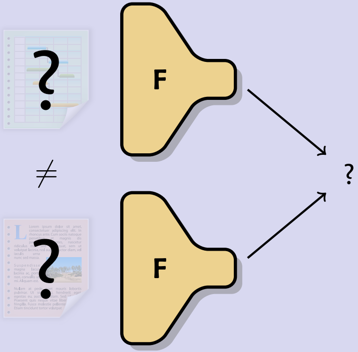


Given F and M_1, find $M_2 \neq M_1$ s.t. $F(M_1) = F(M_2)$.

Ideal security: 2^n.

Security goals

Collision attack



Given F, find $M_1 \neq M_2$ s.t. $F(M_1) = F(M_2)$.

Ideal security: $2^{n/2}$.

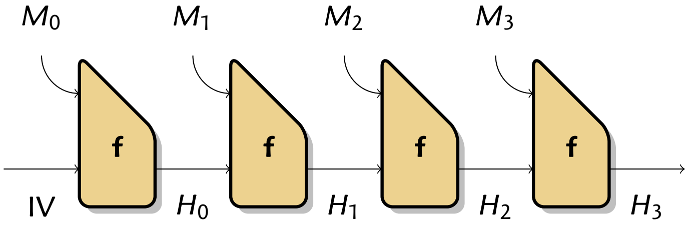
Using Hash Functions

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

Hash function design

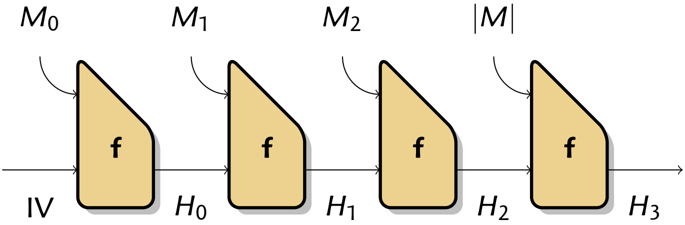
- Build a small compression function, and iterate.
  - Cut the message in chunks $M_0, \ldots M_k$
  - $H_i = f(M_i, H_{-1})$
  - $F(M) = H_k$



Security proof (Merkle, Damgård)

Theorem

If one finds a collision in the hash function, then one has a collision in the compression function.

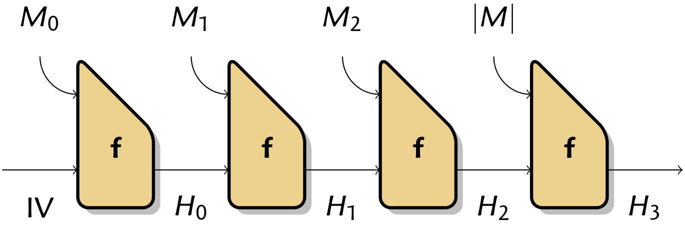


- If $|M| \neq |M'|$, collision in last block.
- Else, look for last block with $H_i = H'_i$.
- The converse is not true

Security proof (Merkle, Damgård)

Theorem

If one finds a collision in the hash function, then one has a collision in the compression function.



- If $|M| \neq |M'|$, collision in last block.
- Else, look for last block with $H_i = H'_i$.
- The converse is not true

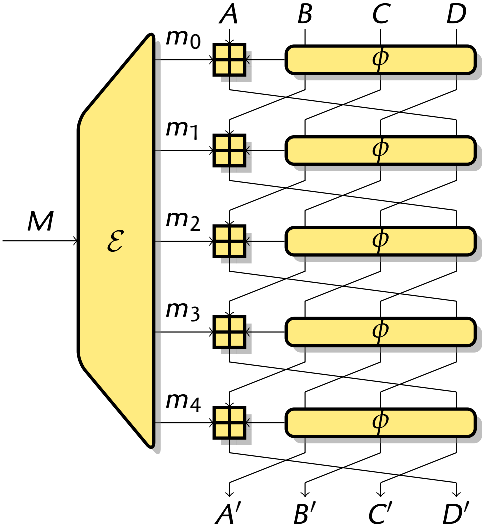
Outline

Outline

MD family design

- MD4 was one of the first dedicated hash functions
- Most of the hash functions used today are derived from MD4
  - MD4, MD5, SHA-1, SHA-2, RIPEMD, ...
- It is important to study their security

MD family design

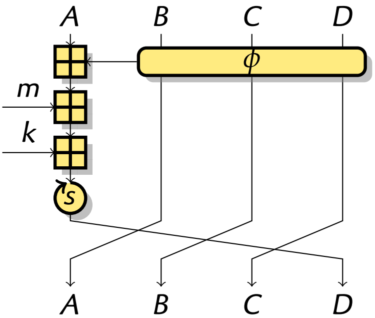


- Input:

$$M \leftarrow \textbf{Message}$$
$$(A, B, C, D) \leftarrow \textbf{Chaining value}$$

- Output: $(A + A', B + B', C + C', D + D')$
- 32/64-bit registers
- Simple operations
- Message expansion: permutation based for MD4/MD5

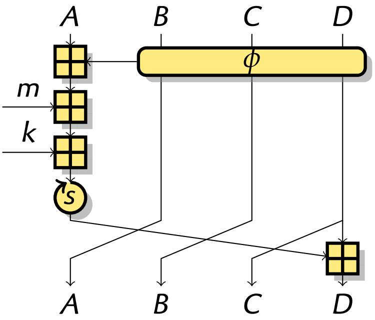
MD4 design



$$Q_i = (Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}))^{\lll s_i}$$

- 48 steps (16 message words)
- Boolean functions: IF, MAJ, XOR

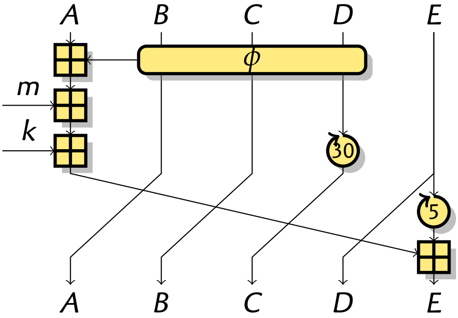
MD5 design



$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i} \boxplus Q_{i-1}$$

- 64 steps (16 message words)
- Boolean functions: IF, MAJ, XOR, ONX

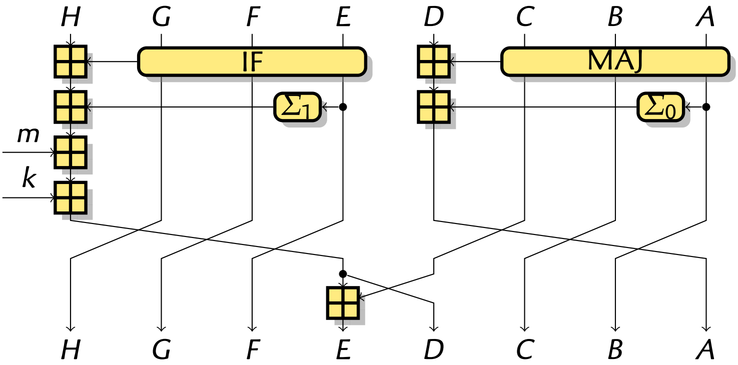
SHA-1 design



$$Q_i = Q_{i-5}^{\lll 30} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-2}, Q_{i-3}^{\lll 30}, Q_{i-4}^{\lll 30}) \boxplus Q_{i-1}^{\lll 5}$$

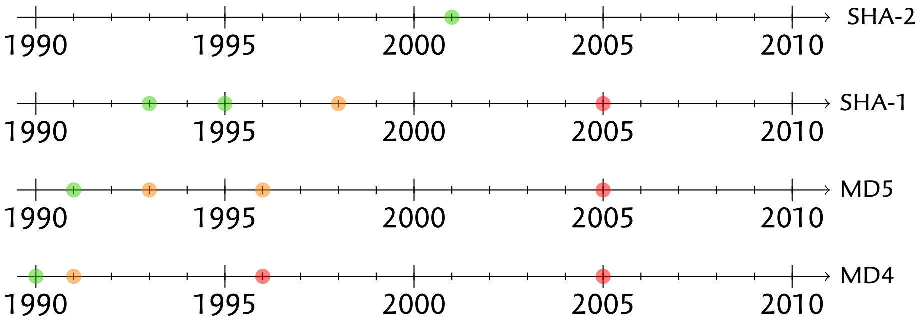
- 80 steps (16 message words)
- Boolean functions: IF, MAJ, XOR
- Stronger message expansion
  - $m_i = (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16})^{\lll 1}$

SHA-2 design



- 64 steps for SHA-224/256; 80 steps for SHA-384/512
- Σ functions: sum of three rotations
- Stronger message expansion: non-linear code
  - $m_i = \sigma_1(m_{i-2}) \boxplus m_{i-7} \boxplus \sigma_0(m_{i-15}) \boxplus m_{i-16}$

Attacks



- In 2005, a series of attacks against MD4, MD5, SHA-1, RIPEMD-0, . . .

Main mistakes

MD4 Not enough rounds

MD5 A difference in the MSB can stay in the MSB
(Den Boer and Bosselaers, 1993)
$Q'_i = Q_i \oplus 2^{31}$
$Q_i = (Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}))^{\lll s_i} \boxplus Q_{i-1}$

SHA-1 Message expansion is a cyclic linear code
It is possible to shift a difference pattern
Used to build local collisions

MD family status

Current status

Collision-resistance is seriously broken (MD4, MD5, SHA-1),
but for most constructions, no real attacks are known:

- Key derivation
- Peer authentication
- HMAC
- ...

More in-depth study and improvement of Wang's attack are needed.

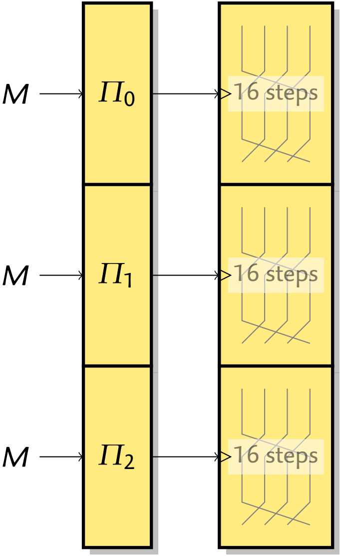
My contributions

- Improvements of Wang *et al.*'s attack, and new applications
  - Message Freedom in MD4 and MD5 Collisions: Application to APOP
    G. Leurent [FSE '07 + IJACT]
  - Automatic Search of Differential Paths in MD4
    P.-A. Fouque, G. Leurent, P. Nguyen [Hash Workshop '07]
  - Full Key-Recovery Attacks on HMAC/NMAC-MD4 and NMAC-MD5
    P.-A. Fouque, G. Leurent, P. Nguyen [Crypto '07]
- The first preimage attack on a member of the MD4 family
  - MD4 is Not One-Way
    G. Leurent [FSE '08]
- A low-complexity side-channel attack on HMAC-SHA1
  - Practical Electromagnetic Template Attack on HMAC
    P.-A. Fouque, G. Leurent, D. Réal, F. Valette [CHES '09]

Wang et. al's attacks

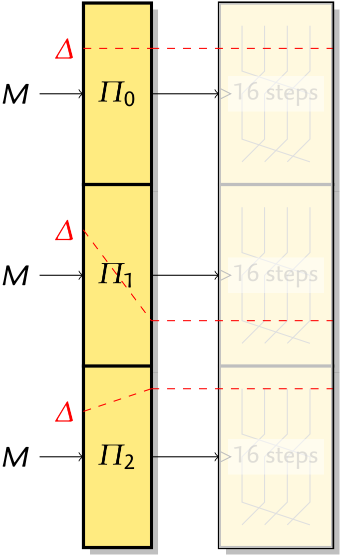
- Based on a differential attack:
  - Consider a pair of message with a small difference
  - Try to control the propagation of the differences
- New ideas:
  - Use a signed difference
  - Use a set of sufficient conditions
  - Some conditions are easy to satisfy:
    message modification

Wang et al.'s Attack



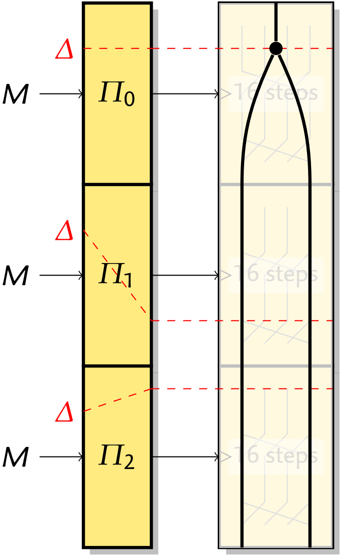
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



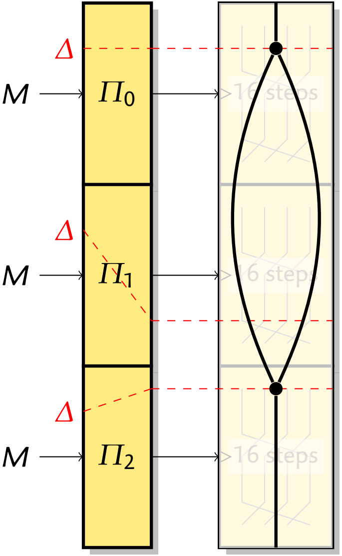
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



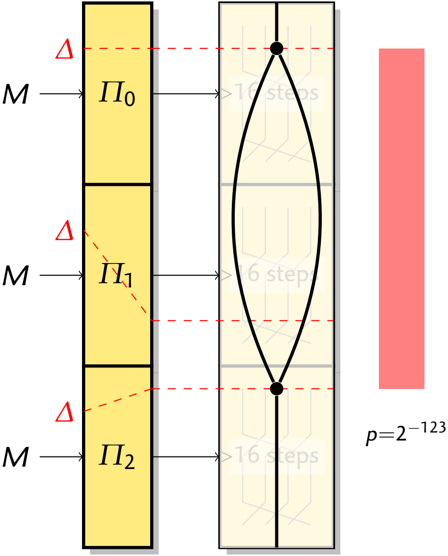
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



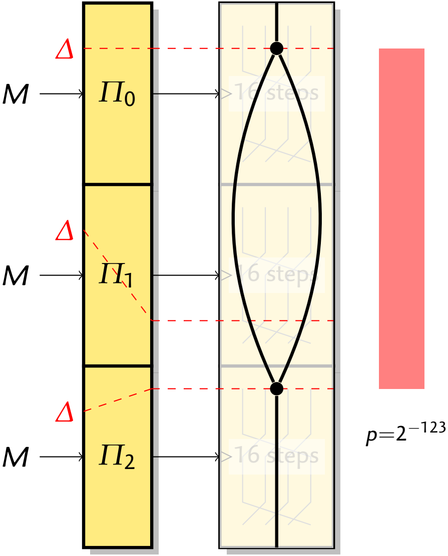
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



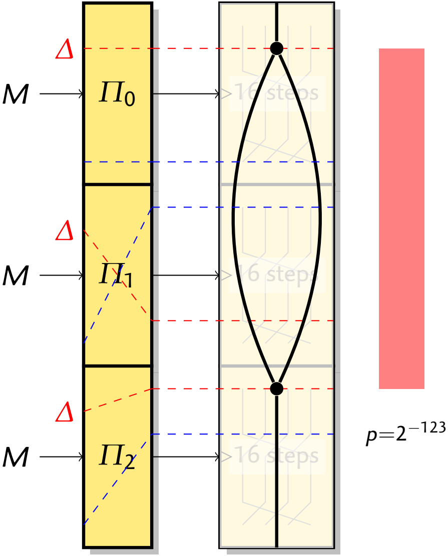
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



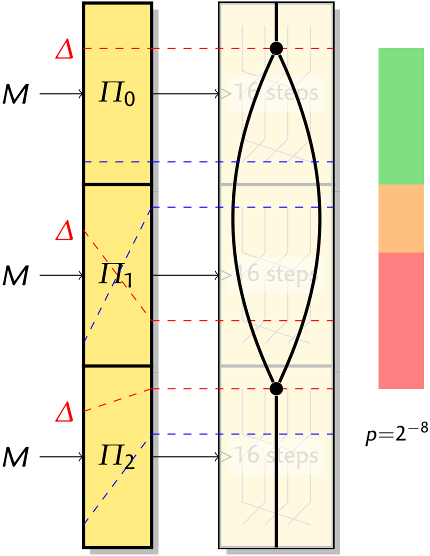
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Wang et al.'s Attack



1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

i	s_i	δm_i	$\partial \Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

i	s_i	δm_i	$\partial\Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

i	s_i	δm_i	$\partial\Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

i	s_i	δm_i	$\partial\Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

i	s_i	δm_i	$\partial\Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

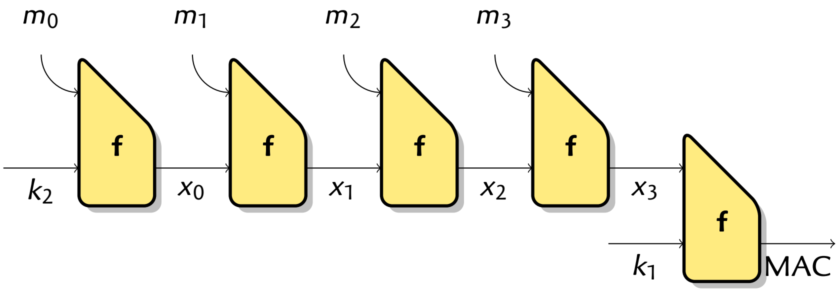
i	s_i	δm_i	$\partial\Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

A Differential Path

$$Q_i = \left(Q_{i-4} \boxplus m_i \boxplus k_i \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})\right)^{\lll s_i}$$

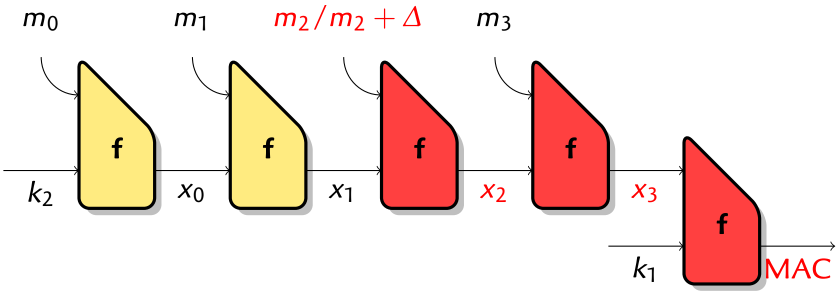
i	s_i	δm_i	$\partial\Phi_i$	∂Q_i	Φ-conditions
0	3				
1	7	$\langle ▲^{[31]} \rangle$		$\langle ▲^{[6]} \rangle$	
2	11	$\langle ▼^{[28]}, ▲^{[31]} \rangle$		$\langle ▼^{[7]}, ▲^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0$, $Q_1^{[7]} = Q_0^{[7]}$, $Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle ▲▼^{[6,7]} \rangle$	$\langle ▲▲▼^{[9\ldots11]} \rangle$	$Q_3^{[6]} = 0$, $Q_3^{[7]} = 1$, $Q_3^{[10]} = 0$
5	7			$\langle ▲^{[13]} \rangle$	$Q_4^{[7]} = 1$, $Q_3^{[9]} = Q_2^{[9]}$, $Q_3^{[10]} = 0$, $Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle ▲▼^{[10,11]} \rangle$	$\langle ▼^{[18]} \rangle$	$Q_5^{[9]} = 0$, $Q_5^{[10]} = 1$, $Q_5^{[11]} = 1$, $Q_4^{[13]} = Q_3^{[13]}$

Application to NMAC



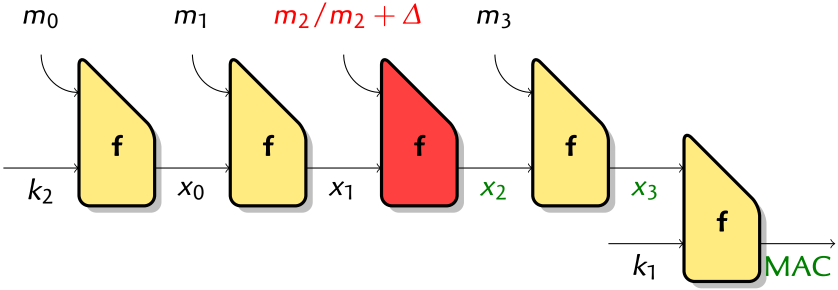
- Cannot use message modification because of the key
- Put a difference Δ in m_2
- With some probability it collides in x_2 and in the MAC (2^{-58})
- The collision reveals some key information
  - Contini and Yin proposed a way to extract key information using message modifications.

Application to NMAC



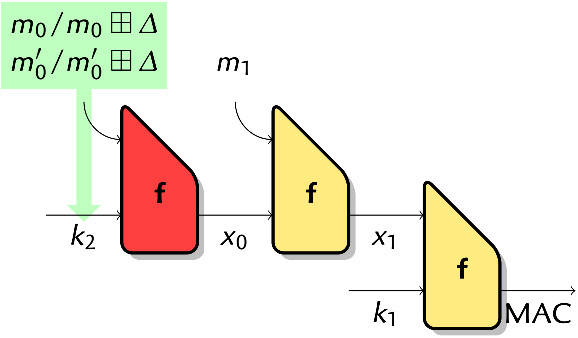
- Cannot use message modification because of the key
- Put a difference Δ in m_2
- With some probability it collides in x_2 and in the MAC (2^{-58})
- The collision reveals some key information
  - Contini and Yin proposed a way to extract key information using message modifications.

Application to NMAC



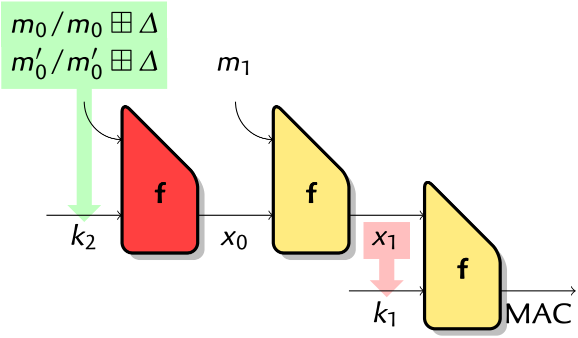
- Cannot use message modification because of the key
- Put a difference Δ in m_2
- With some probability it collides in x_2 and in the *MAC* (2^{-58})
- The collision reveals some key information
  - Contini and Yin proposed a way to extract key information using message modifications.

Application to NMAC



- We can recover k_2 using m_0
- But we don't have control over $x_1 = H_{k_2}(M)$ to recover k_1

Application to NMAC



- We can recover k_2 using m_0
- But we don't have control over x_1 $= H_{k_2}(M)$ to recover k_1

A New IV-recovery Attack

- We want to avoid the need for related messages.
- We look for paths where the existence of collision discloses information about the key.

Advantage

- In attack of Contini and Yin attack, one needs to control a lot of bits of $H_{k_2}(M)$ (related messages).
- We only need to choose differences in $H_{k_2}(M)$.

Using IV-dependent paths

- Use a differential path with $\delta m_0 \neq 0$.
- The beginning of the path depends on a condition (X) of the IV:
  - $\Pr_M[H(M) = H(M+\Delta)|X] \gg 2^{-128}$.

step	δm_i	$\partial\Phi_i$	∂Q_i	conditions
0	$\langle \blacktriangle^{[0]} \rangle$		$\langle \blacktriangle^{[3]} \rangle$	
1				$Q_{-1}^{[3]} = Q_{-2}^{[3]}$ (X)

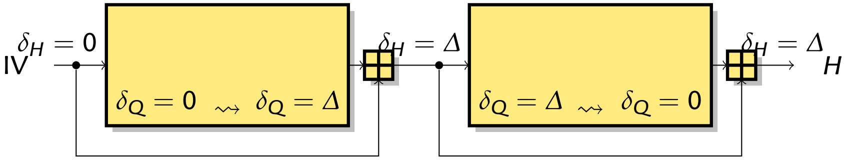
  - $\Pr_M[H(M) = H(M+\Delta)|\neg X] \approx 2^{-128}$.

step	δm_i	$\partial\Phi_i$	∂Q_i	conditions
0	$\langle \blacktriangle^{[0]} \rangle$		$\langle \blacktriangle^{[3]} \rangle$	
1		$\langle \blacktriangle^{[3]} \rangle$	$\langle \blacktriangle^{[10]} \rangle$	$Q_{-1}^{[3]} \neq Q_{-2}^{[3]}$ ($\neg$ X)

- We try $2/p_X$ pairs:
  - If we have a collision then (X) is satisfied.
  - Otherwise, (X) is not satisfied.

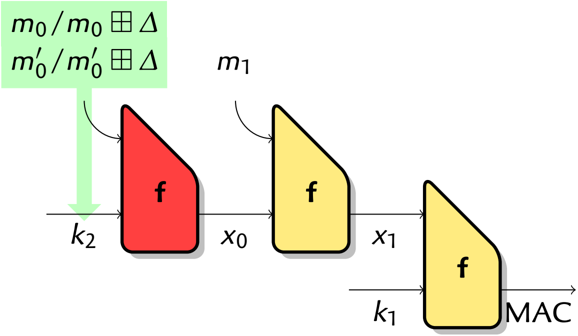
Efficient computation of message pairs

To recover the outer key, we need $2/p_X$ message pairs with $H_{k_2}(M_2) = H_{k_2}(M_1) + \Delta$



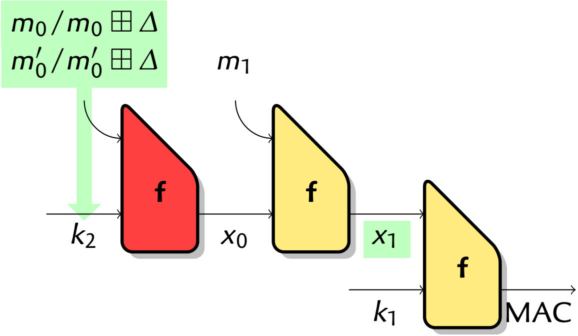
- We start with *one* message pair (R_1, R_2) such that $H_{k_2}(R_2) = H_{k_2}(R_1) + \Delta$ (birthday paradox).
- We compute second blocks (N_1, N_2) such that $H_{k_2}(R_2 \| N_2) = H_{k_2}(R_1 \| N_1) + \Delta$
- This is essentially a collision search with the padding inside the block.

New outer key recovery



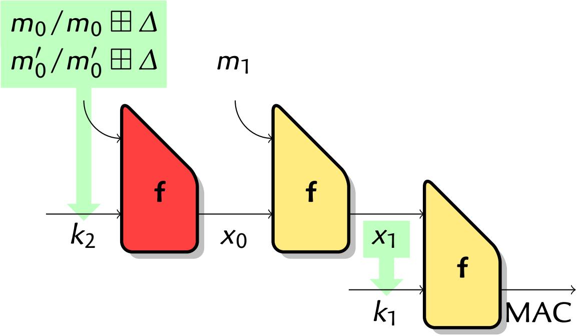
1. Recover k_2.
2. Generate pairs with $H_{k_2}(M_2) = H_{k_2}(M_1) + \Delta$.
3. Learn bits of k_1 by observing collisions.

New outer key recovery



1. Recover k_2.
2. Generate pairs with $H_{k_2}(M_2) = H_{k_2}(M_1) + \Delta$.
3. Learn bits of k_1 by observing collisions.

New outer key recovery



1. Recover k_2.
2. Generate pairs with $H_{k_2}(M_2) = H_{k_2}(M_1) + \Delta$.
3. Learn bits of k_1 by observing collisions.

Results

Attacks		Data	Time	Mem	Remark
Generic	E-Forgery	$2^{n/2}$	-	$2^{n/2}$	Collision based
	U-Forgery	$2^{n/2}$	2^{n+1}	$2^{n/2}$	Collision based
		1	$2^{2n/3}$	$2^{2n/3}$	TM tradeoff, 2^n precpu
NMAC-MD4 HMAC-MD4	E-Forgery	2^{58}	-	-	[Contini-Yin]
	Partial-KR	2^{63}	2^{40}	-	[Contini-Yin]
	U-Forgery	2^{88}	2^{95}	-	Our result
		2^{72}	2^{77}	-	[L.Wang *et al.*]

Outline

The SHA-3 competition

After the attacks on the MD4 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 November 2010?
- Winner in 2012?

Design Choices

SIMD is designed to be:

- Vectorisable
- With a strong message expansion
- Wide-pipe

SIMD is a Message Digest
G. Leurent, C. Bouillaguet, P.-A. Fouque [SHA-3 submission]

Security Analysis of SIMD
C. Bouillaguet, P.-A. Fouque, G. Leurent [SAC '10]

Speed vs Security

NIST wants SHA-3 to be faster and more secure than SHA-2.

- More secure: more operations
- Faster: less time

- We need to cheat (use the hardware more efficiently)
  - Use multiple cores (e.g. MD6)
  - Use AES instructions (e.g. ECHO, SHAvite-3)
  - Use 64-bit integers (e.g. Skein, BMW-512)
  - Use vector instructions (e.g. Blake, CubeHash, Hamsi, JH, Keccak, Luffa, SIMD)

- Vector instructions are more widely available than 64-bit integers or AES instructions.
  - SSE2 on x86, AltiVec on PowerPC, IwMMXt or NEON on ARM, ...

Speed vs Security

NIST wants SHA-3 to be faster and more secure than SHA-2.

- More secure: more operations
- Faster: less time

- We need to cheat (use the hardware more efficiently)
  - Use multiple cores (*e.g.* MD6)
  - Use AES instructions (*e.g.* ECHO, SHAvite-3)
  - Use 64-bit integers (*e.g.* Skein, BMW-512)
  - Use vector instructions (*e.g.* Blake, CubeHash, Hamsi, JH, Keccak, Luffa, SIMD)

- Vector instructions are more widely available than 64-bit integers or AES instructions.
  - SSE2 on x86, AltiVec on PowerPC, IwMMXt or NEON on ARM, ...

Speed vs Security

NIST wants SHA-3 to be faster and more secure than SHA-2.

- More secure: more operations
- Faster: less time

- We need to cheat (use the hardware more efficiently)
  - Use multiple cores (*e.g.* MD6)
  - Use AES instructions (*e.g.* ECHO, SHAvite-3)
  - Use 64-bit integers (*e.g.* Skein, BMW-512)
  - Use vector instructions (*e.g.* Blake, CubeHash, Hamsi, JH, Keccak, Luffa, SIMD)

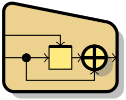
- Vector instructions are more widely available than 64-bit integers or AES instructions.
  - SSE2 on x86, AltiVec on PowerPC, IwMMXt or NEON on ARM, ...

Strong Message Expansion

- The inputs of a compression function have different roles:
  - The message is controlled by the adversary
  - The chaining value is only known
- Use a strong transformation on the message.
  - Trade-off: spend more time where it matters.

- In Davies-Meyer mode, we have a message expansion.

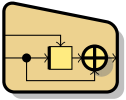
- Davies-Meyer:



$$H_i = E_M(H_{i-1}) \oplus H_{i-1}$$

- differential attack on C
  $\rightsquigarrow$ related key attack on E

- Matyas-Meyer-Oseas:



$$H_i = E_{H_{i-1}}(M) \oplus M$$

- differential attack on C
  $\rightsquigarrow$ differential attacks E

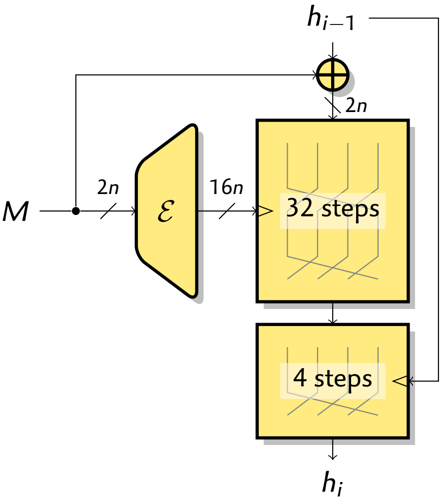
SIMD Message Expansion

- Code with large minimal distance:

	Msg. block	Expanded msg.	Min. distance
SIMD-256	512 bits	4096 bits	520 bits
SIMD-512	1024 bits	8192 bits	1032 bits

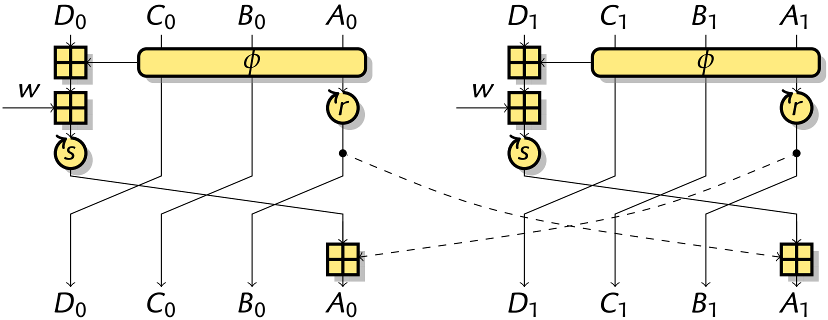
- Concatenated code
  - Outer code gives a high word distance
    - Reed-Solomon code over $\mathbb{F}_{257}$
  - Inner code gives a high bit distance
    - Multiplication by a constant (185 / 233)
- We can derive bounds for differential paths.

SIMD Compression Function



- Block cipher based
  - Well understood
- Davies-Meyer
  - Allows a strong message expansion
- Add the message at the start
  - Prevents some message modifications
- Modified feed-forward: Feistel rounds instead of XOR
  - Avoids some fixed point and multi-block attacks

SIMD Feistel Rounds



- Follows the SHA/MD legacy
  - Additions, rotations, boolean functions
  - Well understood

- 4 Parallel lanes for SIMD-256, 8 for SIMD-512
- Parallel Feistel rounds allow vectorized implementation

Performance

- Vectorized implementations for SSE2, Altivec, and IwMMXt
  - Gives an idea of performances for a generic CPU with SIMD unit

Processor	Core 2	Atom	PowerPC G4	ARM Xscale
SHA-1	1	1	1	1
SHA-256	0.55	0.55	0.55	0.60
SHA-512	0.70	0.20	0.15	0.15
SIMD256	0.85	0.95	0.75	0.45
SIMD512	0.75	0.75	0.55	

Normalized speed

- Vector units are available on all desktop/laptop/netbook and becoming available on embedded machines
- They will get more powerful:
  AVX on Intel (Q4 2010), AVX+XOP on AMD (2011)

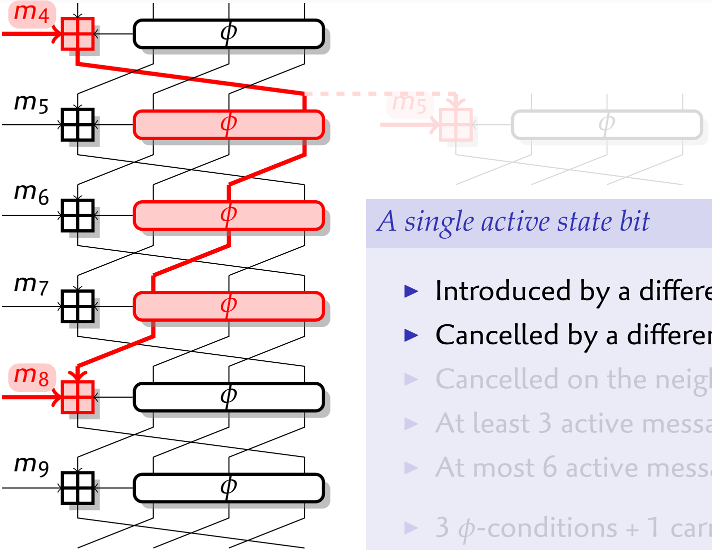
Security Analysis: Differential Attacks

- We assume that the adversary builds a differential path with a signed difference.
- We consider paths with a non-zero message difference
  - paths with no message difference only give free-start attacks

- Each active state bit lowers the probability
  - Minimize active state bits

- The message expansion gives many message differences
  - 520 for SIMD-256
  - 1032 for SIMD-512

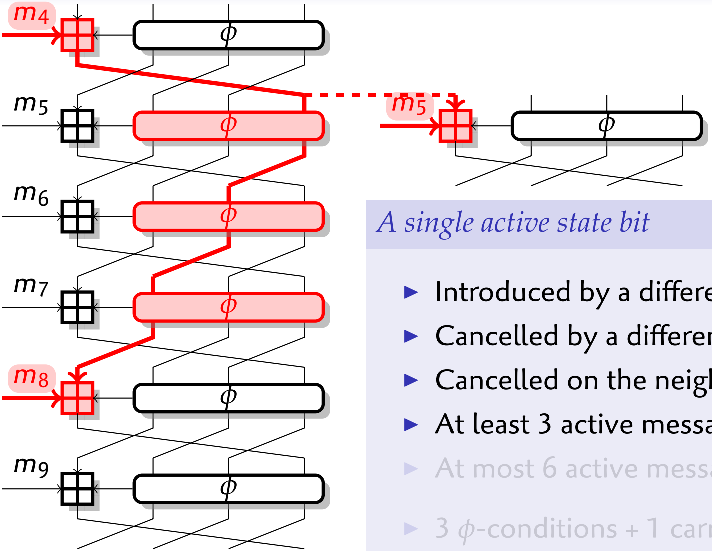
Local Collisions



A single active state bit

- Introduced by a difference in m_4
- Cancelled by a difference in m_8
- Cancelled on the neighbour lane
- At least 3 active messages
- At most 6 active messages
- 3 ϕ-conditions + 1 carry condition

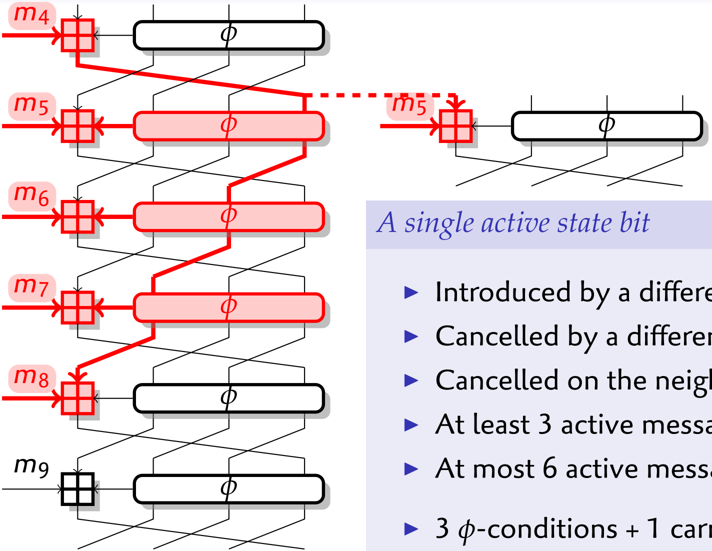
Local Collisions



A single active state bit

- Introduced by a difference in m_4
- Cancelled by a difference in m_8
- Cancelled on the neighbour lane
- At least 3 active messages
- At most 6 active messages
- 3 ϕ-conditions + 1 carry condition

Local Collisions



A single active state bit

- Introduced by a difference in m_4
- Cancelled by a difference in m_8
- Cancelled on the neighbour lane
- At least 3 active messages
- At most 6 active messages

- 3 ϕ-conditions + 1 carry condition

Heuristic

Heuristic

The adversary can build an expanded message of minimal weight

- such that the differences create local collisions
- but without any extra property

- Optimal path: all Boolean function transmit differences
  - Minimizes the number of active state bits
- 6 active message bits per active state bit
  - 87 active state bits for SIMD-256 / 172 for SIMD-512
- 4 conditions per active state bit
  - 348 conditions for SIMD-256 / 688 for SIMD-512

Outline

Introduction
- Hash Functions

Analysis of the MD4 family
- Description of the MD4 family
- Wang et al.'s attack
- Key-recovery attack on HMAC/NMAC-MD4

The Design of SIMD
- The SHA-3 Competition
- Design choices
- Description of SIMD
- Security Analysis: Differential Paths

Attacks on New Hash Functions
- The cancellation property
- Application to *Lesamnta*

My contributions I

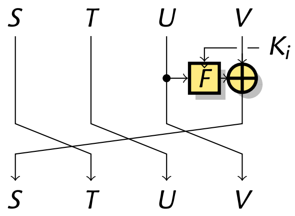
- Attacks on SHA-3 candidates
  - Practical Key Recovery Attack against Secret-IV EDON-$\mathcal{R}$
    G. Leurent [CT-RSA '10]
  - Another Look at the Complementation Property
    C. Bouillaguet, O. Dunkelman, P.-A. Fouque, G. Leurent [FSE '10]
  - Cryptanalysis of ESSENCE
    M. Naya-Plasencia, A. Röck, J.-P. Aumasson, Y. Laigle-Chapuy, G. Leurent, W. Meier, T. Peyrin [FSE '10]
  - Attacks on Hash Functions based on Generalized Feistel - Application to Reduced-Round Lesamnta and *SHAvite-3*$_{512}$
    C. Bouillaguet, O. Dunkelman, P.-A. Fouque, G. Leurent [SAC '10]
  - Cryptanalysis of the 10-Round Hash and Full Compression Function of *SHAvite-3*$_{512}$
    P. Gauravaram, G. Leurent, F. Mendel, M. Naya-Plasencia, T. Peyrin, C. Rechberger, M. Schläffer [Africacrypt '10]

My contributions II

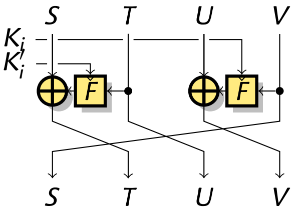
- Other results
  - Cryptanalysis of a Hash Function Based on Quasi-Cyclic Codes
    P.-A. Fouque et G. Leurent [CT-RSA '08]
  - How risky is the Random-Oracle Model?
    G. Leurent et P. Q. Nguyen [Crypto '09]

Generalized Feistel schemes

- Build a 4n-bit hash function out of an n-bit function:



Lesamnta structure

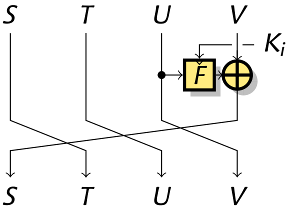


SHAvite-3$_{512}$ structure

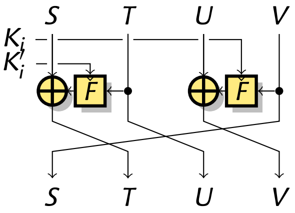
- Ideal: each F_i is an independent ideal function/permutation
- In practice: $F_i(x) = F(k_i \oplus x)$ with a fixed F

Generalized Feistel schemes

- Build a 4n-bit hash function out of an n-bit function:



Lesamnta structure



SHAvite-3$_{512}$ structure

- Ideal: each F_i is an independent ideal function/permutation
- In practice: $F_i(x) = F(k_i \oplus x)$ with a fixed F

Cancellation Cryptanalysis

Main idea

Cancel the effect of non-linear components
by using the same input pairs twice

- Generalized Feistel with slow diffusion
- Hash function setting
- $F_i(x) = F(k_i \oplus x)$
  - $\exists c_{i,j} : \forall x,\ F_i(x \oplus c_{i,j}) = F_j(x)$
    - $c_{ij} = k_i \oplus k_j$

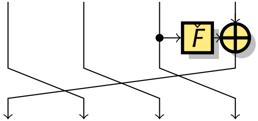
Cancellation Cryptanalysis

Main idea

Cancel the effect of non-linear components
by using the same input pairs twice

- Generalized Feistel with slow diffusion
- Hash function setting
- $F_i(x) = F(k_i \oplus x)$
  - $\exists c_{i,j} : \forall x,\ F_i(x \oplus c_{i,j}) = F_j(x)$
    - $c_{ij} = k_i \oplus k_j$

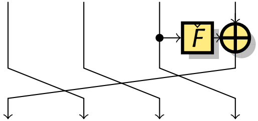
The Cancellation Property



- Full diffusion after 9 rounds
- If $y_1 = y_2 = y$, the differences cancel out
- Use constraints on the state

i	S_i	T_i	U_i	V_i	
0	x	-	-	-	
1	-	x	-	-	
2	-	-	x	-	
3	y_1	-	-	x	$x \to y_1$
4	x	y_1	-	-	
5	-	x	y_1	-	
6	z	-	x	y_1	$y_1 \to z$
7	y'	z	-	x	$x \to y_2, y' = y_1 \oplus y_2$
8	x	y'	z	-	
9	w	x	y'	z	$z \to w$

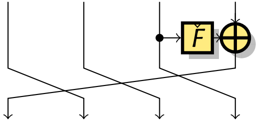
The Cancellation Property



- Full diffusion after 9 rounds
- If $y_1 = y_2 = y$, the differences cancel out
- Use constraints on the state

i	S_i	T_i	U_i	V_i	
0	x	-	-	-	
1	-	x	-	-	
2	-	-	x	-	
3	y_1	-	-	x	$x \to y_1$
4	x	y_1	-	-	
5	-	x	y_1	-	
6	z	-	x	y_1	$y_1 \to z$
7	y'	z	-	x	$x \to y_2, y' = y_1 \oplus y_2$
8	x	y'	z	-	
9	w	x	y'	z	$z \to w$

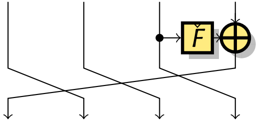
The Cancellation Property



- Full diffusion after 9 rounds
- If $y_1 = y_2 = y$, the differences cancel out
- Use constraints on the state

i	S_i	T_i	U_i	V_i	
0	x	-	-	-	
1	-	x	-	-	
2	-	-	x	-	
3	y	-	-	x	$x \to y$
4	x	y	-	-	
5	-	x	y	-	
6	z	-	x	y	$y_1 \to z$
7	-	z	-	x	$x \to y$
8	x	-	z	-	
9	w	x	-	z	$z \to w$

The Cancellation Property



- Full diffusion after 9 rounds
- If $y_1 = y_2 = y$, the differences cancel out
- Use constraints on the state

i	S_i	T_i	U_i	V_i	
0	x	-	-	-	
1	-	x	-	-	
2	-	-	x	-	
3	y	-	-	x	$x \to y$
4	x	y	-	-	
5	-	x	y	-	
6	z	-	x	y	$y_1 \to z$
7	-	z	-	x	$x \to y$
8	x	-	z	-	
9	w	x	-	z	$z \to w$

The Cancellation Property: Looking at the Values

We study values, starting at round 2:

i	S_i	T_i	U_i	V_i
2	a	b	c	d
3	$F_2(c) \oplus d$	a	b	c
4	$F_3(b) \oplus c$	$F_2(c) \oplus d$	a	b
5	$F_4(a) \oplus b$	$F_3(b) \oplus c$	$F_2(c) \oplus d$	a
6	$F_5(F_2(c) \oplus d) \oplus a$	$F_4(a) \oplus b$	$F_3(b) \oplus c$	$F_2(c) \oplus d$
7	$\cancel{F_6(F_3(b) \oplus \underline{c})} \oplus \cancel{F_2(\underline{c})} \oplus d$	$F_5(F_2(c) \oplus d) \oplus a$	$F_4(a) \oplus b$	$F_3(b) \oplus c$

Round 7: $F_6(F_3(b) \oplus \underline{c}) \oplus F_2(\underline{c})$. They cancel if:

$F_3(b) = c_{2,6} = K_2 \oplus K_6$

i.e. $b = F_3^{-1}(K_2 \oplus K_6)$

Attack Overview

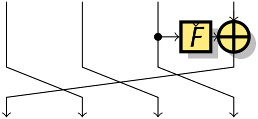
- Partial preimage: Choose one part of the output
  - Gives preimage and collision attacks.
- Hash function setting: no key.
- Mostly generic in the round function.

Attack Strategy

- Set parts of the state to satisfy the cancellation conditions.
- The truncated differential path describes
  how the output depends on the remaining degrees of freedom
- Compute the required value

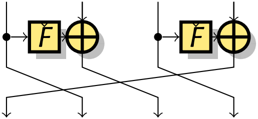
Result Overview

- Attacks on reduced *Lesamnta*



  - 24 rounds out of 32: collision and preimage
  - previous attacks: 16 rounds

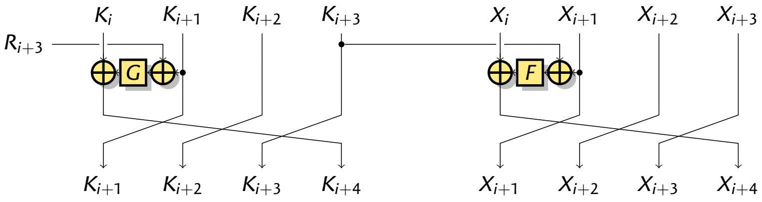
- Attacks on reduced *SHAvite-3*$_{512}$



  - 10 rounds out of 14: preimage
  - 14 rounds out of 14: chosen-salt pseudo-preimage
  - previous attacks: 8 rounds

skip details

Lesamnta (cont.)



$$X_{i+4} = X_i \oplus F\left(X_{i+1} \oplus K_{i+3}\right)$$
$$K_{i+4} = K_i \oplus G\left(K_{i+1} \oplus R_{i+3}\right).$$

- Chaining value loaded to $K_{-3}, K_{-2}, K_{-1}, K_0$
- Message loaded to $X_{-3}, X_{-2}, X_{-1}, X_0$
- F and G AES-based

Lesamnta: Truncated Differential

i	S_i	T_i	U_i	V_i
0	x	-	-	-
1	-	x	-	-
2	-	-	x	-
⋮		$(x \to x_1)$		
19	x_1	?	?	r
20	?	x_1	?	?
21	?	?	x_1	?
22	?	?	?	x_1
FF	?	?	?	x_1

i	S_i	T_i	U_i	V_i	
2	-	-	x	-	
3	y	-	-	x	$x \to y$
4	x	y	-	-	
5	-	x	y	-	
6	z	-	x	y	$y \to z$
7	-	z	-	x	$x \to y$
8	x	-	z	-	
9	w	x	-	z	$z \to w$
10	z	w	x	-	
11	x_1	z	w	x	$x \to x_1$
12	r	x_1	z	w	$w \to x \oplus r$
13	-	r	x_1	z	$z \to w$
14	?	-	r	x_1	
15	$x_1 + t$	?	-	r	$r \to t$
16	r	$x_1 + t$	?	-	
17	?	r	$x_1 + t$	?	
18	?	?	r	$x_1 + t$	
19	x_1	?	?	r	$r \to t$

Lesamnta: Truncated Differential

i	S_i	T_i	U_i	V_i
0	x	-	-	-
1	-	x	-	-
2	-	-	x	-
$\vdots$		$(x \to x_1)$		
19	x_1	?	?	r
20	?	x_1	?	?
21	?	?	x_1	?
22	?	?	?	x_1
FF	?	?	?	x_1

Properties

- Using conditions on the state, probability 1.
- The transition $x \to x_1$ is known.

How to use it

- Start with a random message
- x_1 is the difference between the output and the target value
- Compute x from x_1
- Use $M + (x, 0, 0, 0)$

Lesamnta: Truncated Differential

i	S_i	T_i	U_i	V_i
0	x	-	-	-
1	-	x	-	-
2	-	-	x	-
$\vdots$		$(x \to x_1)$		
19	x_1	?	?	r
20	?	x_1	?	?
21	?	?	x_1	?
22	?	?	?	x_1
FF	?	?	?	x_1

Properties

- Using conditions on the state, probability 1.
- The transition $x \to x_1$ is known.

How to use it

- Start with a random message
- x_1 is the difference between the output and the target value
- Compute x from x_1
- Use $M + (x, 0, 0, 0)$

Lesamnta: Values

i	$X_i (= S_i)$
-1	d
0	c
1	b
2	a
3	$F_2(c) \oplus d$
4	$F_3(b) \oplus c$
5	$F_4(a) \oplus b$
6	$F_5(F_2(c) \oplus d) \oplus a$
7	$\cancel{F_6(F_3(b) \oplus c)} \oplus \cancel{F_2(c)} \oplus d$
8	$F_7(F_4(a) \oplus b) \oplus F_3(b) \oplus c$
9	$F_8(F_5(F_2(c) \oplus d) \oplus a) \oplus F_4(a) \oplus b$
10	$F_9(d) \oplus F_5(F_2(c) \oplus d) \oplus a$
11	$F_{10}(F_7(F_4(a) \oplus b) \oplus F_3(b) \oplus c) \oplus d$
12	$F_{11}(F_8(F_5(F_2(c) \oplus d) \oplus a) \oplus F_4(a) \oplus b) \oplus F_7(F_4(a) \oplus b) \oplus F_3(b) \oplus c$
13	$\cancel{F_{12}(F_9(d) \oplus F_5(F_2(c) \oplus d) \oplus a)} \oplus \cancel{F_8(F_5(F_2(c) \oplus d) \oplus a)} \oplus F_4(a) \oplus b$
15	$F_{14}(X_{12}) \oplus F_{10}(F_7(F_4(a) \oplus b) \oplus F_3(b) \oplus c) \oplus d$
16	$F_{15}(F_4(a) \oplus b) \oplus X_{12}$
19	$\cancel{F_{18}(F_{15}(F_4(a) \oplus b) \oplus X_{12})} \oplus \cancel{F_{14}(X_{12})} \oplus F_{10}(F_7(F_4(a) \oplus b) \oplus F_3(b) \oplus c) \oplus d$

Lesamnta Cancellation Conditions

Round 7: $F_6(F_3(b) \oplus \underline{c}) \oplus F_2(\underline{c})$.
They cancel if: $F_3(b) = c_{2,6} = K_2 \oplus K_6$
i.e. $b = F_3^{-1}(K_2 \oplus K_6)$

Round 13: $F_{12}(F_9(d) \oplus \underline{F_5(F_2(c) \oplus d) \oplus a}) \oplus F_8(\underline{F_5(F_2(c) \oplus d) \oplus a})$.
They cancel if: $F_9(d) = c_{8,12} = K_8 \oplus K_{12}$
i.e. $d = F_9^{-1}(K_8 \oplus K_{12})$

Round 19: $F_{18}(F_{15}(F_4(a) \oplus b) \oplus \underline{X_{12}}) \oplus F_{14}(\underline{X_{12}})$.
They cancel if: $F_{15}(F_4(a) \oplus b) = c_{14,18} = K_{14} \oplus K_{18}$
i.e. $a = F_4^{-1}(F_{15}^{-1}(K_{14} \oplus K_{18}) \oplus b)$

22-round Attacks

- Compute a, b, d, to satisfy the cancellation conditions.
- Set the state at round 2 to (a, b, c, d).

- Express the output as a function of c
- $V_0 \quad = \eta$
  - $\eta = b \oplus F_0(a \oplus F_3(d))$
- $V_{22} = F(c \oplus \alpha) \oplus \beta$
  - $\alpha = K_{11} \oplus F_8(F_5(a) \oplus b) \oplus F_4(b)$
  - $\beta = d$

- For a target value $\overline{H}$, set $c = F^{-1}(\overline{H} \oplus \eta \oplus \beta) \oplus \alpha$
- This gives $V_0 \oplus V_{22} = \overline{H}$

22-round Attacks

- Compute a, b, d, to satisfy the cancellation conditions.
- Set the state at round 2 to (a, b, c, d).
- Express the output as a function of c
- $V_0 \quad = \eta$
  - $\eta = b \oplus F_0(a \oplus F_3(d))$
- $V_{22} = F(c \oplus \alpha) \oplus \beta$
  - $\alpha = K_{11} \oplus F_8(F_5(a) \oplus b) \oplus F_4(b)$
  - $\beta = d$
- For a target value $\overline{H}$, set $c = F^{-1}(\overline{H} \oplus \eta \oplus \beta) \oplus \alpha$
- This gives $V_0 \oplus V_{22} = \overline{H}$

22-round Attacks

- Compute a, b, d, to satisfy the cancellation conditions.
- Set the state at round 2 to (a, b, c, d).

- Express the output as a function of c
- $V_0 \;\; = \eta$
  - $\eta = b \oplus F_0(a \oplus F_3(d))$
- $V_{22} = F(c \oplus \alpha) \oplus \beta$
  - $\alpha = K_{11} \oplus F_8(F_5(a) \oplus b) \oplus F_4(b)$
  - $\beta = d$

- For a target value $\overline{H}$, set $c = F^{-1}(\overline{H} \oplus \eta \oplus \beta) \oplus \alpha$
- This gives $V_0 \oplus V_{22} = \overline{H}$

Results: Lesamnta

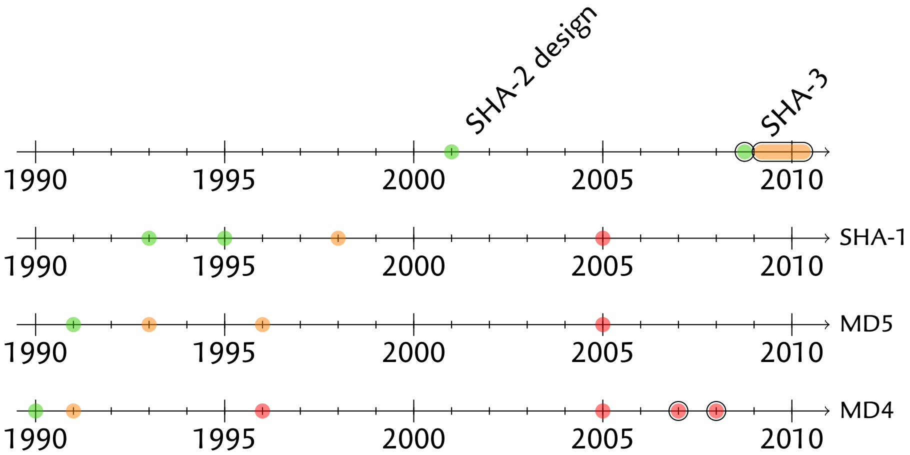
			Lesamnta-256		*Lesamnta*-512	
	Attack	Rounds	Time	Memory	Time	Memory
Generic	Collision	22	2^{96}	-	2^{192}	-
	2nd Preimage	22	2^{192}	-	2^{384}	-
	Collision	24	2^{96}	2^{64}	2^{192}	2^{128}
	2nd Preimage	24	2^{192}	2^{64}	2^{384}	2^{128}
Specific	Collision	24	2^{112}	-	2^{224}	-
	2nd Preimage	24	2^{240}	-	*N/A*	

Results: SHAvite-3_{512}

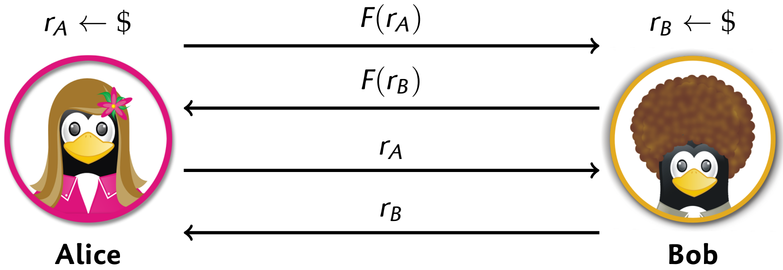
		Comp. Fun.		Hash Fun.	
Attack	Rounds	Time	Mem.	Time	Mem.
2nd Preimage	9	2^{384}	-	2^{448}	2^{64}
2nd Preimage	10	2^{448}	-	2^{480}	2^{32}
2nd Preimage	10	2^{416}	2^{64}	2^{464}	2^{64}
2nd Preimage	10	2^{384}	2^{128}	2^{448}	2^{128}
Collision[1]	14	2^{192}	2^{128}	*N/A*	
Preimage[1]	14	2^{384}	2^{128}	*N/A*	
Preimage[1]	14	2^{448}	-	*N/A*	

[1] Chosen salt attacks

Conclusion



Coin Flipping



$$\boxed{r = r_A + r_B}$$

- Alice and Bob pick each pick a random number
- They commit to it, and reveal it afterwards

One-Time-Password

$\boxed{k \leftarrow \$}$

$\boxed{y \leftarrow F^{(n)}(k)}$

Alice

$x = F^{(n-1)}(k)$ $\longrightarrow$

if $F(x) = y$
OK, $y \leftarrow x$

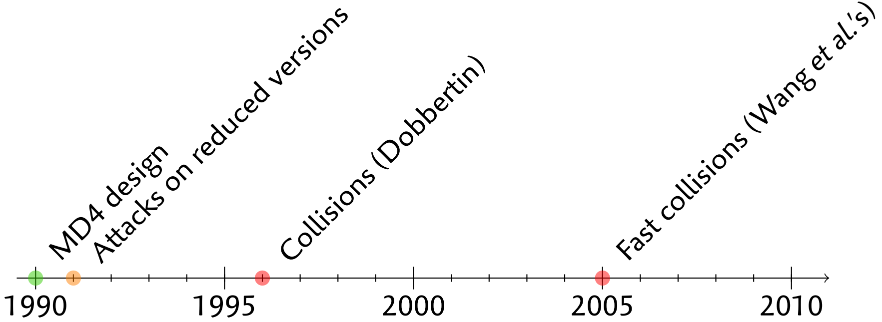
$x = F^{(n-2)}(k)$ $\longrightarrow$

if $F(x) = y$
OK, $y \leftarrow x$

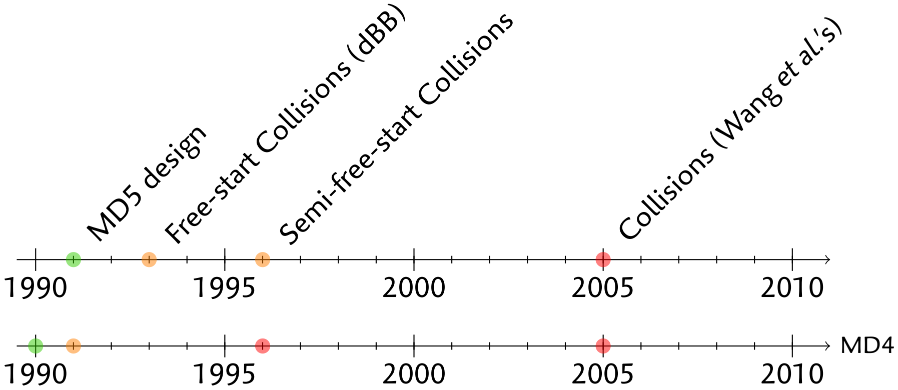
Server

- Alice choose a secret k, and the server stores $y = F^{(n)}(k)$
- For each identification, Alice send a preimage of the value stored and the server stores the new value.

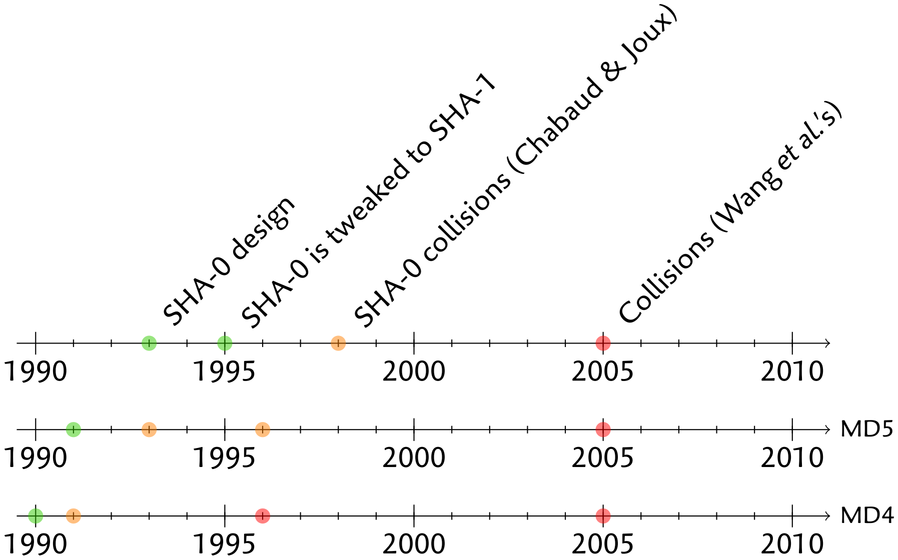
Time-line


MD4 design
Attacks on reduced versions
Collisions (Dobbertin)
Fast collisions (Wang et al.'s)
1990
1995
2000
2005
2010

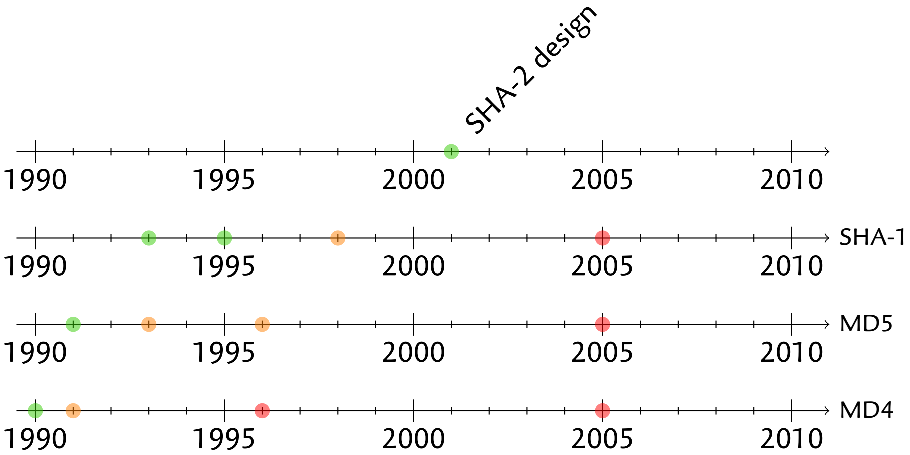
Time-line


MD5 design
Free-start Collisions (dBB)
Semi-free-start Collisions
Collisions (Wang et al.'s)
1990
1995
2000
2005
2010
MD4
1990
1995
2000
2005
2010

Time-line


SHA-0 design
SHA-0 is tweaked to SHA-1
SHA-0 collisions (Chabaud & Joux)
Collisions (Wang et al.'s)
1990
1995
2000
2005
2010
MD5
MD4

Time-line


SHA-2 design
1990
1995
2000
2005
2010
SHA-1
1990
1995
2000
2005
2010
MD5
1990
1995
2000
2005
2010
MD4
1990
1995
2000
2005
2010

Bibliography I

Message Freedom in MD4 and MD5 Collisions: Application to APOP
G. Leurent [FSE '07 + IJACT]

Automatic Search of Differential Paths in MD4
P.-A. Fouque, G. Leurent, P. Nguyen [Hash Workshop '07]

Full Key-Recovery Attacks on HMAC/NMAC-MD4 and NMAC-MD5
P.-A. Fouque, G. Leurent, P. Nguyen [Crypto '07]

MD4 is Not One-Way
G. Leurent [FSE '08]

SIMD is a Message Digest
G. Leurent, C. Bouillaguet, P.-A. Fouque [SHA-3 submission]

Security Analysis of SIMD
C. Bouillaguet, P.-A. Fouque, G. Leurent [SAC '10]

Bibliography II

- Another Look at the Complementation Property
  C. Bouillaguet, O. Dunkelman, P.-A. Fouque, G. Leurent [FSE '10]
- Cryptanalysis of ESSENCE
  M. Naya-Plasencia, A. Röck, J.-P. Aumasson, Y. Laigle-Chapuy, G. Leurent, W. Meier, T. Peyrin [FSE '10]
- Practical Key Recovery Attack against Secret-IV EDON-$\mathcal{R}$
  G. Leurent [CT-RSA '10]
- Attacks on Hash Functions based on Generalized Feistel - Application to Reduced-Round Lesamnta and *SHAvite-3*$_{512}$
  C. Bouillaguet, O. Dunkelman, P.-A. Fouque, G. Leurent [SAC '10]
- Cryptanalysis of the 10-Round Hash and Full Compression Function of SHAvite-3-512
  P. Gauravaram, G. Leurent, F. Mendel, M. Naya-Plasencia, T. Peyrin, C. Rechberger, M. Schläffer [Africacrypt '10]

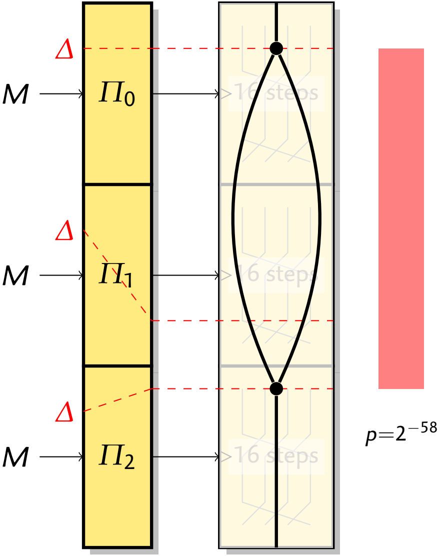
Bibliography III

Cryptanalysis of a Hash Function Based on Quasi-Cyclic Codes
P.-A. Fouque et G. Leurent [CT-RSA '08]

How risky is the Random-Oracle Model?
G. Leurent et P. Q. Nguyen [Crypto '09]

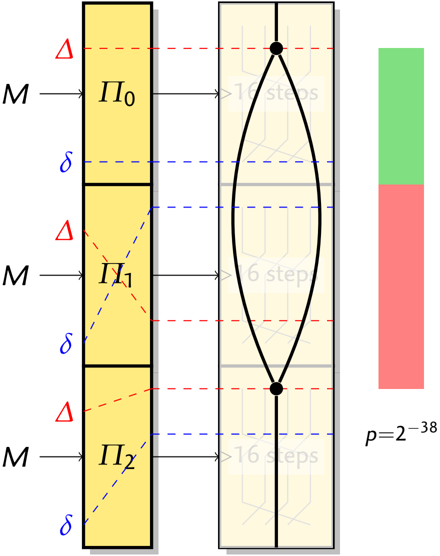
Practical Electromagnetic Template Attack on HMAC
P.-A. Fouque, G. Leurent, Denis Réal, Frédéric Valette [CHES '09]

Contini-Yin NMAC Attack



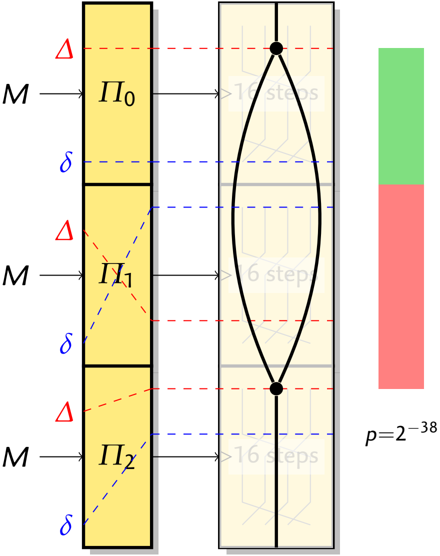
1. Find a collision:
   - Use random messages
2. Use message modifications:
   - Instead of M / $M \boxplus \Delta$, use $M \boxplus \delta$ / $M \boxplus \delta \boxplus \Delta$
   - Use carries to recover state bits

Contini-Yin NMAC Attack



1. Find a collision:
   - Use random messages
2. Use message modifications:
   - Instead of $M / M \boxplus \Delta$, use $M \boxplus \delta / M \boxplus \delta \boxplus \Delta$
   - Use carries to recover state bits

Contini-Yin NMAC Attack



1. Find a collision:
   - Use random messages
2. Use message modifications:
   - Instead of $M \,/\, M \boxplus \Delta$, use $M \boxplus \delta \,/\, M \boxplus \delta \boxplus \Delta$
   - Use carries to recover state bits

Differential paths

We need very constrained paths:

- At least one difference in m_0.
- No difference in $m_4 \dots m_{15}$.
- High probability.
- Many paths (each one gives only one bit of the key).

Differential path algorithm

- We use an algorithm to find a differential path from the message difference Δ.
- We found 22 paths with $p_X \approx 2^{-79}$.
- Attack complexity: 2^{88} data, 2^{105} time.

Differential paths

We need very constrained paths:

- At least one difference in m_0.
- No difference in $m_4 \ldots m_{15}$.
- High probability.
- Many paths (each one gives only one bit of the key).

Differential path algorithm

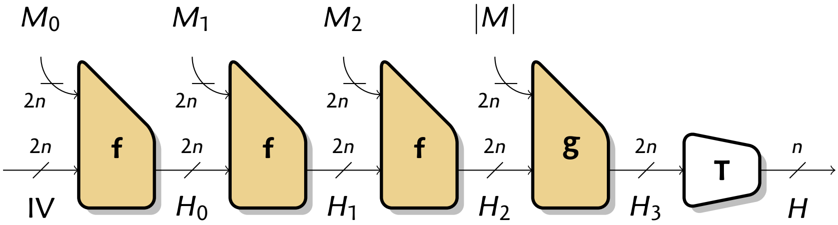
- We use an algorithm to find a differential path from the message difference Δ.
- We found 22 paths with $p_X \approx 2^{-79}$.
- Attack complexity: 2^{88} data, 2^{105} time.

Wide pipe

- Avoid generic attacks on SHA-2
  - Length-extension attack
  - MAC forgery in $2^{n/2}$
  - Multicollisions
  - Nostradamus attack (herding)
  - Second-preimage for long messages
  - Various theoretical weaknesses

- Good degradation of security:
  - Several results show that indifferentiability proofs are quite resilient
    In a wide-pipe design, indifferentiability implies all security notions.
  - Most distinguishers on the compression function
    do not weaken the iterated function.

SIMD Iteration Mode



- Finalisation function
- Use only the message length as input in the last block
  - Acts as a kind of blank round
  - Can break unexpected properties

Weaker assumptions

Strong adversary

The adversary can build an expanded message with any difference pattern

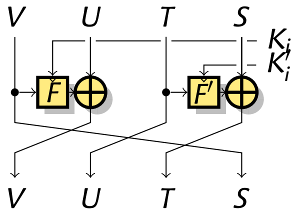
- If active state words are adjacent, some ϕ conditions disappear
  - If two inputs of the MAJ function are active we know the output
- 1 active state bit gives
  - 4.5 active message bits
  - 1 conditions
- SIMD-256: 116 conditions
- SIMD-512: 230 conditions

Modeling Differential Paths

- Impossible to have two active inputs for all active function
- Hard to proof any useful bound...
- We model the this problem as an Integer Linear Program
  - about 30,000 variables, 80,000 equations
- Solver computes a lower bound,
  and tries to improve the lower bound

  SIMD-256 $p \leq 2^{-132}$
  SIMD-512 $p \leq 2^{-253}$ (several weeks of computation)

SHAvite-3_{512}



- 14 rounds
- Davies-Meyer (message is the key)
- $F_i(x) = AES(AES(AES(AES(x \oplus k_i^0) \oplus k_i^1) \oplus k_i^2) \oplus k_i^3)$
- F is one AES round.
- Key schedule mixes linear operations and AES rounds.

SHAvite-3_{512}: Truncated Differential

i	S_i	T_i	U_i	V_i	
0	?	x_2	?	x	
1	x	-	x_2	x_1	
2	x_1	x	-	-	$x_1 \to x_2$
3	-	-	x	-	$x \to x_1$
4	-	-	-	x	
5	x	-	-	y	$x \to y$
6	y	x	-	z	$y \to z$
7	z	-	x	w	$x \to y$, $z \to w$
8	w	z	-	?	
9	?	-	z	?	$z \to w$
FF	?	x_2	?	?	

Properties

- Using conditions on the state, probability 1.
- The transitions $x \to x_1$ and $x_1 \to x_2$ are known.
- Compute x from x_2

Problem

- F has many keys

SHAvite-3_{512}: Truncated Differential

i	S_i	T_i	U_i	V_i	
0	?	x_2	?	x	
1	x	-	x_2	x_1	
2	x_1	x	-	-	$x_1 \to x_2$
3	-	-	x	-	$x \to x_1$
4	-	-	-	x	
5	x	-	-	y	$x \to y$
6	y	x	-	z	$y \to z$
7	z	-	x	w	$x \to y$, $z \to w$
8	w	z	-	?	
9	?	-	z	?	$z \to w$
FF	?	x_2	?	?	

Properties

- Using conditions on the state, probability 1.
- The transitions $x \to x_1$ and $x_1 \to x_2$ are known.
- Compute x from x_2

Problem

- F has many keys

SHAvite-3_{512}: Values

i	X_i / Y_i
X_0	$b \oplus F_3(c) \oplus F'_1(c \oplus F_2(d \oplus F'_3(a)))$
Y_0	$d \oplus F'_3(a) \oplus F_1(a \oplus F'_2(b \oplus F_3(c)))$
X_1	$a \oplus F'_2(b \oplus F_3(c))$
Y_1	$c \oplus F_2(d \oplus F'_3(a))$
X_2	$d \oplus F'_3(a)$
Y_2	$b \oplus F_3(c)$
X_3	c
Y_3	a
X_4	b
Y_4	d
X_5	$a \oplus F_4(b)$
Y_5	$c \oplus F'_4(d)$
X_6	$d \oplus F_5(a \oplus F_4(b))$
Y_6	$b \oplus F'_5(c \oplus F'_4(d))$
X_7	$c \oplus \cancel{F'_4(d)} \oplus \cancel{F_6(d \oplus F_5(a \oplus F_4(b)))}$
Y_7	$a \oplus F_4(b) \oplus F'_6(b \oplus F'_5(c \oplus F'_4(d)))$
X_8	$b \oplus F'_5(c \oplus F'_4(d)) \oplus F_7(c)$
Y_8	$d \oplus F_5(a \oplus F_4(b)) \oplus F'_7(a \oplus F_4(b) \oplus F'_6(b \oplus F'_5(c \oplus F'_4(d))))$
X_9	$a \oplus F_4(b) \oplus \cancel{F'_6(b \oplus F'_5(c \oplus F'_4(d)))} \oplus \cancel{F_8(b \oplus F'_5(c \oplus F'_4(d)) \oplus F_7(c))}$

Message Conditions: SHAvite-3_{512}

Round 7 $F'_4(\underline{d}) \oplus F_6(\underline{d} \oplus F_5(a \oplus F_4(b)))$.
They cancel if: $F_5(a \oplus F_4(b)) = k^0_{1,4} \oplus k^0_{0,6}$
and $(k^1_{1,4}, k^2_{1,4}, k^3_{1,4}) = (k^1_{0,6}, k^2_{0,6}, k^3_{0,6})$.

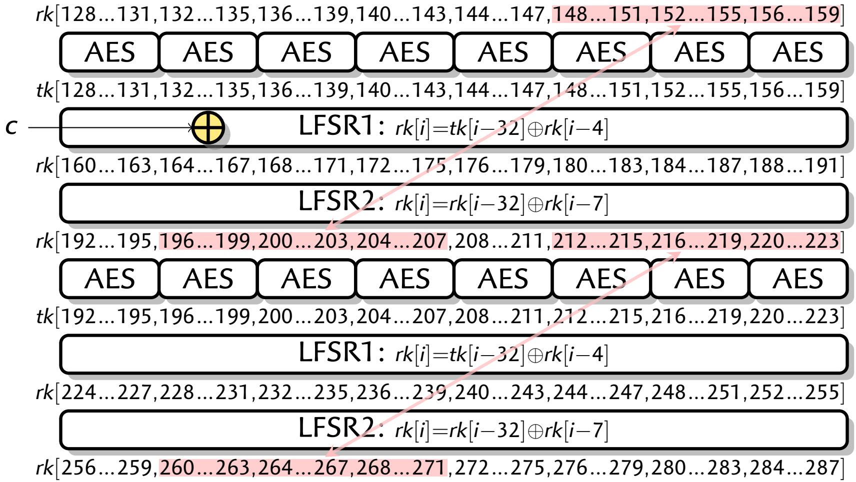
Round 9 $F'_6(\underline{b \oplus F'_5(c \oplus F'_4(d))}) \oplus F_8(\underline{b \oplus F'_5(c \oplus F'_4(d))} \oplus F_7(c))$.
They cancel if: $F_7(c) = k^0_{1,6} \oplus k^0_{0,8}$
and $(k^1_{1,6}, k^2_{1,6}, k^3_{1,6}) = (k^1_{0,8}, k^2_{0,8}, k^3_{0,8})$.

Message Conditions: SHAvite-3_{512}

Round 7 $F'_4(\underline{d}) \oplus F_6(\underline{d} \oplus F_5(a \oplus F_4(b)))$.
They cancel if: $F_5(a \oplus F_4(b)) = k^0_{1,4} \oplus k^0_{0,6}$
and $(k^1_{1,4}, k^2_{1,4}, k^3_{1,4}) = (k^1_{0,6}, k^2_{0,6}, k^3_{0,6})$.

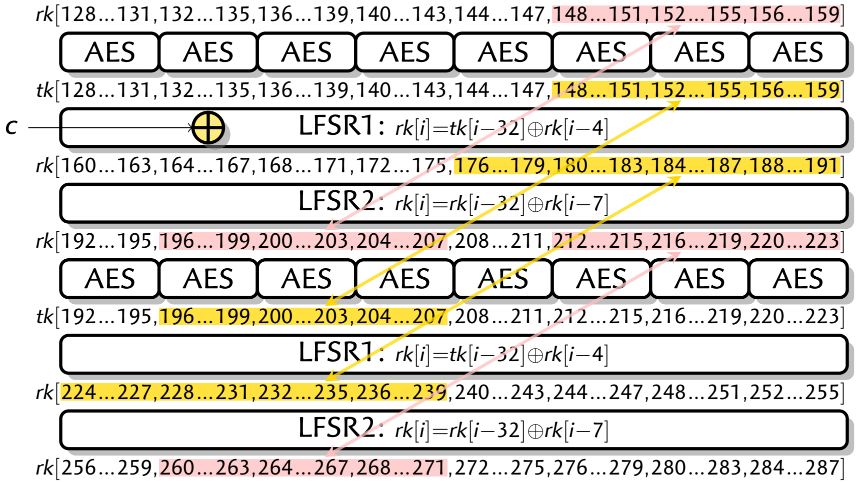
Round 9 $F'_6(\underline{b \oplus F'_5(c \oplus F'_4(d))}) \oplus F_8(\underline{b \oplus F'_5(c \oplus F'_4(d))} \oplus F_7(c))$.
They cancel if: $F_7(c) = k^0_{1,6} \oplus k^0_{0,8}$
and $(k^1_{1,6}, k^2_{1,6}, k^3_{1,6}) = (k^1_{0,8}, k^2_{0,8}, k^3_{0,8})$.

Message Expansion



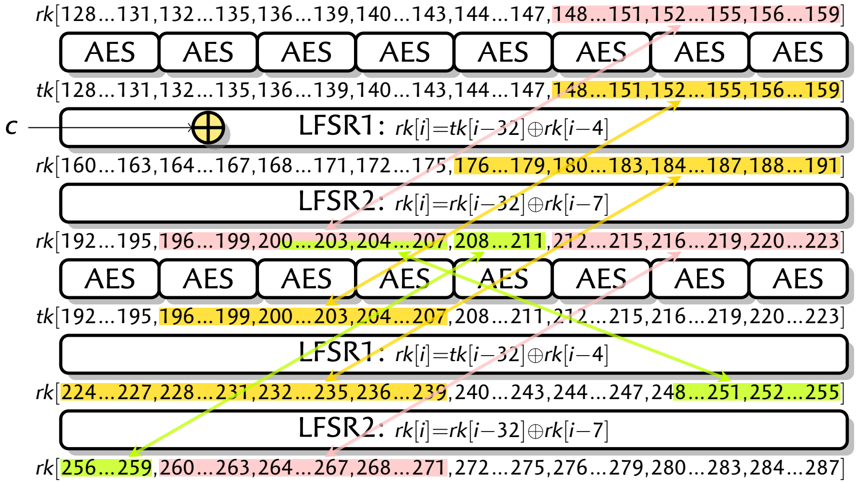
1. Propagate constraints

Message Expansion



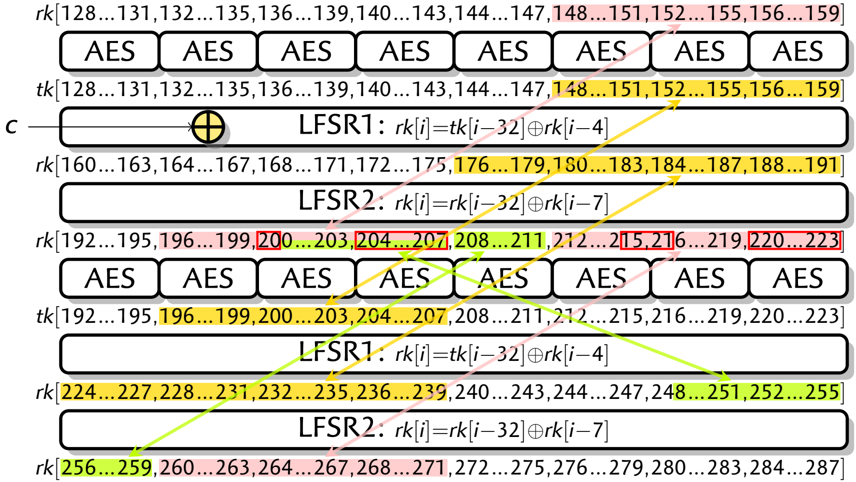
1. Propagate constraints

Message Expansion



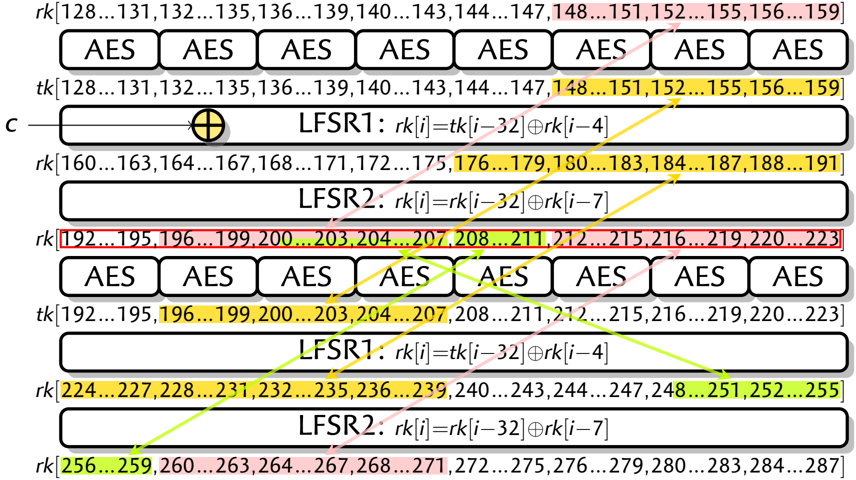
1. Propagate constraints

Message Expansion



2 Guess values

Message Expansion



3 Compute the missing values; check coherence

Solving the Conditions

- We can build a chaining value satisfying the 6 conditions with cost 2^{96}.
- Each chaining value can be used 2^{128} times to fix 128 bits of the output.
  - Cost of finding a good message is amortized.
- Attacks on 9-round *SHAvite-3_{512}*:
  - Free-start preimage with complexity 2^{384}
  - Second-Preimage with complexity 2^{448}.

Extension to 10 rounds

- We only use two cancellation: two conditions on the state
- We still have two degrees of freedom

- We change the variables, with u, v degrees of freedom, and z, w fixed by the cancellation conditions

$$H = z \oplus F_2(u) \oplus F'_0(u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))) \oplus w$$

$$z \oplus F_2(u) \oplus F'_0(u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))) \oplus w = H$$

$$F'_0(u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))) = H \oplus z \oplus F_2(u) \oplus w$$

$$F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z))) = u \oplus F'^{-1}_0(H \oplus z \oplus F_2(u) \oplus w)$$

- We have u on one side and v on the other side: we can solve with cost 2^{64} by the birthday paradox

Extension to 10 rounds

- We only use two cancellation: two conditions on the state
- We still have two degrees of freedom

- We change the variables, with u, v degrees of freedom, and z, w fixed by the cancellation conditions

$$H = z \oplus F_2(u) \oplus F'_0(u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))) \oplus w$$
$$z \oplus F_2(u) \oplus F'_0(u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))) \oplus w = H$$
$$F'_0(u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))) = H \oplus z \oplus F_2(u) \oplus w$$
$$F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z))) = u \oplus F_0'^{-1}(H \oplus z \oplus F_2(u) \oplus w)$$

- We have u on one side and v on the other side: we can solve with cost 2^{64} by the birthday paradox

Extension to 10 rounds

- We only use two cancellation: two conditions on the state
- We still have two degrees of freedom

- We change the variables, with u, v degrees of freedom, and z, w fixed by the cancellation conditions

$$H = z \oplus F_2(u) \oplus F_0'(u \oplus F_1(w \oplus F_4(v) \oplus F_2'(v \oplus F_3(z)))) \oplus w$$

$$z \oplus F_2(u) \oplus F_0'(u \oplus F_1(w \oplus F_4(v) \oplus F_2'(v \oplus F_3(z)))) \oplus w = H$$

$$F_0'(u \oplus F_1(w \oplus F_4(v) \oplus F_2'(v \oplus F_3(z)))) = H \oplus z \oplus F_2(u) \oplus w$$

$$F_1(w \oplus F_4(v) \oplus F_2'(v \oplus F_3(z))) = u \oplus F_0'^{-1}(H \oplus z \oplus F_2(u) \oplus w)$$

- We have u on one side and v on the other side: we can solve with cost 2^{64} by the birthday paradox

Extension to 14 rounds

i	X_i	Y_i
0	$v \oplus F_3(z) \oplus F'_1(z \oplus F_2(u))$	$u \oplus F_1(w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z)))$
1	$w \oplus F_4(v) \oplus F'_2(v \oplus F_3(z))$	$z \oplus F_2(u)$
2	u	$v \oplus F_3(z)$
3	z	$w \oplus F_4(v)$
4	v	$u \oplus F'_3(Y_3)$
5	w	$z \oplus F'_4(Y_4)$
6	$Y_4 \oplus F_5(w)$	$v \oplus F'_5(Y_5)$
7	$z \oplus \cancel{F'_4(\underline{Y_4})} \oplus \cancel{F_6(\underline{Y_4} \oplus F_5(w))}$	$w \oplus F'_6(Y_6)$
8	$Y_6 \oplus F_7(z)$	$Y_4 \oplus F_5(w) \oplus F'_7(Y_7)$
9	$w \oplus \cancel{F'_6(\underline{Y_6})} \oplus \cancel{F_8(\underline{Y_6} \oplus F_7(z))}$	$z \oplus F'_8(Y_8)$
10	$Y_8 \oplus F_9(w)$	$Y_6 \oplus F_7(z) \oplus F'_9(Y_9)$
11	$z \oplus \cancel{F'_8(\underline{Y_8})} \oplus \cancel{F_{10}(\underline{Y_8} \oplus F_9(w))}$	$w \oplus F'_{10}(Y_{10})$
12	$Y_{10} \oplus F_{11}(z)$	$Y_8 \oplus F_9(w) \oplus F'_{11}(Y_{11})$
13	$w \oplus \cancel{F'_{10}(\underline{Y_{10}})} \oplus \cancel{F_{12}(\underline{Y_{10}} \oplus F_{11}(z))}$	$z \oplus F'_{12}(Y_{12})$

14-round cancellation conditions

Round 7 $\underline{F'_4(Y_4)} \oplus F_6(\underline{Y_4 \oplus} F_5(w))$.
They cancel if: $F_5(w) = k^0_{1,4} \oplus k^0_{0,6}$ and
$(k^1_{1,4}, k^2_{1,4}, k^3_{1,4}) = (k^1_{0,6}, k^2_{0,6}, k^3_{0,6})$.

Round 9 $\underline{F'_6(Y_6)} \oplus F_8(\underline{Y_6 \oplus} F_7(z))$.
They cancel if: $F_7(z) = k^0_{1,6} \oplus k^0_{0,8}$ and
$(k^1_{1,6}, k^2_{1,6}, k^3_{1,6}) = (k^1_{0,8}, k^2_{0,8}, k^3_{0,8})$.

Round 11 $\underline{F'_8(Y_8)} \oplus F_{10}(\underline{Y_8 \oplus} F_9(w))$.
They cancel if: $F_9(w) = k^0_{1,8} \oplus k^0_{0,10}$ and
$(k^1_{1,8}, k^2_{1,8}, k^3_{1,8}) = (k^1_{0,10}, k^2_{0,10}, k^3_{0,10})$.
Since w is fixed: $F_5^{-1}(k^0_{1,4} \oplus k^0_{0,6}) = F_9^{-1}(k^0_{1,8} \oplus k^0_{0,10})$

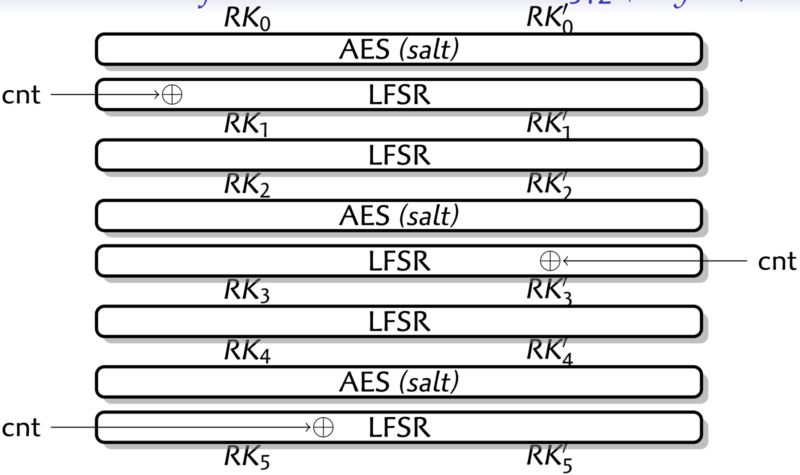
Round 13 $\underline{F'_{10}(Y_{10})} \oplus F_{12}(\underline{Y_{10} \oplus} F_{11}(z))$.
They cancel if: $F_{11}(z) = k^0_{1,10} \oplus k^0_{0,12}$ and
$(k^1_{1,10}, k^2_{1,10}, k^3_{1,10}) = (k^1_{0,12}, k^2_{0,12}, k^3_{0,12})$.
Since z is fixed: $F_7^{-1}(k^0_{1,6} \oplus k^0_{0,8}) = F_{11}^{-1}(k^0_{1,10} \oplus k^0_{0,12})$

14-round cancellation conditions

- 256-bit condition on the state
- 1792-bit condition on the expanded message

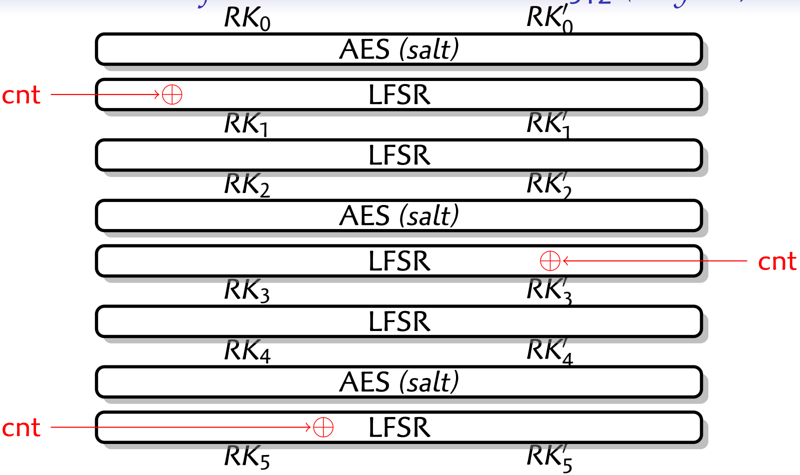
$$
\begin{aligned}
F_5(w) &= k^0_{1,4} \oplus k^0_{0,6} \\
F_7(z) &= k^0_{1,6} \oplus k^0_{0,8} \\
(k^1_{1,4}, k^2_{1,4}, k^3_{1,4}) &= (k^1_{0,6}, k^2_{0,6}, k^3_{0,6}) \\
(k^1_{1,6}, k^2_{1,6}, k^3_{1,6}) &= (k^1_{0,8}, k^2_{0,8}, k^3_{0,8}) \\
(k^1_{1,8}, k^2_{1,8}, k^3_{1,8}) &= (k^1_{0,10}, k^2_{0,10}, k^3_{0,10}) \\
(k^1_{1,10}, k^2_{1,10}, k^3_{1,10}) &= (k^1_{0,12}, k^2_{0,12}, k^3_{0,12}) \\
F_5^{-1}(k^0_{1,4} \oplus k^0_{0,6}) &= F_9^{-1}(k^0_{1,8} \oplus k^0_{0,10}) \\
F_7^{-1}(k^0_{1,6} \oplus k^0_{0,8}) &= F_{11}^{-1}(k^0_{1,10} \oplus k^0_{0,12})
\end{aligned}
$$

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



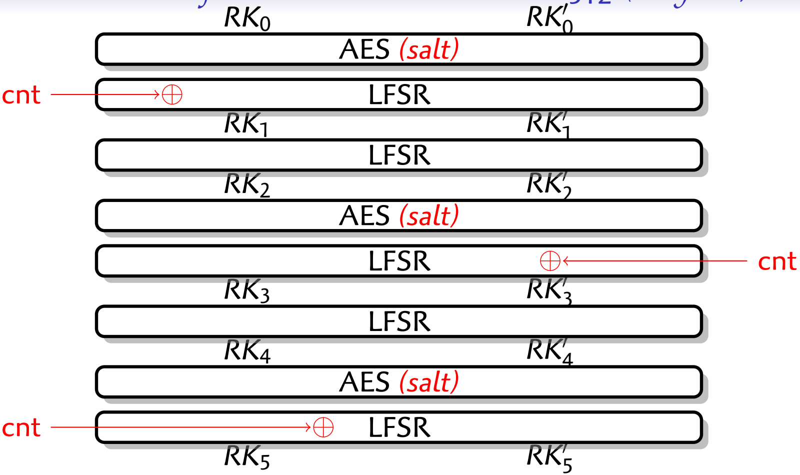
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



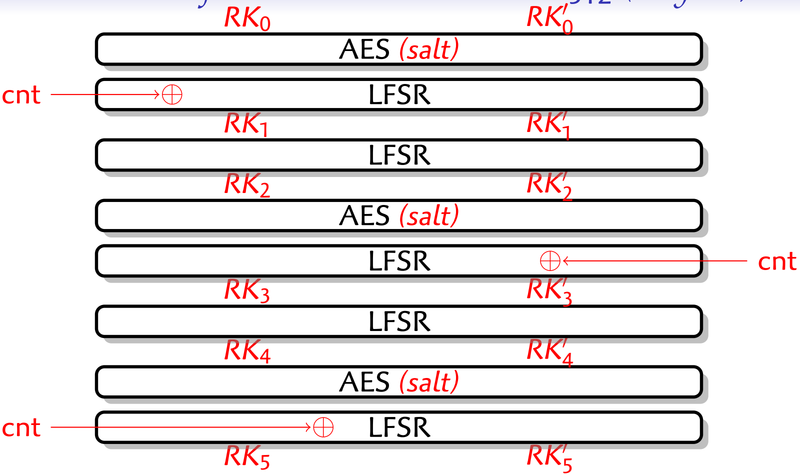
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



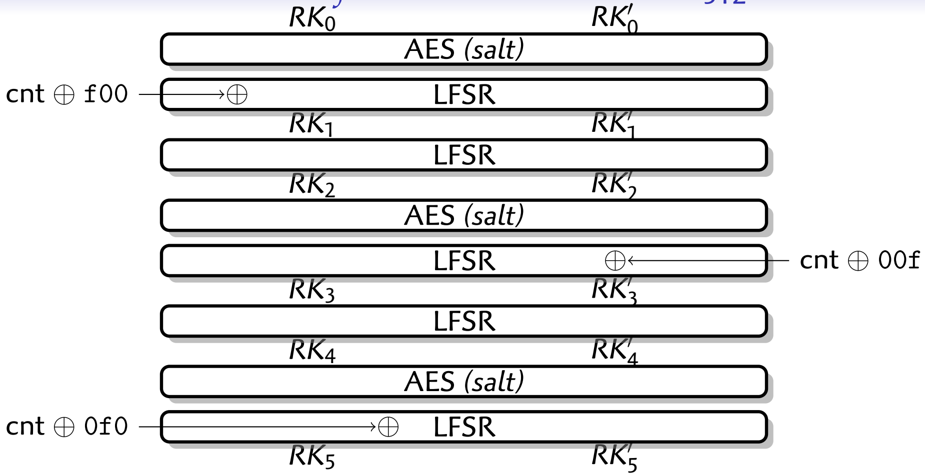
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



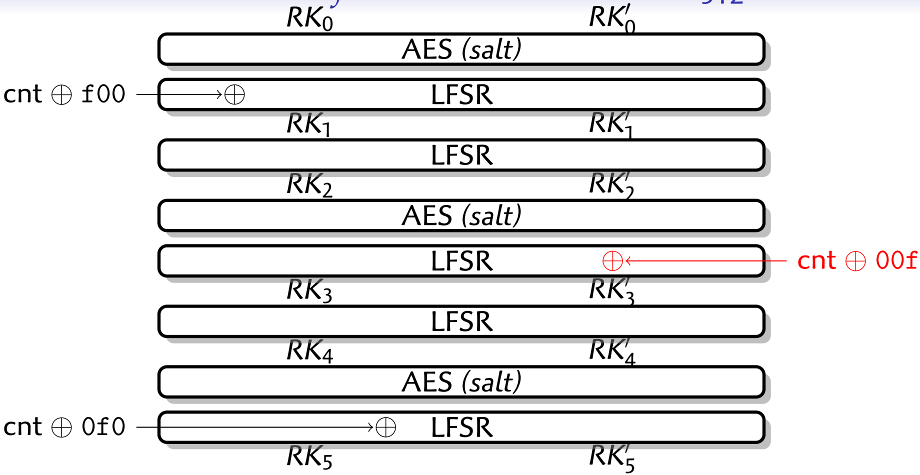
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-2 SHAvite-3_{512}



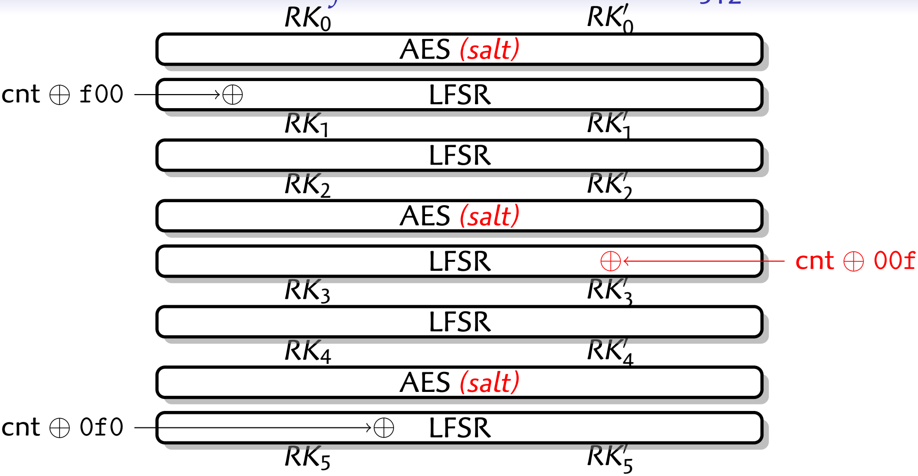
- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}



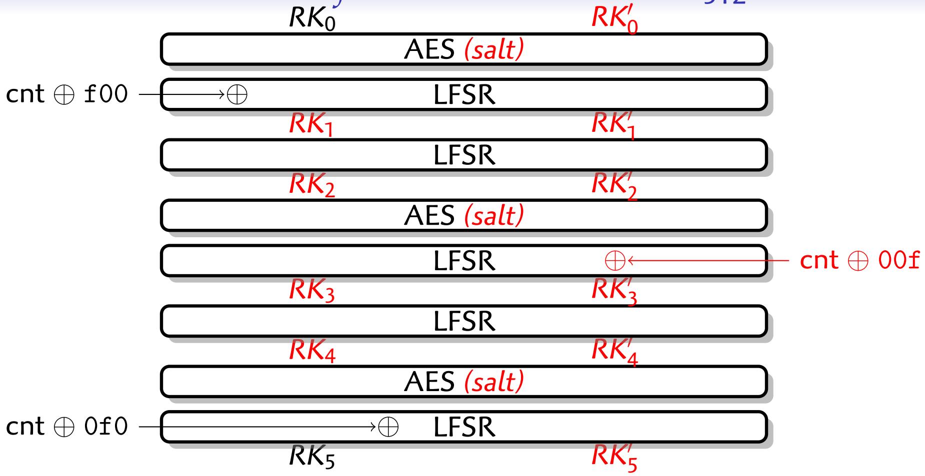
- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}



- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}



- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}

i	RK_i				RK'_i				r
	$k^0_{0,i}$	$k^1_{0,i}$	$k^2_{0,i}$	$k^3_{0,i}$	$k^0_{1,i}$	$k^1_{1,i}$	$k^2_{1,i}$	$k^3_{1,i}$	
0	?	?	?	?	?	?	?	?	M
1	?★	?	?	?	?	?	?	0	1
2	0	?	?	?	?	0	0	0	
3	0	?	?	?	0	0	0	0	2
4	0	?	0	0	0	0	0	0	
5	0	0★	0	0	0	0	0	0	3
6	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	4
8	0	0	0	0	0	0	0	0	
9	0	0	0	0★	0	0	0	0	5
10	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	6
12	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	?★	?	7