Cryptanalysis of Alternative Hash Functions

Florian Mendel

http://www.iaik.tugraz.at/aboutus/people/mendel/index.php

Hash&Stream, Salzburg, 2007/02/02

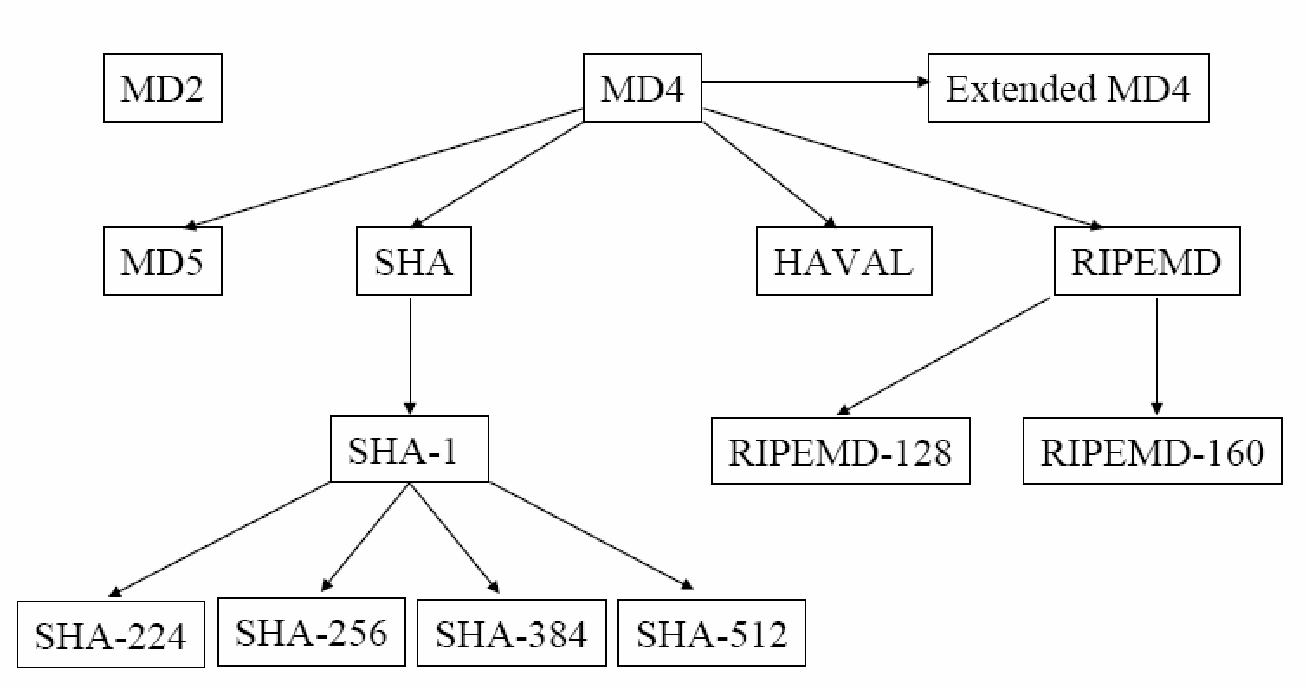
Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology

Outline

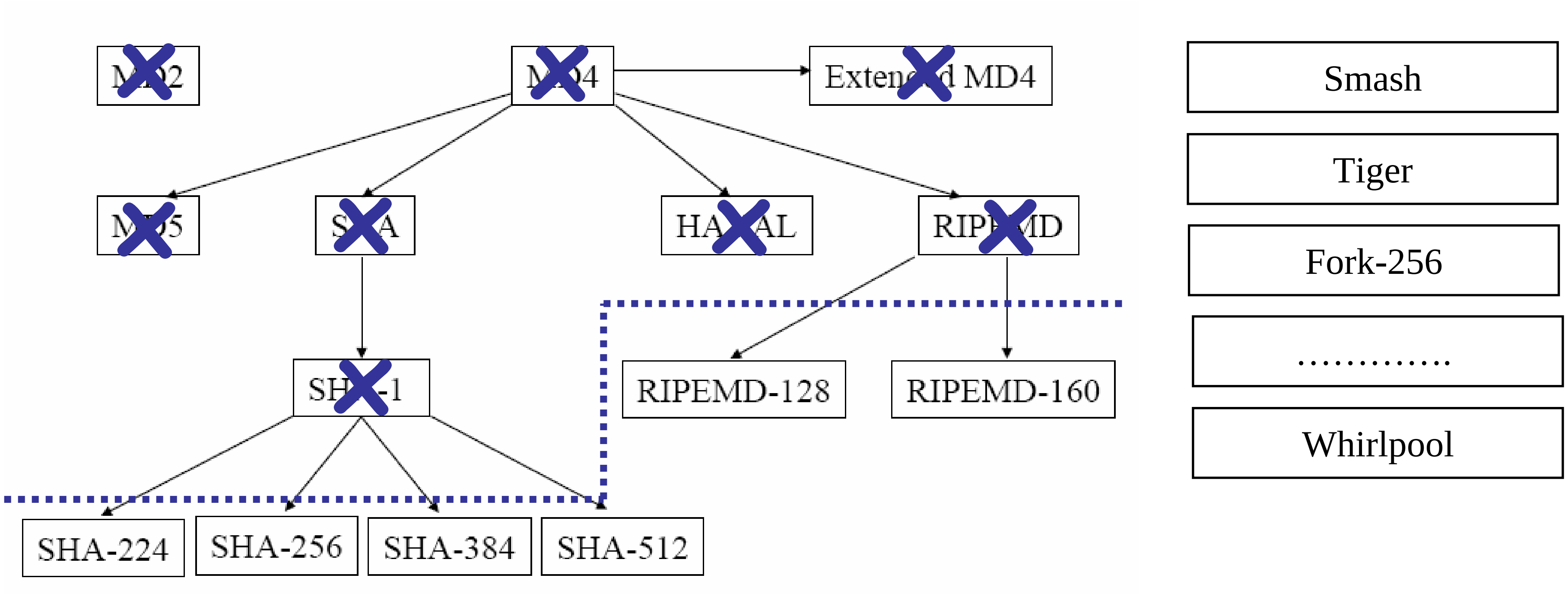
- Motivation
- Cryptanalysis of
  - SHA-256
  - RIPEMD-160 and RIPEMD-128
  - Smash
  - Tiger
- Conclusion

Motivation

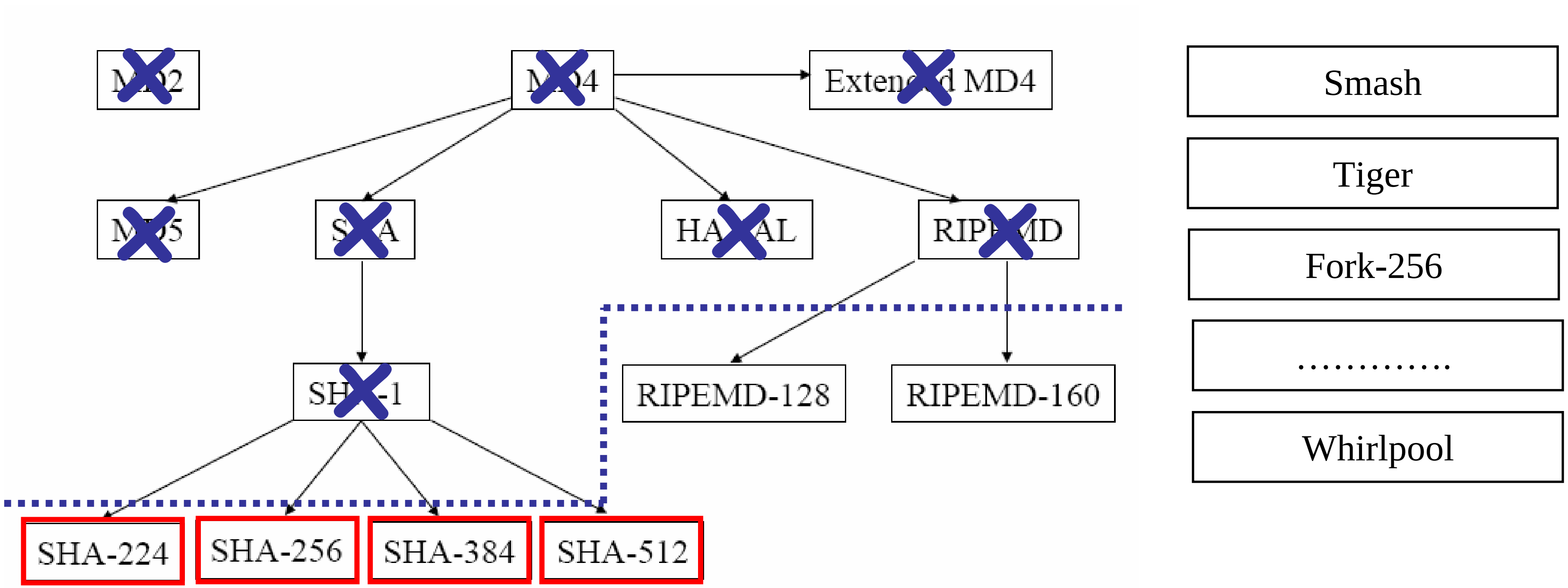

MD2
MD4
Extended MD4
MD5
SHA
HAVAL
RIPEMD
SHA-1
RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512


Smash
Tiger
Fork-256
………….
Whirlpool

Motivation


RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
………….
Whirlpool

Motivation


RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
....
Whirlpool

Analysis of Step-Reduced SHA-256

Florian Mendel and Norbert Pramstaller and
Christian Rechberger and Vincent Rijmen

presented at FSE 2006

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology

SHA-256 is Interesting and Challenging

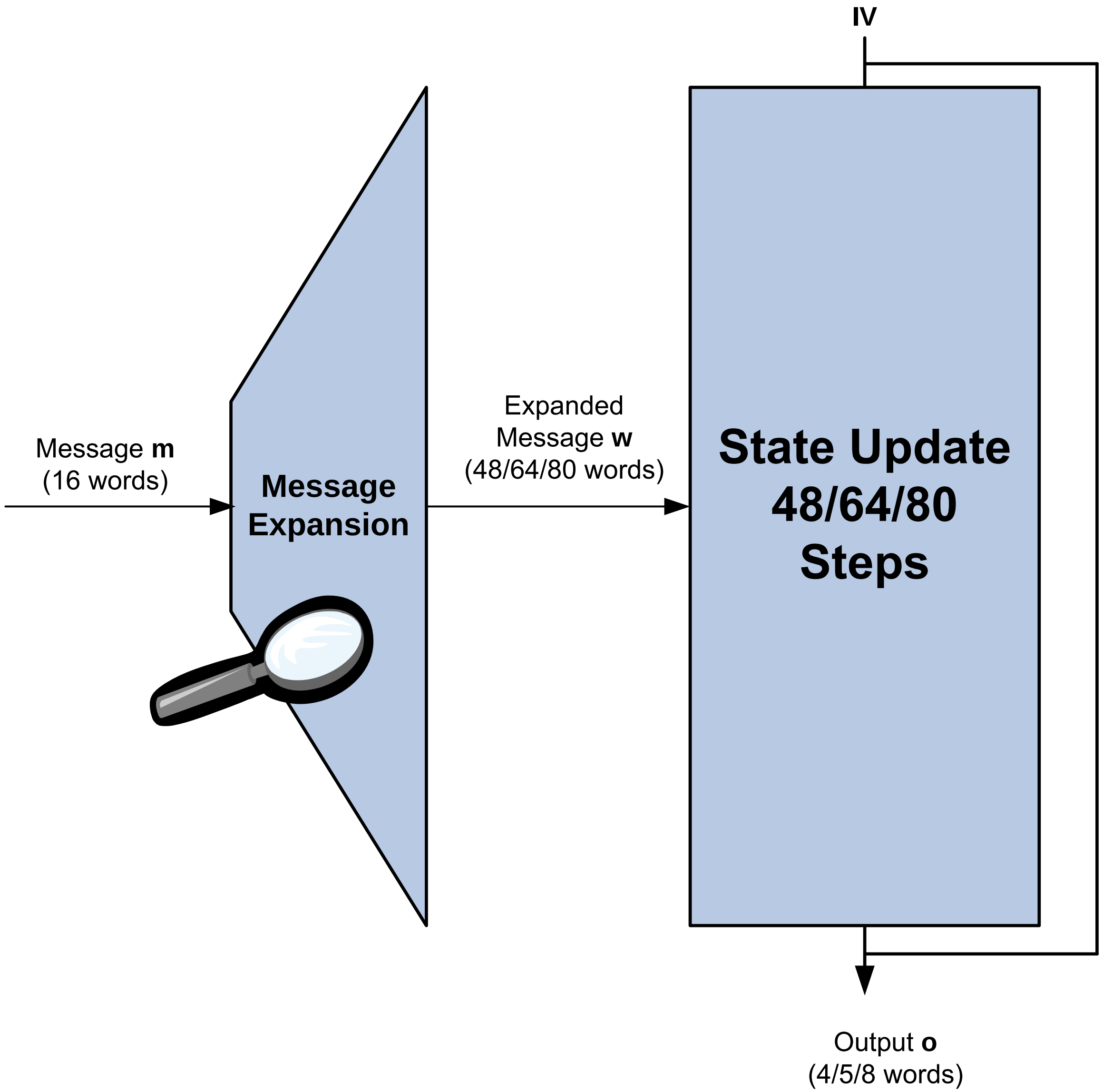
FIPS Standard since 2002

Option for a SHA-1 upgrade

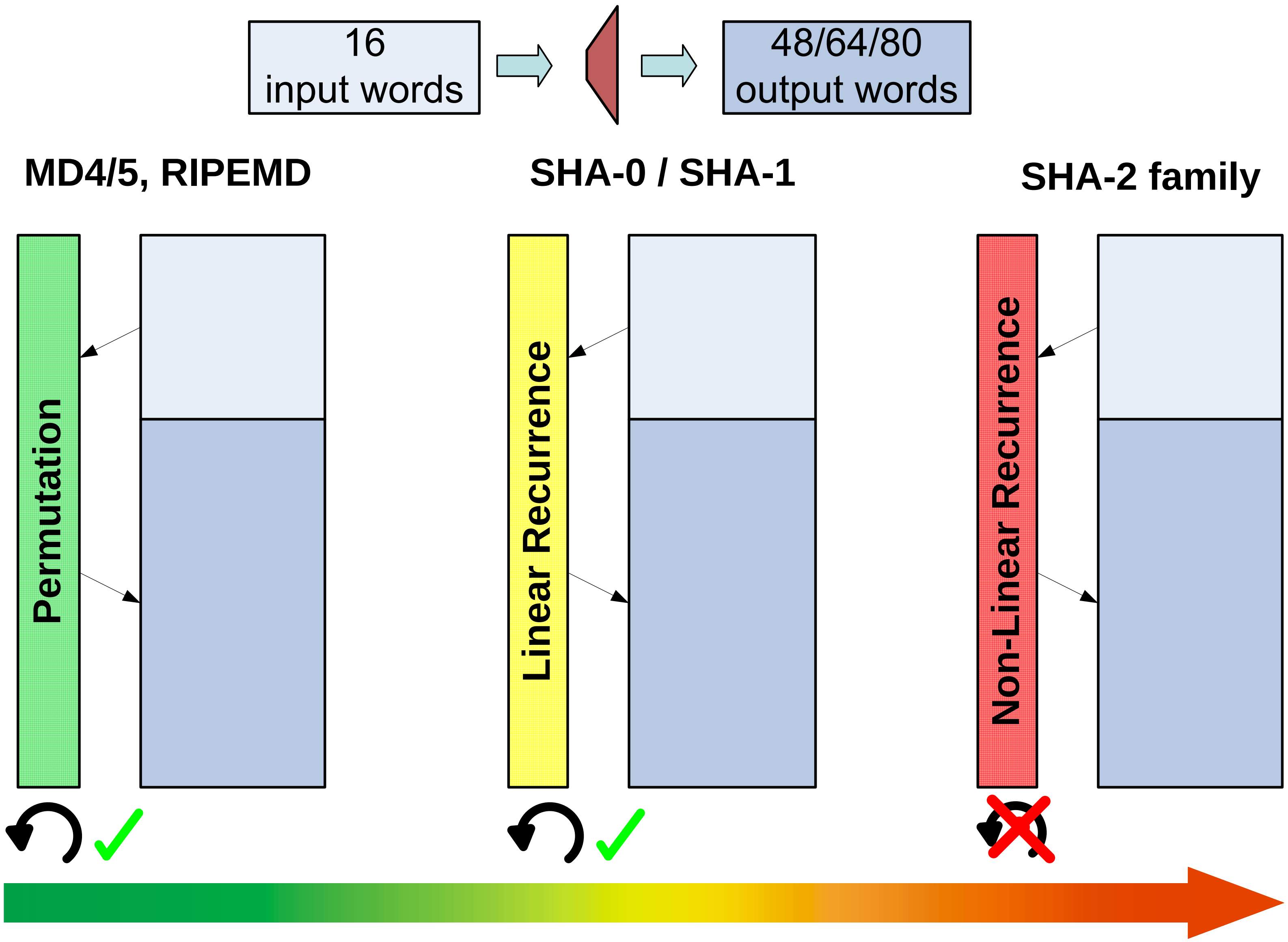
Prudent to know:

How hard is it to find collisions for SHA-256?
What about step-reduced variants (security margin)?

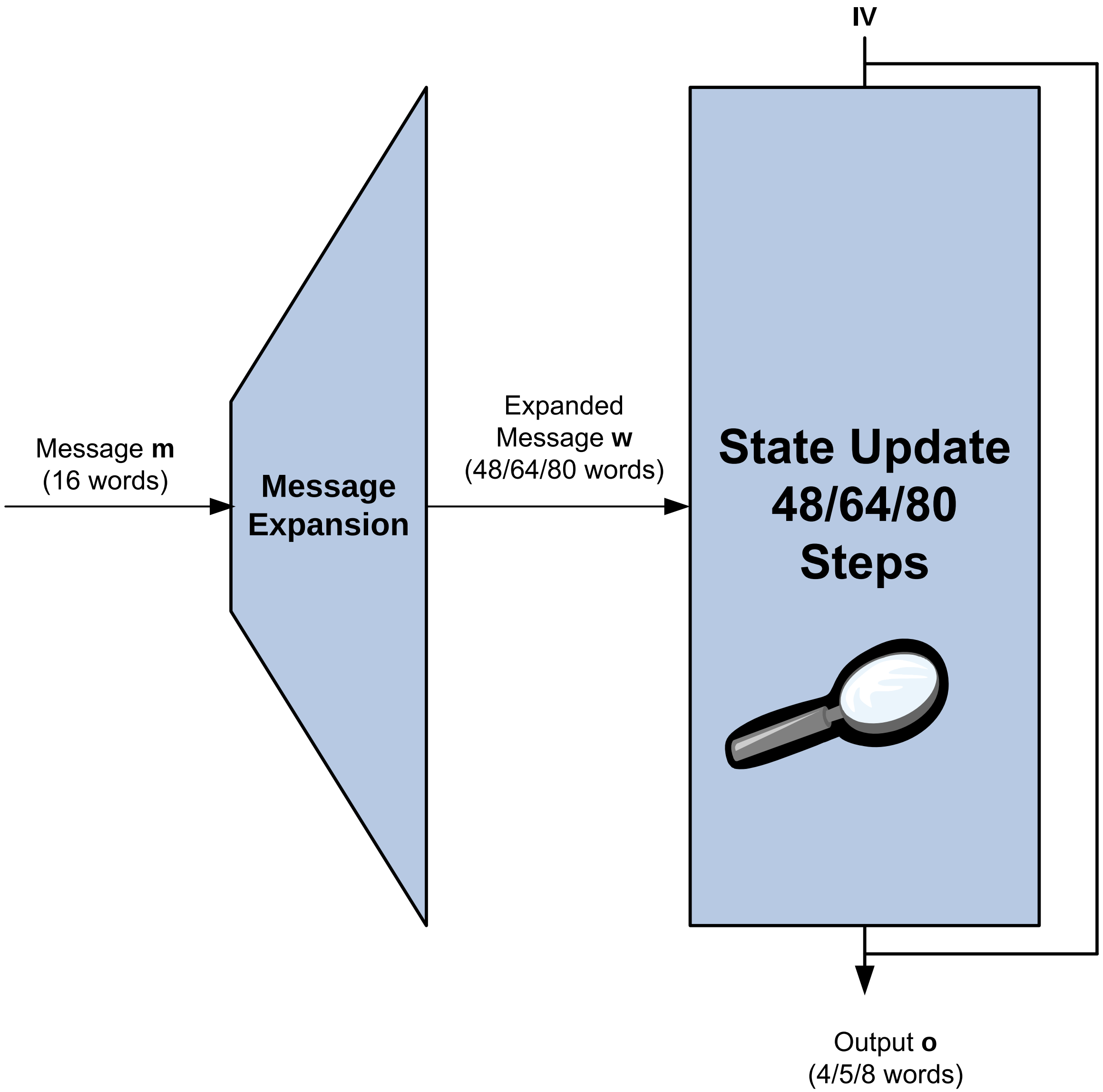
Outline of MD4-style Hash Functions



Message Expansions in the MD4 family

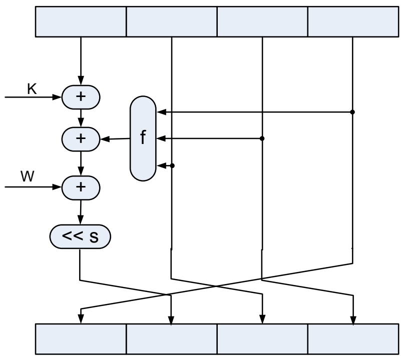


Outline of MD4-style Hash Functions

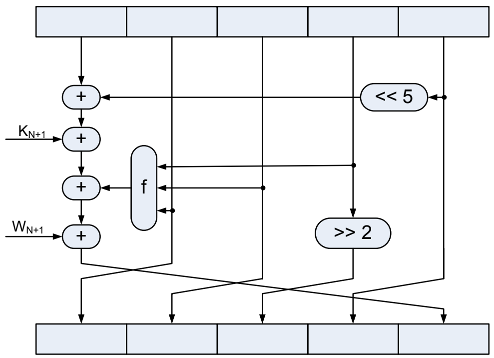


Evolution of the State Updates in the MD4 Family

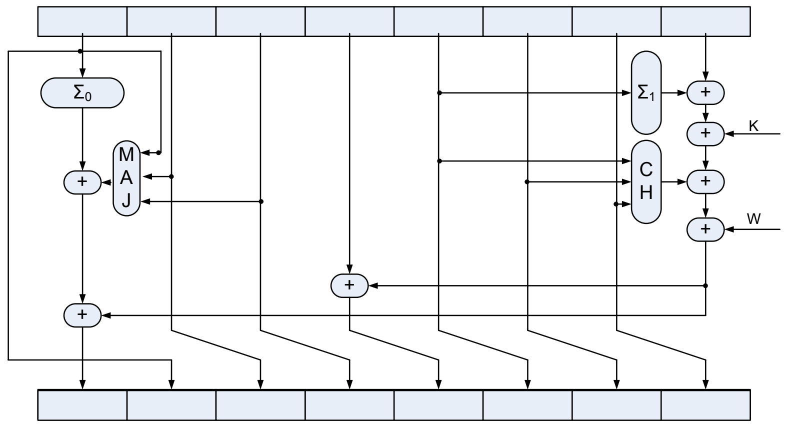
MD4



SHA-0/1

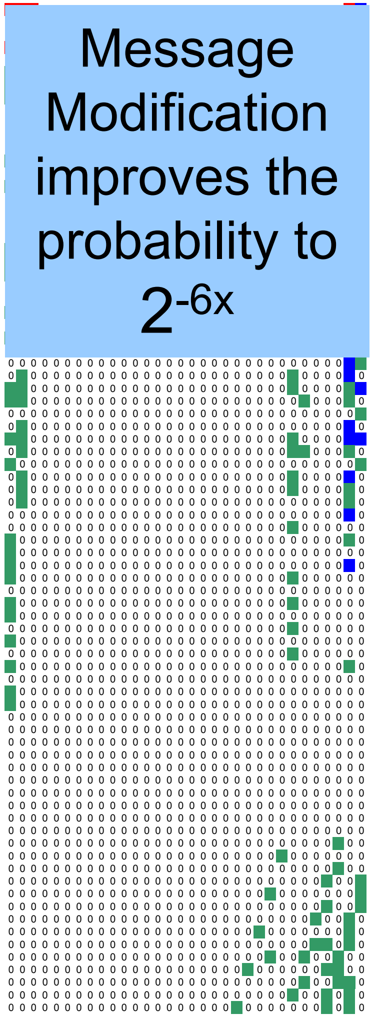


SHA-2 family



Design Complexity

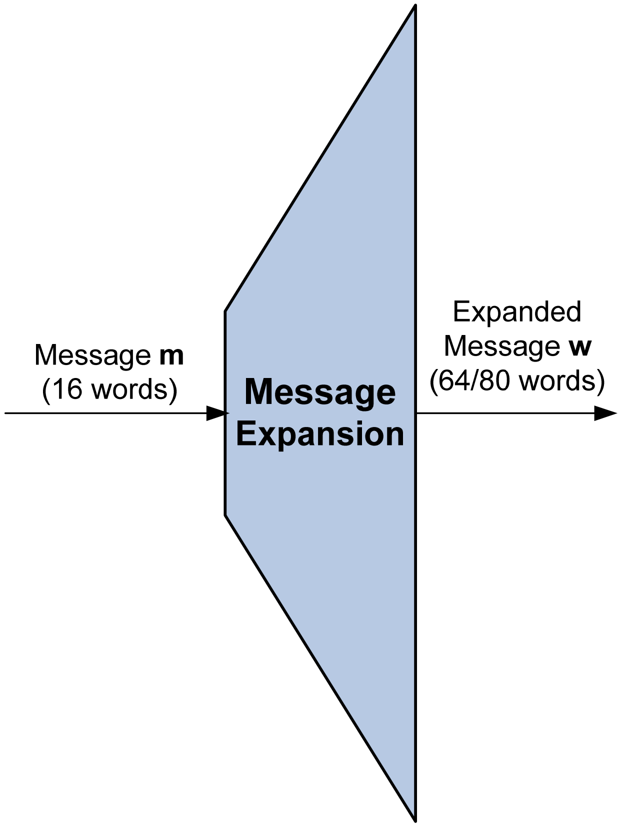
Attack of Wang *etal.* On SHA-1



Low-probability characteristic

High-probability characteristic (about 2^{-83})

Comparison of SHA Message Expansions



SHA-1

$$W_t = \begin{cases} M_t & for\,(0 \le t \le 15) \\ ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & for\,(16 \le t \le 79) \end{cases}$$

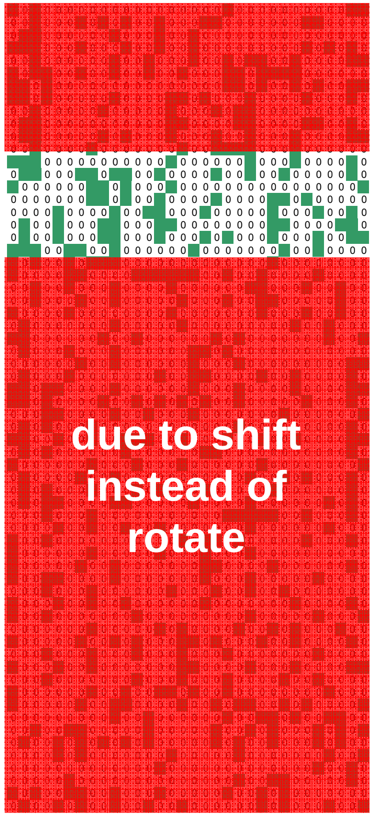
SHA-256

$$W_t = \begin{cases} M_t & for\,(0 \le t \le 15) \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} & for\,(16 \le t \le 63) \end{cases}$$

$$\sigma_0(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$$

$$\sigma_1(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$$

Approach does not apply to SHA-2



Low-probability characteristic

~~High-probability characteristic~~

Low-probability characteristic

Example of 19-step Characteristic

Step	W'	A'	B'	C'	D'	E'	F'	G'	H'
1-4	0	0	0	0	0	0	0	0	0
05	85009008	85009008	0	0	0	85009008	0	0	0
06	a14cae12	a1442610	85009008	0	0	02000802	85009008	0	0
07	0	0	a1442610	85009008	0	084c4120	02000802	85009008	0
08	8200a8a8	00000020	0	a1442610	85009008	00000020	084c4120	02000802	85009008
09	85009008	85009008	00000020	0	a1442610	01008008	00000020	084c4120	02000802
10	0	0	85009008	00000020	0	02000802	01008008	00000020	084c4120
11	0	0	0	85009008	00000020	0	02000802	01008008	00000020
12	0	00000020	0	0	85009008	0	0	02000802	01008008
13	0	0	00000020	0	0	84001000	0	0	02000802
14	00088802	0	0	00000020	0	0	84001000	0	0
15	0	0	0	0	00000020	0	0	84001000	0
16	0	0	0	0	0	00000020	0	0	84001000
17	0	0	0	0	0	0	00000020	0	0
18	0	0	0	0	0	0	0	00000020	0
19	0	0	0	0	0	0	0	0	00000020

collision for SHA-224

Interesting Results

- Perturbation pattern is no valid expanded message
  - But the sum of perturbations and corrections is
- More freedom for the carry
  - … to prevent contradictions in characteristics
- The overall probability is much higher than the product of the probabilities of each individual local collision
  - Different to SHA-0 / SHA-1
  - Example: low-weight 19-step characteristic
    - 23 local collisions of probability around 2^{-40}
    - Total probability is much higher: instead of 2^{-920} around 2^{-200}

    (Compare this to a similar probability of the best known 80-step characteristic for SHA-1)

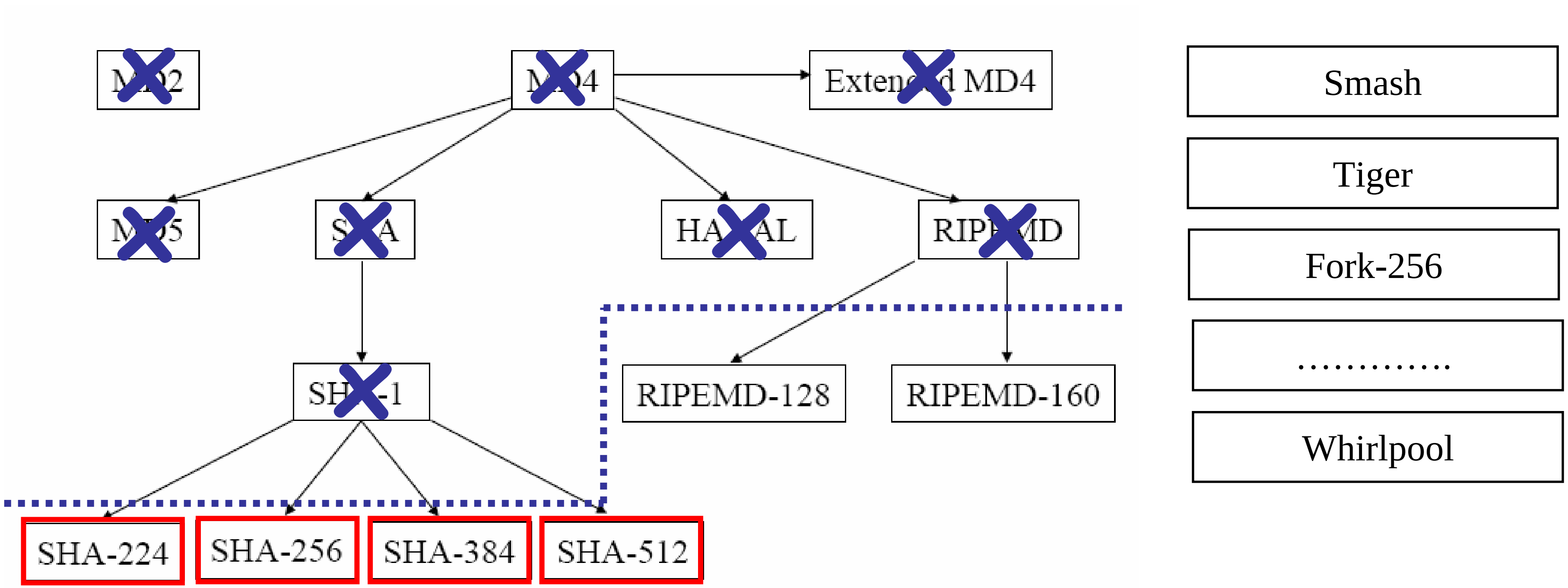
Summary

- First analysis of unmodified SHA-256/224 for a nontrivial number of steps

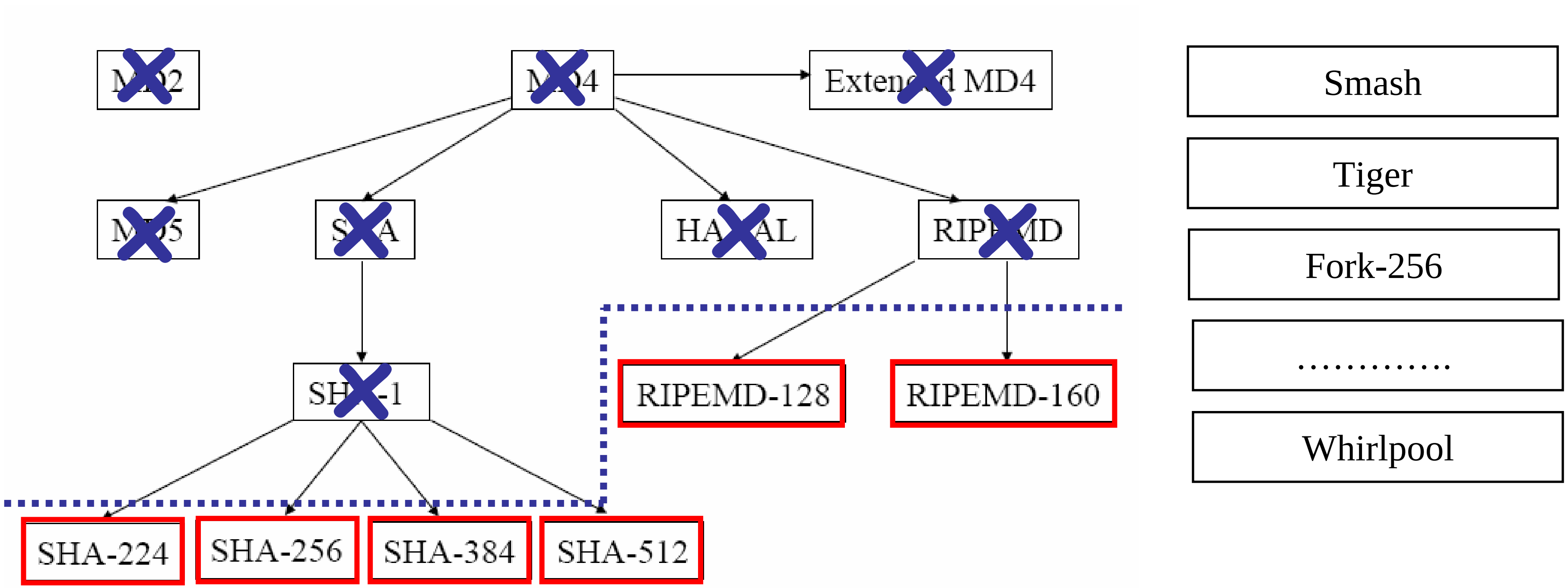
- Collision resistance of SHA-256/224 is not threatened
- All publicly known attacks on SHA-0/1 since 1997 are not directly applicable to any SHA-2 member

- New analysis method
  - New type of perturbation pattern
  - Probability of a local collision is much less relevant
  - Explicit control of carry extensions is possible and needed

Motivation


RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
………….
Whirlpool

Motivation


MD2
MD4
Extended MD4
MD5
SHA
HAVAL
RIPEMD
SHA-1
RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
………….
Whirlpool

On the Collision-Resistance of RIPEMD-160

Florian Mendel, Christian Rechberger,
Norbert Pramstaller, and Vincent Rijmen

presented at ISC 2006

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

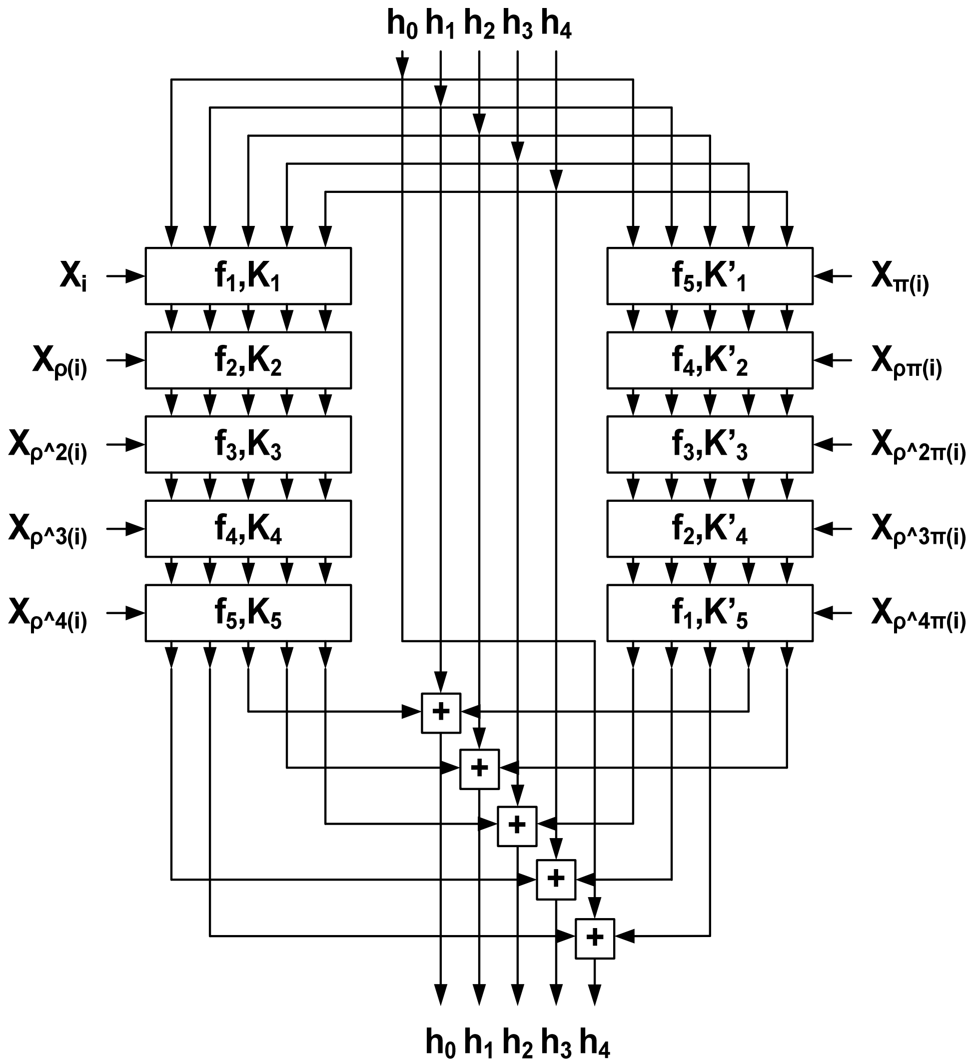
Faculty of Computer Science
Graz University of Technology

The RIPEMD-family

- RIPEMD
  - Results by Dobbertin (round reduced)
  - Collisions announced in 2004 by Wang et al.
- Introduction of two strengthened versions
  - RIPEMD-128
  - RIPEMD-160
- RIPEMD-160 is frequently recommended
- Attacks extendable to RIPEMD-160?

RIPEMD-160 / 128

- RIPEMD-160
  - Output is 160 bits
  - Process message in 16 words (512-bit)
  - Uses 10 rounds of 16 steps in **2 parallel lines of 5**
- RIPEMD-128
  - Output 128 bits
  - Uses 8 rounds of 16 steps in **2 parallel lines of 4**

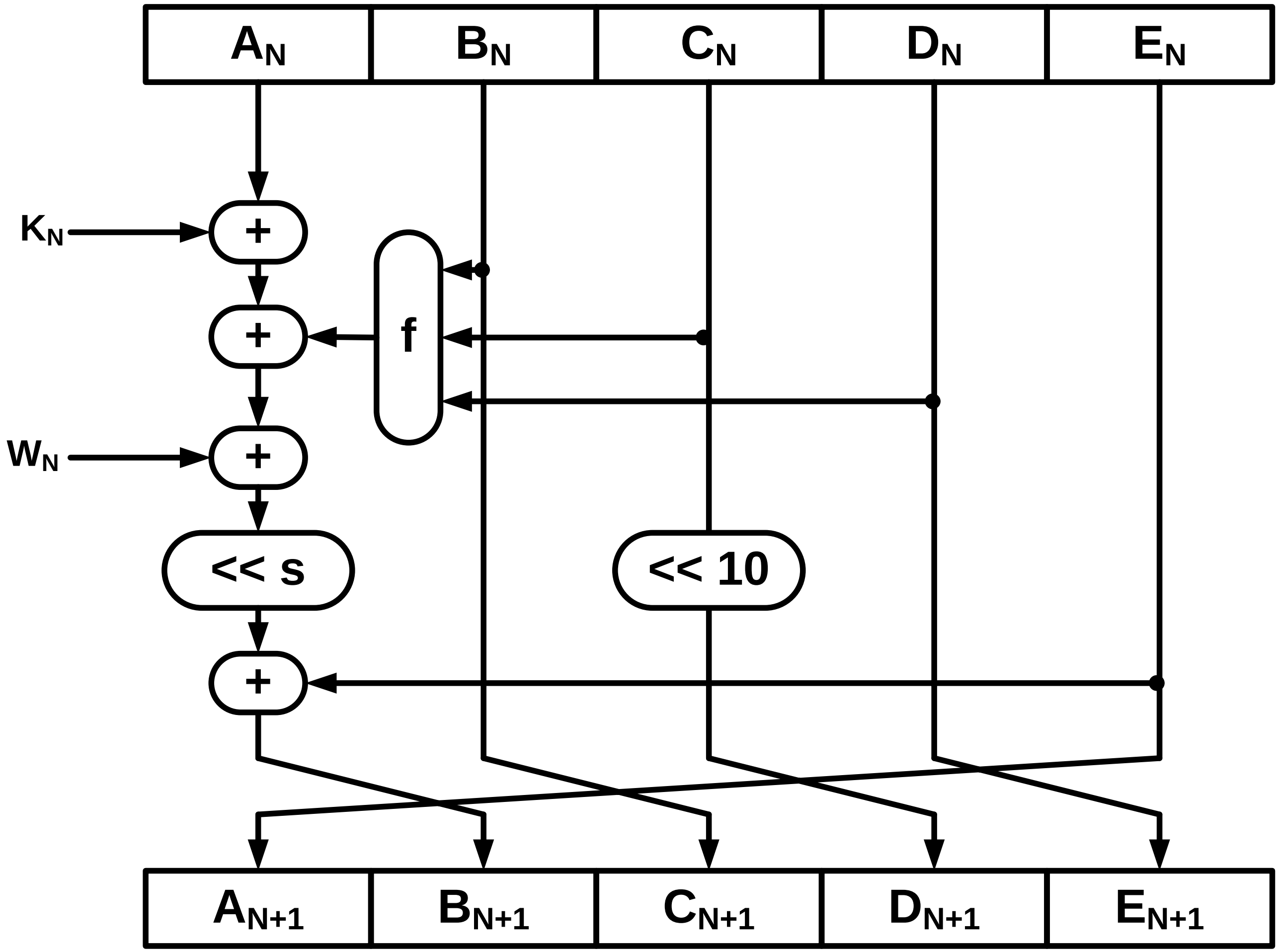


Step Function of RIPEMD-160

Modular Additions

Boolean Functions

Variable Rotation

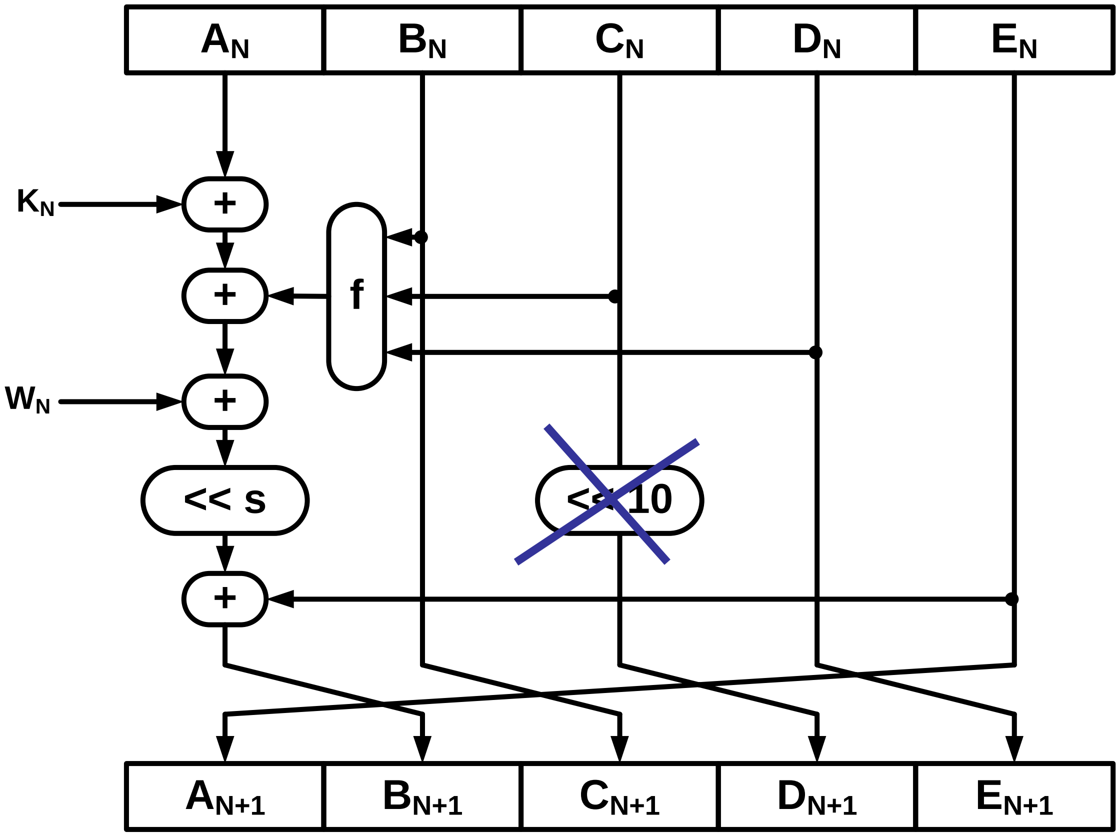


Results of the low-weight search using a general characteristic

The attack of Wang *etal.* on SHA-1 does not apply to RIPEMD-160 – no characteristic with low Hamming weight can be found

	Hamming weight	Stream	#Steps
RIPEMD - 160	480	Both	17 - 80
	352	Both	17 - 64
	224	Both	17 - 48
RIPEMD - 128	448	Both	17 - 64
	18	Both	17 - 48

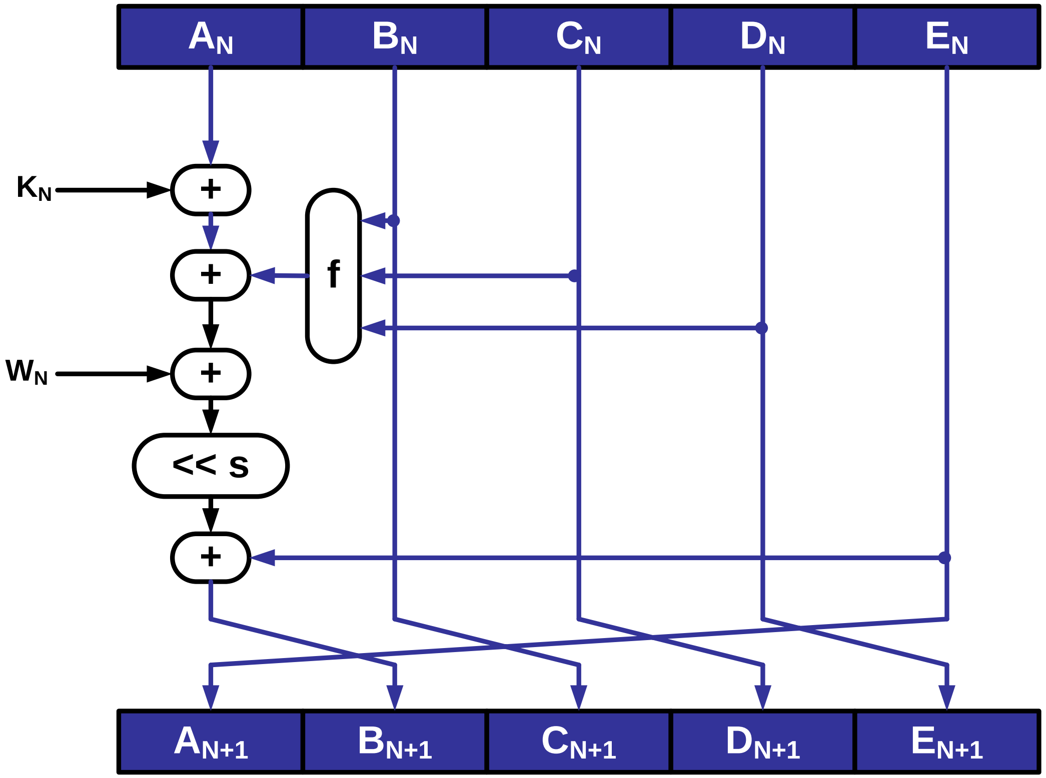
A simplified variant of RIPEMD-160



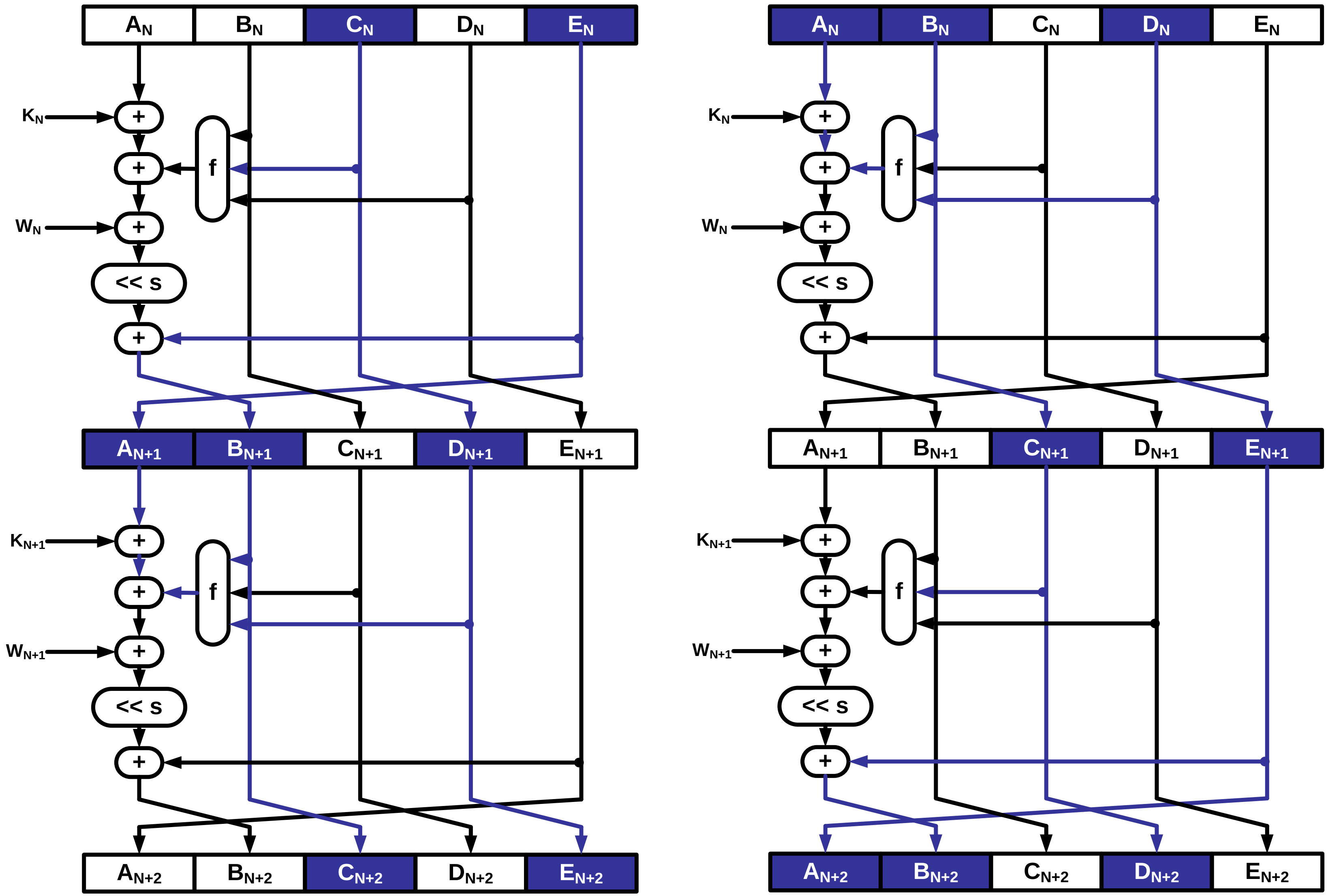
Note: Rotation of C is removed

Fixed Points in the simplified variant of RIPEMD-160

- Input differences = output differences
- Properties of **f** can be used to cancel differences in W_T



Fixed Points (for 2 steps)



Using fixed points for collisions

- Collision for RIPEMD-160 variant reduced to 3 rounds using fixed point FP_1:
  - 1 message block
  - 64 equations on A_N
- Collision for RIPEMD-160 variant reduced to 3 rounds using fixed point FP_{2a} or FP_{2b}
  - 5 message blocks
  - For each message block there are 48 equations on A_N
- Theoretical attack for RIPEMD-160 variant reduced to 3 rounds

Summary

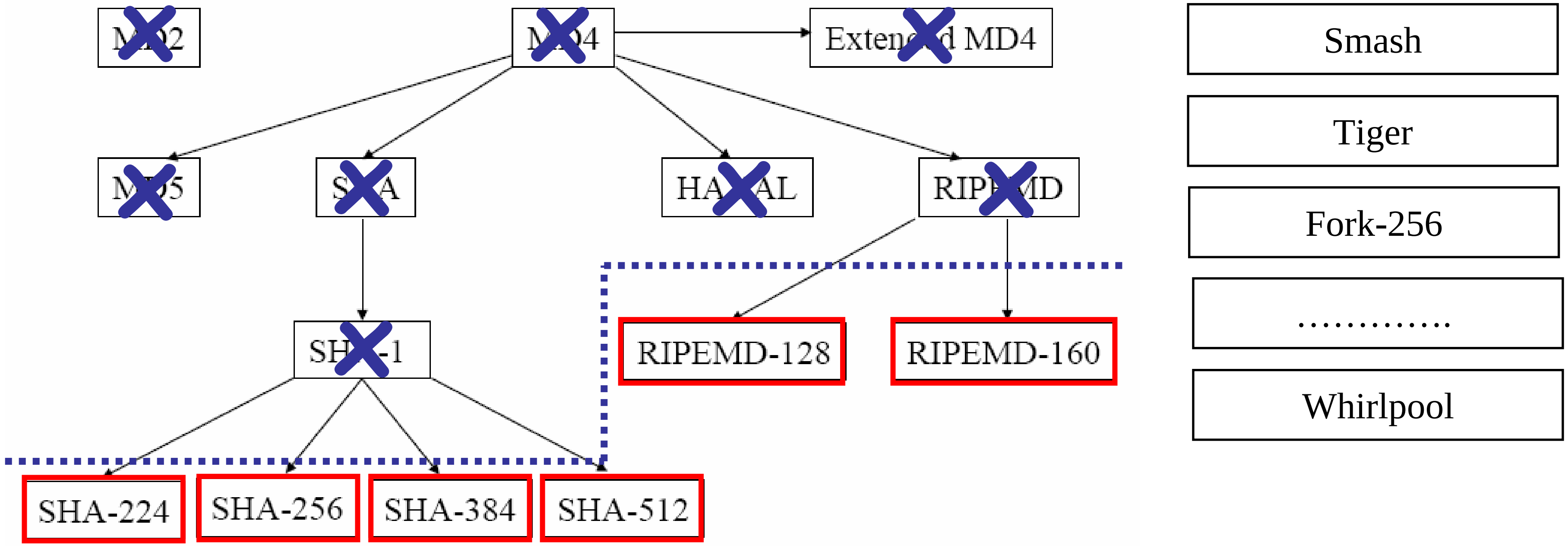
- Theoretical attack on 3 rounds of a simplified variant of RIPEMD-160
- So far no results for the original RIPEMD-160 hash function
  - Number of equations is too large
  - No differential pattern found with low Hamming weight
- RIPEMD-160 seems to be secure against these kind of collision-attacks

Summary

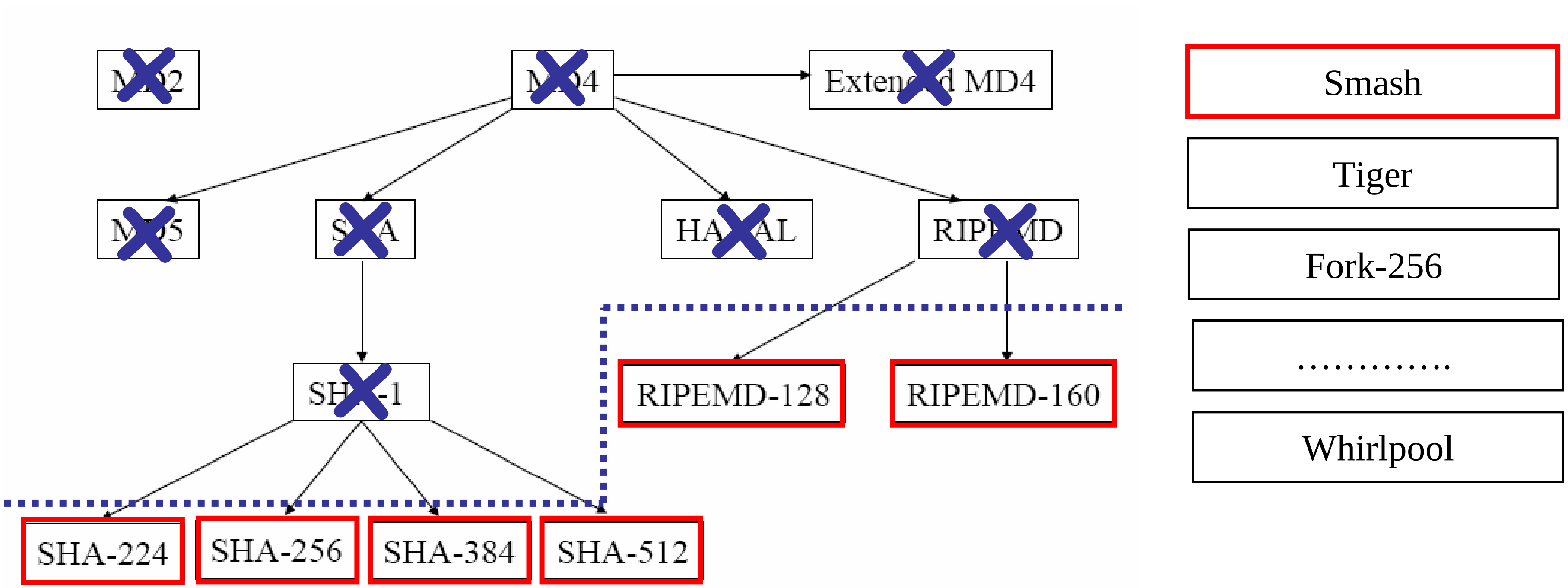
- Theoretical attack on 3 rounds of a simplified variant of RIPEMD-160
- So far no results for the original RIPEMD-160 hash function
  - Number of equations is too large
  - No differential pattern found with low Hamming weight
- RIPEMD-160 seems to be secure against these kinds of collision-attacks

Further analysis is required to get a good view on the security margins of RIPEMD-160 and RIPEMD-128

Motivation


Smash
Tiger
Fork-256
....
Whirlpool
RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512

Motivation


MD2
MD4
Extended MD4
MD5
SHA
HAVAL
RIPEMD
SHA-1
RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
………….
Whirlpool

Structural Analysis of SMASH

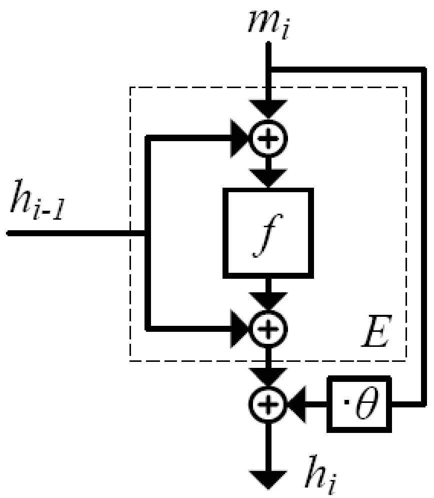
Mario Lamberger, Norbert Pramstaller,
Christian Rechberger, and Vincent Rijmen

presented at SAC 2005 and CT-RSA 2007

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology

SMASH Design Strategy

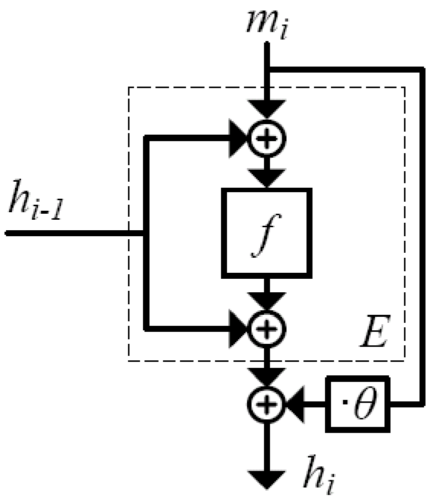


$$h_0 = f(\mathrm{iv}) + \mathrm{iv}$$
$$h_i = f(h_{i-1} + m_i) + h_{i-1} + \theta m_i \quad \text{for } i = 1, \ldots, t$$
$$h_{t+1} = f(h_t) + h_t \ .$$

- Compression function based on nonlinear bijective n-bit mapping f
- θ is an arbitrary field element in $GF(2^n)$ with $\theta \neq \{0, 1\}$

$+$ … addition in $GF(2^n)$

SMASH Design Strategy



$$h_0 = f(\text{iv}) + \text{iv}$$
$$h_i = f(h_{i-1} + m_i) + h_{i-1} + \theta m_i \quad \text{for } i = 1, \ldots, t$$
$$h_{t+1} = f(h_t) + h_t \ .$$

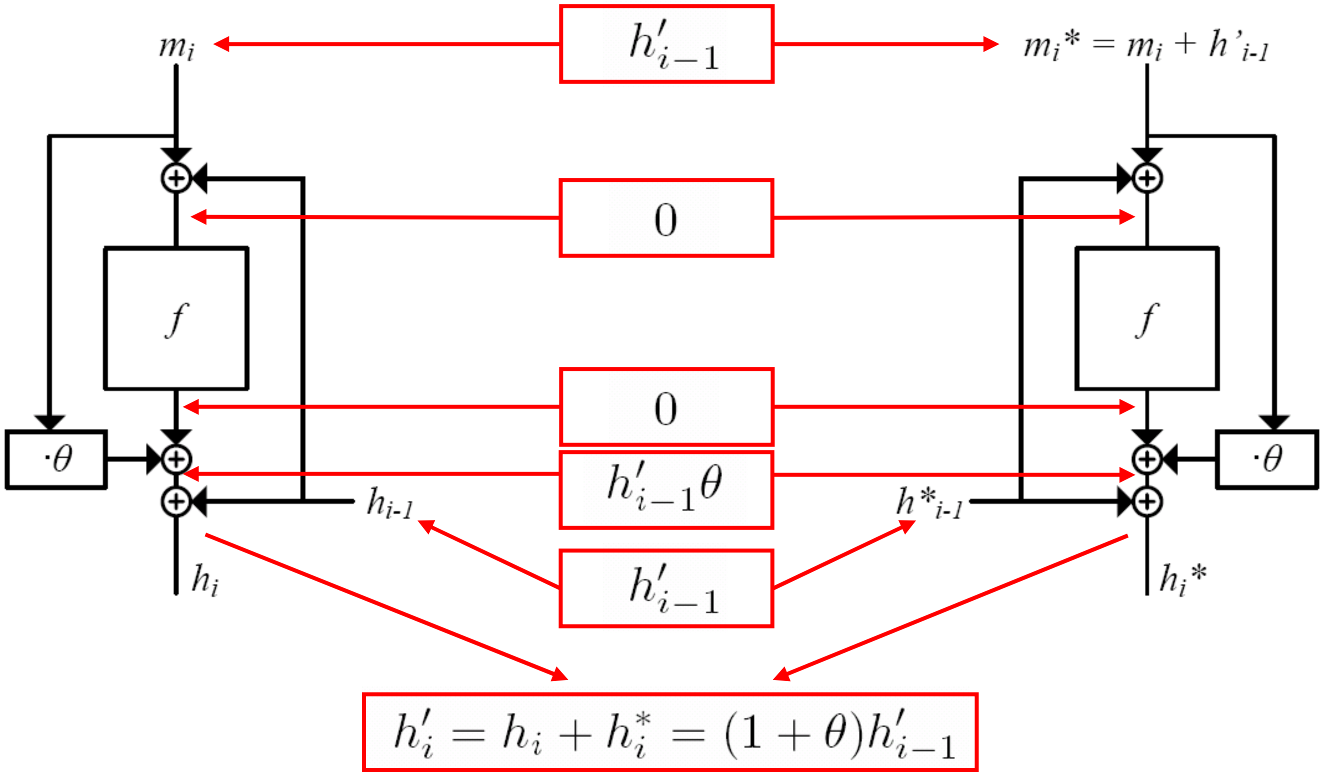
- Compression function based on nonlinear bijective n-bit mapping f
- θ is an arbitrary field element in $GF(2^n)$ with $\theta \neq \{0,1\}$

- Specific instance: SMASH-256 (n=256)
  - $GF(2^{256})$ defined by $q(\alpha) = \alpha^{256} + \alpha^{16} + \alpha^3 + \alpha + 1$
  - Element θ is defined as root of $q(\alpha)$

$+$ … addition in $GF(2^n)$

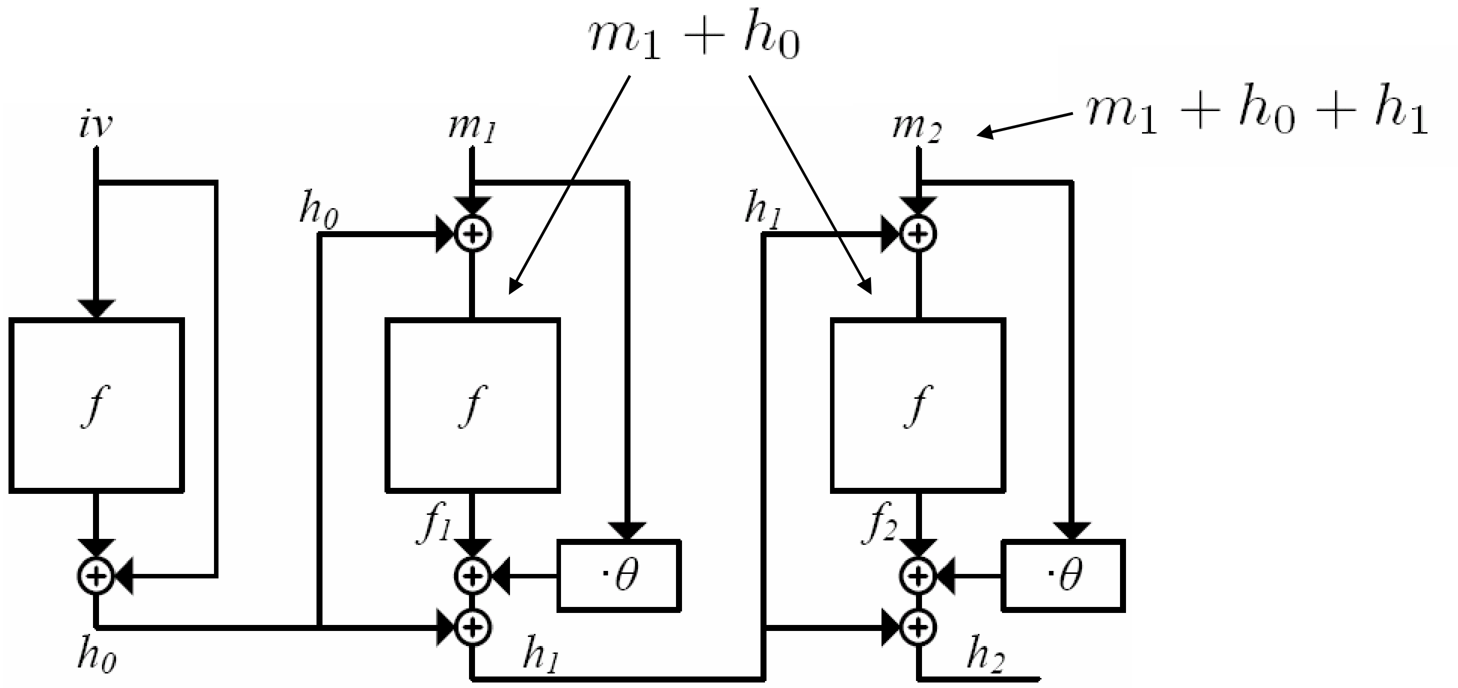
Forward Prediction Property (FPP)

- Given intermediate hash values h_{i-1}, h^*_{i-1} with difference $h'_{i-1} = h_{i-1} + h^*_{i-1}$
- Choose m_i and compute $m^*_i = m_i + h'_{i-1}$



Pattern Construction Property (PCP)

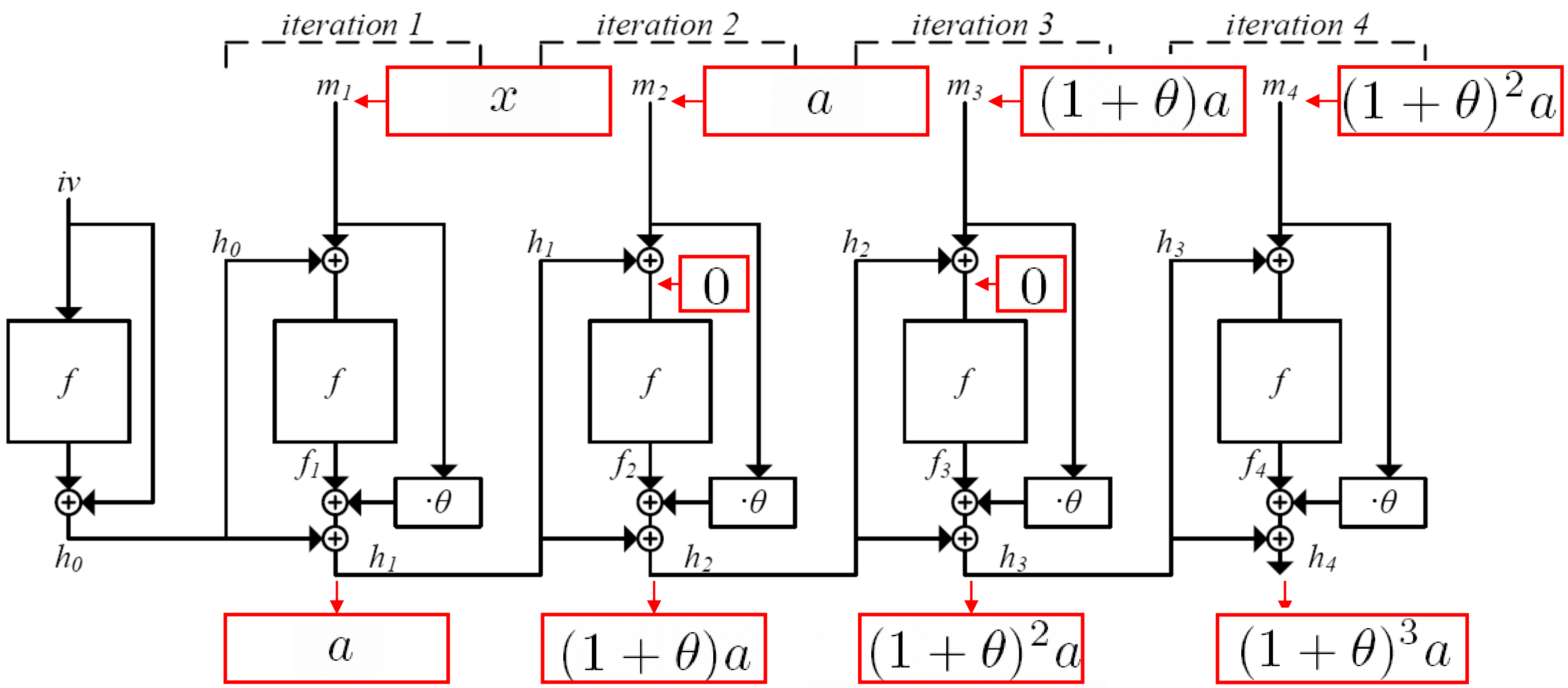
- Input of f must be the same for both iterations



$$m_2 = m_1 + f_1 + \theta m_1 \Rightarrow f_2 = f_1$$

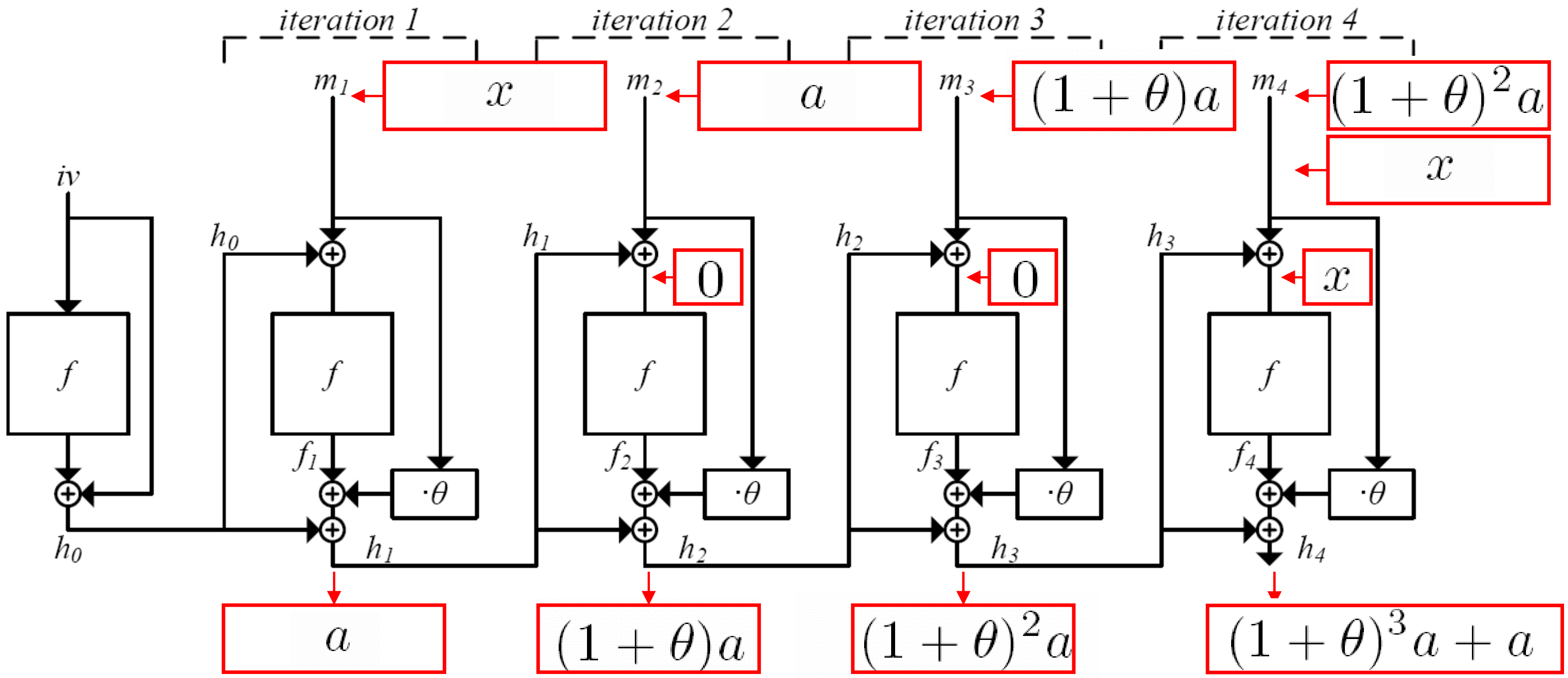
Exploiting FPP/PCP for Collisions – The Principle

- Assume we can choose a θ such that $(1+\theta)^3 = 1$



Exploiting FPP/PCP for Collisions – The Principle

- Assume we can choose a θ such that $(1+\theta)^3 = 1$

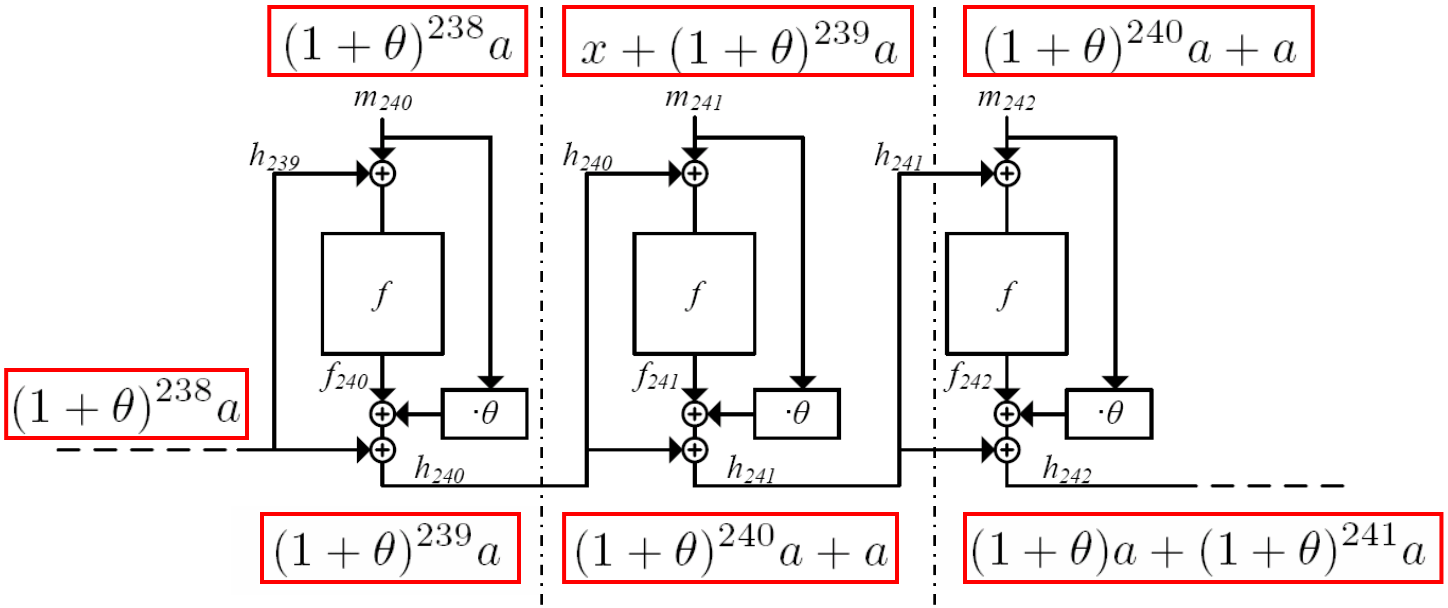


Exploiting FPP/PCP for Collisions in SMASH-256

- For a collision we need

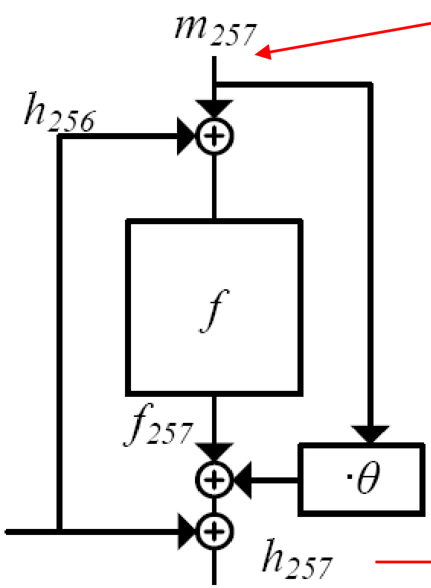
$$a \cdot q(\theta) = (1+\theta)^{256}a + (1+\theta)^{16}a + (1+\theta)^{3}a + (1+\theta)^{2}a + a$$

- Constructing the polynomial



Exploiting FPP/PCP for Collisions in SMASH-256

- Introduce non-zero difference x in $i = 1, 241, 254, 255, 257$
  - 257 message blocks needed
  - 4 message blocks determined by attack
  - 253 message blocks can be chosen arbitrarily



$$x + (1+\theta)^{255}a + (1+\theta)^{15}a + (1+\theta)^{2}a + (1+\theta)a$$

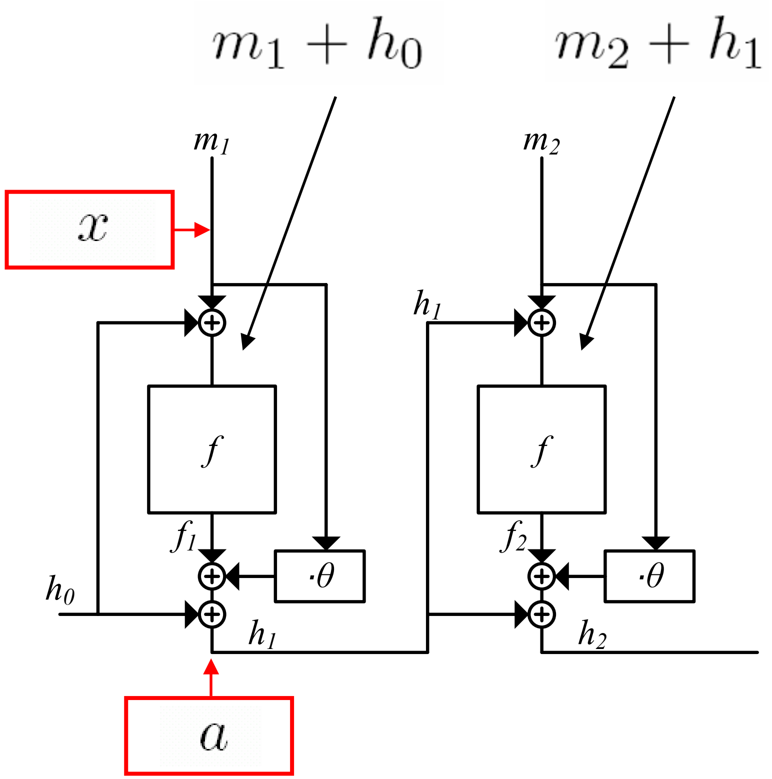
$$\theta^{256} + \theta^{16} + \theta^{3} + \theta + 1$$
$$= (1+\theta)^{256} + (1+\theta)^{16} + (1+\theta)^{3} + (1+\theta)^{2} + 1$$

$$(1+\theta)^{256}a + (1+\theta)^{16}a + (1+\theta)^{3}a + (1+\theta)^{2}a + a$$

=> collision after iteration 257

Second Preimages for SMASH

- Message *m*: allow PCP in each iteration

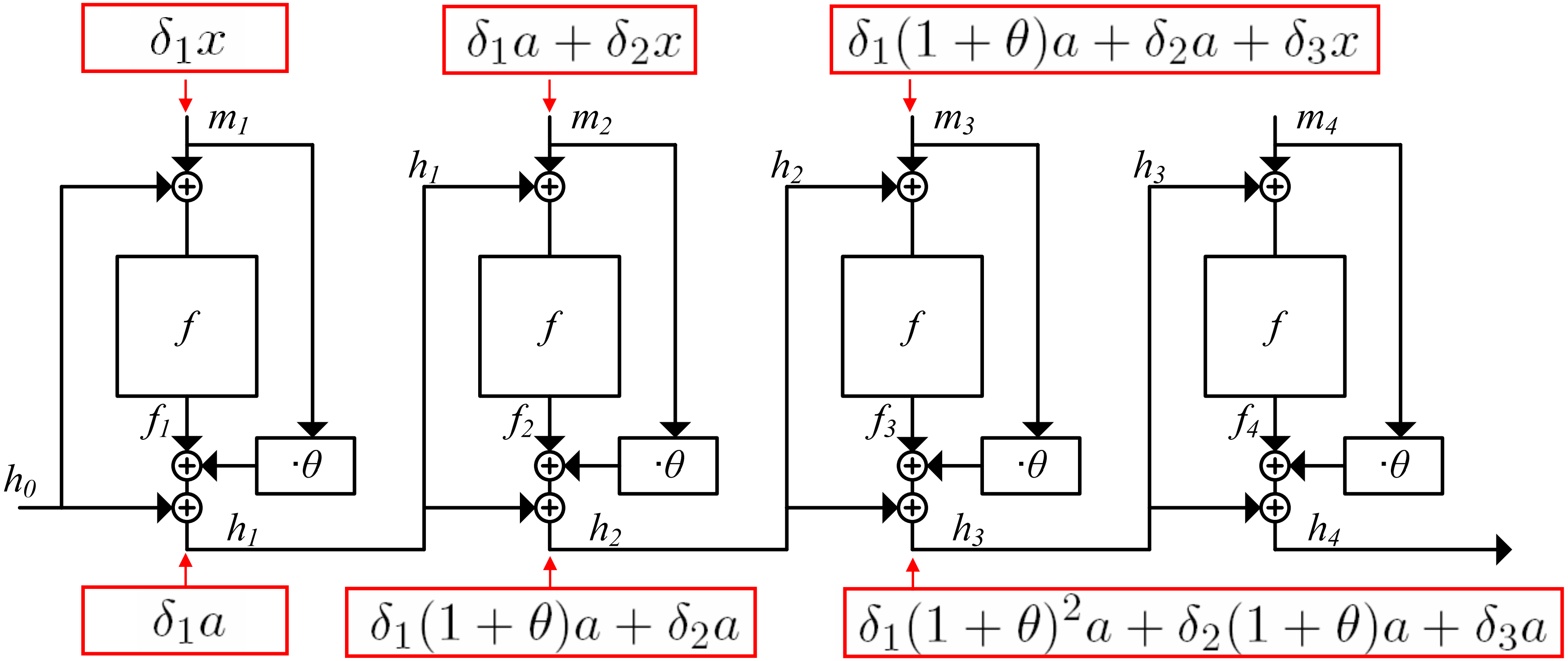


$$\Longrightarrow m_2 = m_1 + h_0 + h_1$$

$$\mathrm{m}_i = \mathrm{m}_1 + \mathrm{h}_0 + \mathrm{h}_{i-1}, 1 < i \leq n$$

Second Preimages for SMASH

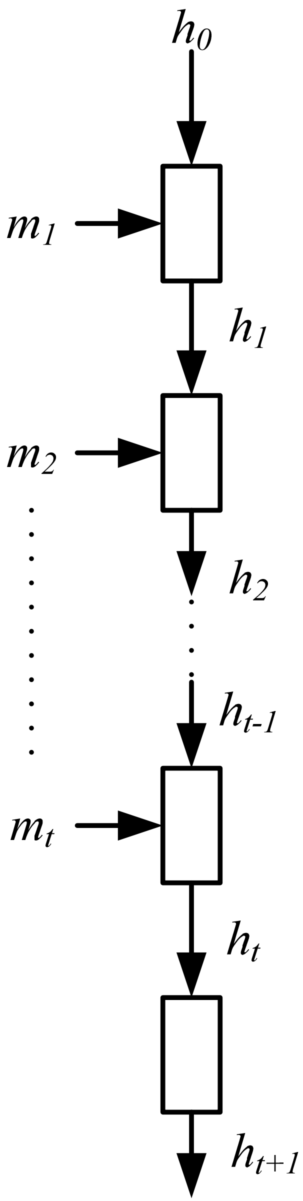
- Message $m^*(\delta)$:



$$\delta_i \in \{0, 1\}$$

$$\mathrm{h}_n + \mathrm{h}_n^* = a \sum_{j=1}^{n} \delta_j (1+\theta)^{n-j}$$

Second Preimages for SMASH



- Given m, difference x, and δ

$$\mathrm{h}_n + \mathrm{h}_n^* = a \sum_{j=1}^{n} \delta_j (1+\theta)^{n-j}$$

- Given m, difference x, and $\mathrm{h}_n + \mathrm{h}_n^*$
  - Set of n linear equations in unknowns δ_i

$$A_{n \times n} \times \delta \neq 0$$

  - SMASH-256/512: $A_{n \times n}$ full rank => solution

Second Preimages for SMASH

- For t-block messages
  - $t \geq n$: same approach applies
  - $t < n$: same approach but probabilistic

- Summary of second preimage attacks

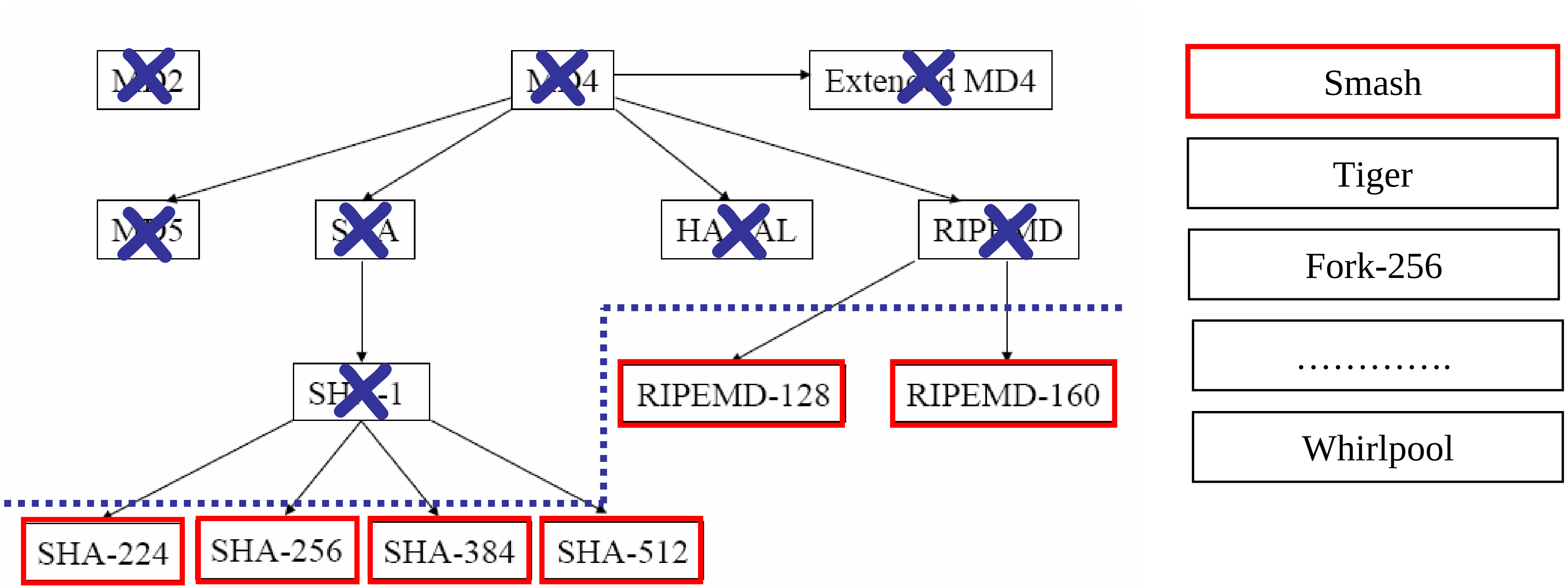
type	message length t	number of blocks the attacker can choose	probability
meet-in-the-middle [5]	≥ 2	$t-2$	$2^{-n/2}$
this paper	$\geq n+1$	$t-n$	1
this paper	$< n+1$	1	2^{t-1-n}

Summary and Further Work

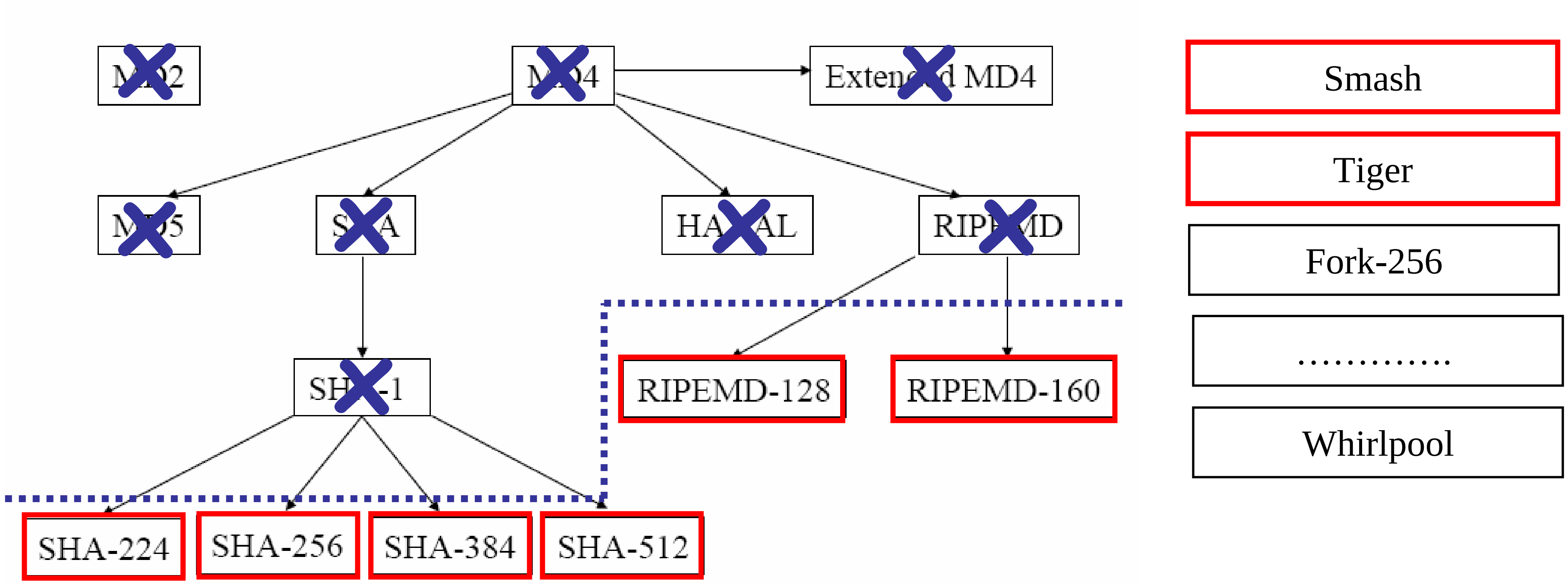
- Structural analysis of SMASH
  - Second preimages
    - direct construction
  - Special case: collisions

- Further work
  - Other hash functions
  - Preimages for SMASH
  - Generalizing strategy
    - FPP and PCP
    - Looking at different compression functions

Motivation


MD2
MD4
Extended MD4
MD5
SHA
HAVAL
RIPEMD
SHA-1
RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
………….
Whirlpool

Motivation


MD2
MD4
Extended MD4
MD5
SHA
HAVAL
RIPEMD
SHA-1
RIPEMD-128
RIPEMD-160
SHA-224
SHA-256
SHA-384
SHA-512
Smash
Tiger
Fork-256
………….
Whirlpool

Update on Tiger

Florian Mendel, Vincent Rijmen,
Graz University of Technology
Institute for Applied Information Processing and Communications
Hirotaka Yoshida, and Dai Watanabe
Systems Development Laboratory, Hitachi, Ltd.
Bart Preneel
Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC

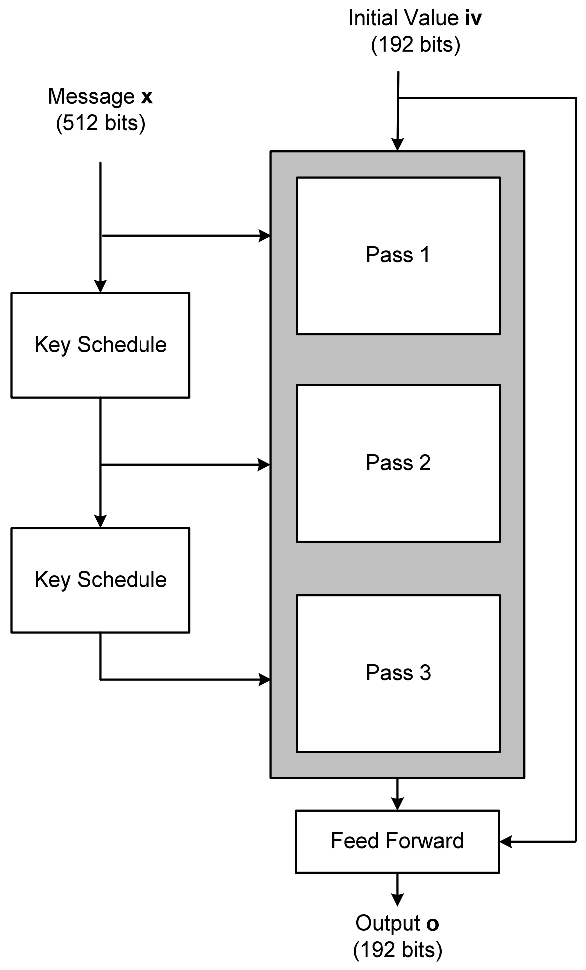
presented at Indocrypt 2006

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology

The Tiger Hash Function

- Iterated Hash Function processes 512-bit blocks and produces a 192-bit hash value



- Message expansion
  - 8 64-bit words to 24 64-bit words
- State Update Transformation
  - 3 passes each consists of 8 rounds

Message Expansion

- The message expansion of Tiger consists of 2 applications of the Key Schedule:

$$(X_8, \ldots, X_{15}) = \mathrm{KeySchedule}(X_0, \ldots, X_7)$$
$$(X_{16}, \ldots, X_{23}) = \mathrm{KeySchedule}(X_8, \ldots, X_{15})$$

Key Schedule

The Key Schedule of Tiger consists of 2 steps

first step

$$
\begin{aligned}
Y_0 &= Y_0 - (Y_7 \oplus \mathtt{A5A5A5A5A5A5A5A5})\\
Y_1 &= Y_1 \oplus Y_0\\
Y_2 &= Y_2 + Y_1\\
Y_3 &= Y_3 - (Y_2 \oplus ((\neg Y_1) \ll 19))\\
Y_4 &= Y_4 \oplus Y_3\\
Y_5 &= Y_5 + Y_4\\
Y_6 &= Y_6 - (Y_5 \oplus ((\neg Y_4) \gg 23))\\
Y_7 &= Y_7 \oplus Y_6
\end{aligned}
$$

second step

$$
\begin{aligned}
Y_0 &= Y_0 + Y_7\\
Y_1 &= Y_1 - (Y_0 \oplus ((\neg Y_7) \ll 19))\\
Y_2 &= Y_2 \oplus Y_1\\
Y_3 &= Y_3 + Y_2\\
Y_4 &= Y_4 - (Y_3 \oplus ((\neg Y_2) \gg 23))\\
Y_5 &= Y_5 \oplus Y_4\\
Y_6 &= Y_6 + Y_5\\
Y_7 &= Y_7 - (Y_6 \oplus \mathtt{0123456789ABCDEF})
\end{aligned}
$$

State Update Transformation

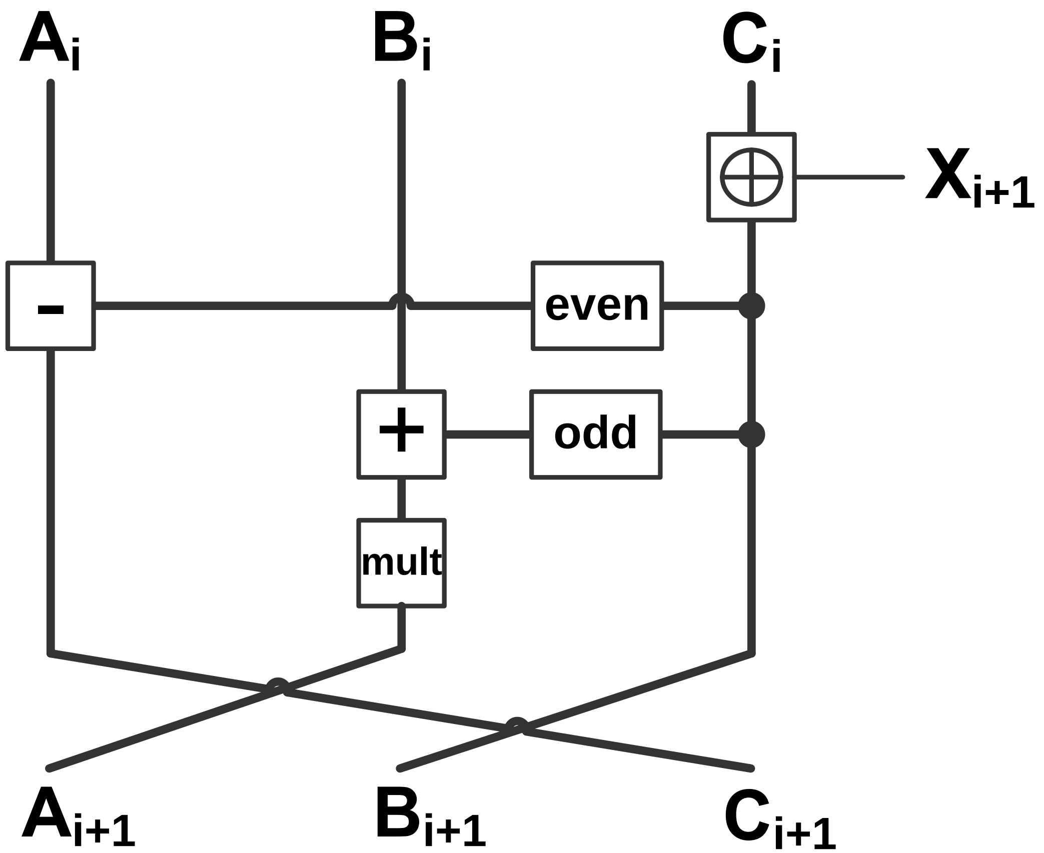
- 3 Passes (8 rounds each)

XOR Operation

Modular Additions/Subtractions

Non-linear Functions

Multiplication by a constant



State Update Transformation

- The non-linear functions *even* and *odd* used in each round are defined as follows:

$$\mathbf{even}(C) = T_1[c_0] \oplus T_2[c_2] \oplus T_3[c_4] \oplus T_4[c_6]$$
$$\mathbf{odd}(C) = T_4[c_1] \oplus T_3[c_3] \oplus T_2[c_5] \oplus T_1[c_7]$$

- 4 S-boxes are used $T_1, \ldots, T_4 : \{0,1\}^8 \rightarrow \{0,1\}^{64}$

- At the end of each round *B* is multiplied by a constant $\mathtt{mult} \in \{5, 7, 9\}$. This constant is different for each pass of Tiger.

Basic Attack Strategy

- Choose a characteristic for the Key Schedule of Tiger that holds with high probability (ideally with probability 1).
- Use a kind of message modification technique to construct certain differences in the chaining variables, which can then be canceled by the differences in the message words in the following rounds.

Attack on 16 Rounds of Tiger

- Key Schedule difference for collision in Tiger-16

$$(I, I, I, I, 0, 0, 0, 0) \rightarrow (I, I, 0, 0, 0, 0, 0, 0)$$

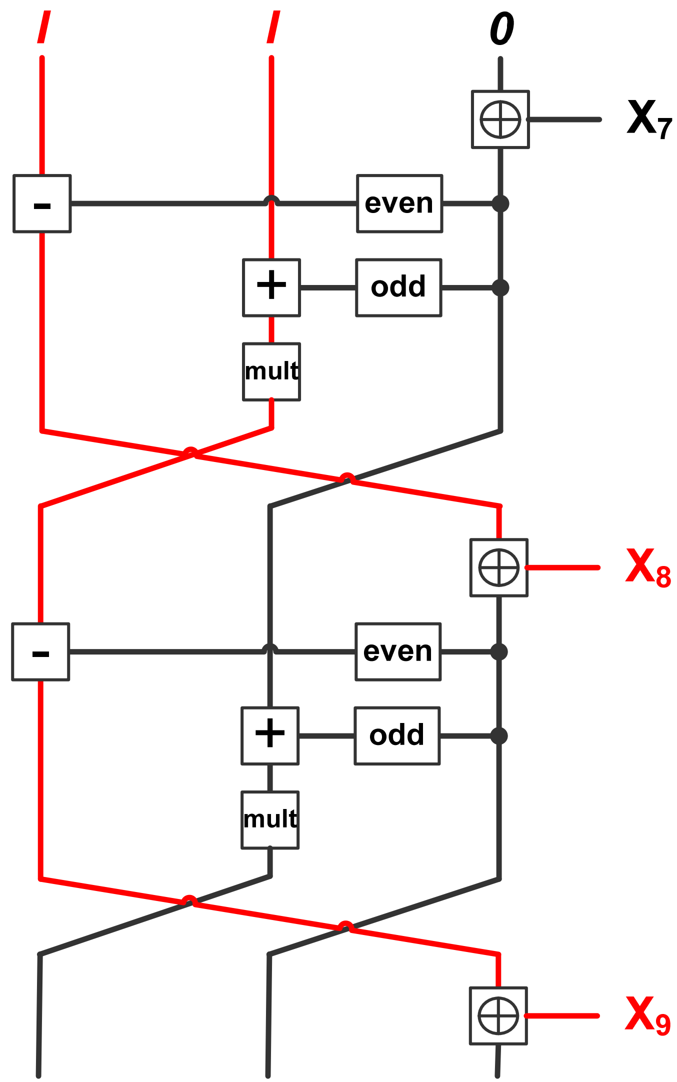
- To have a collision after 16 rounds the following difference is needed in the chaining variables in round 7

$$\Delta^+(A_6) = I, \quad \Delta^+(B_6) = I, \quad \Delta^+(C_6) = 0$$

- In the attack Kelsey and Lucks use a kind of Message modification technique developed for Tiger to construct the needed differences.

Collision for 16 rounds of Tiger

- Needed target difference *(I,I,0)*
- Canceled by words *8 and 9*
- Collision after 10 rounds of Tiger
- No difference in remaining words
  => Collision for 16 rounds



Message Modification by Meet-in-the-Middle

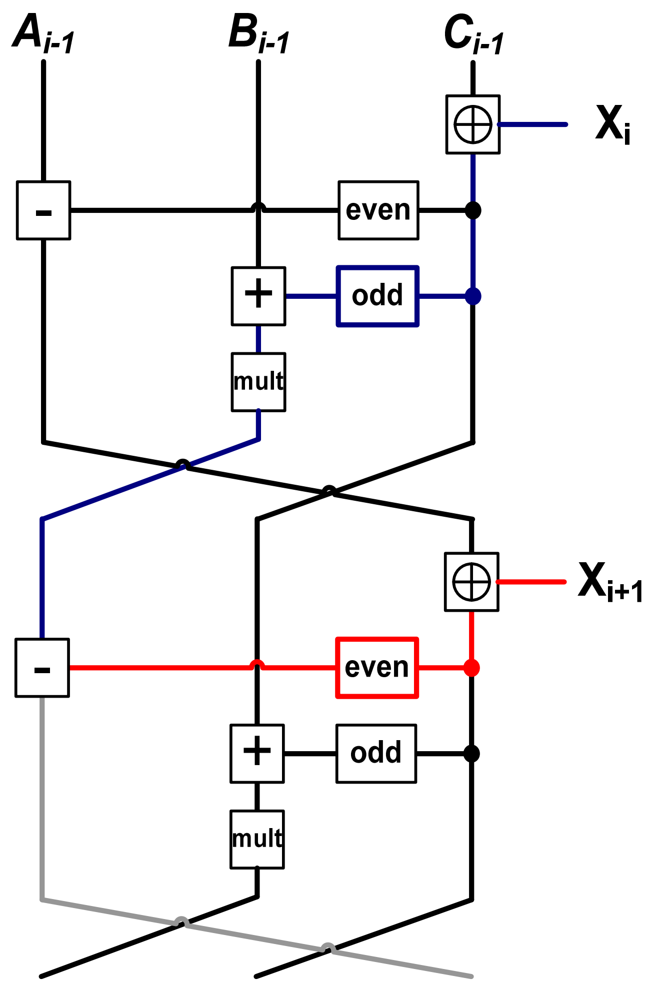
$$\texttt{mult} \times (\Delta^+(B_{i-1}) + \Delta^+(\mathbf{odd}(B_i)))$$

$$-$$

$$\Delta^+(\mathbf{even}(B_{i+1}))$$

$$=$$

$$\delta^*$$



Message Modification by Meet-in-the-Middle

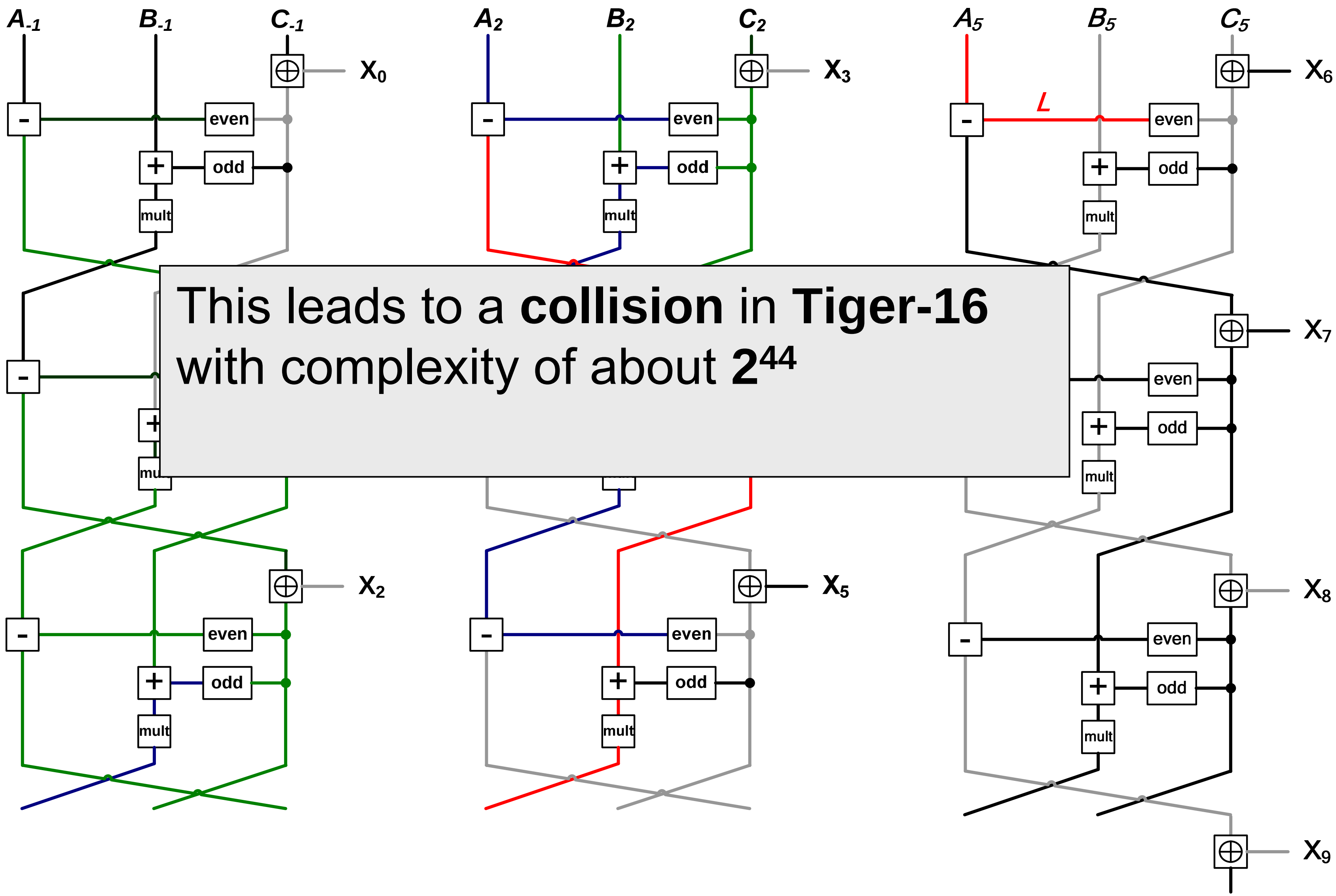
- Use a MITM approach to solve the equation:

$$\underbrace{\boxed{\mathtt{mult} \times (\Delta^{+}(B_{i-1}) + \Delta^{+}(\mathbf{odd}(B_i)))}}_{\mathbf{E}} - \underbrace{\boxed{\Delta^{+}(\mathbf{even}(B_{i+1}))}}_{\mathbf{F}} = \boxed{\delta^{*}}$$

- Store the 2^{32} candidates for **E** in a table
- For all 2^{32} candidates for **F** test if some **E** exists with **E** – **F** = δ^{*}

This technique takes about $\mathbf{2^{29}}$ evaluations of the compression function of Tiger

Outline of the Attack



This leads to a **collision** in **Tiger-16** with complexity of about $\mathbf{2^{44}}$

Going beyond 16 Rounds

- Attack of Kelsey and Lucks (FSE 2006)
  - Collision 16 rounds of Tiger with complexity of about 2^{44}
  - Pseudo-near-collision for 20 rounds of Tiger (4 - 24) with complexity of about 2^{48}

- Extended Attack of Mendel *etal.* (Indocrypt 2006)
  - Collision for 19 rounds of Tiger with complexity of about 2^{62}
  - Pseudo-near-collision for 22 rounds of Tiger (1 - 22) with complexity of about 2^{44}
  - …

A Collision for Tiger-19

- In the attack we use the Key Schedule difference:

$$(0,0,0,I,I,I,I,0) \rightarrow (0,0,0,I,I,0,0,0) \rightarrow (0,0,0,\not{I},\not{I},\not{I},\not{I},\not{I})$$

A Collision for Tiger-19

- In the attack we use the Key Schedule difference:

$$(0,0,0,\boxed{I,I,I,I,0) \rightarrow (0,0,0,I,I,0,0,0) \rightarrow (0,0,0,}\cancel{I},\cancel{I},\cancel{I},\cancel{I},\cancel{I})$$

- Note that the Key Schedule difference from round 3 to 18 is the 16-round difference used in the attack on Tiger-16

$$\boxed{(I,I,I,I,0,0,0,0) \rightarrow (I,I,0,0,0,0,0,0)}$$

Outline of the Attack

- Choose arbitrary values for A_2, B_2, C_2 in round 3

Outline of the Attack

- Choose arbitrary values for A_2, B_2, C_2 in round 3
- Employ the attack on 16 rounds, to find message words $X_3, \ldots, X_7$ and $X_8[\mathbf{even}], X_9[\mathbf{even}]$ such that the outputs collide after 19 rounds

Outline of the Attack

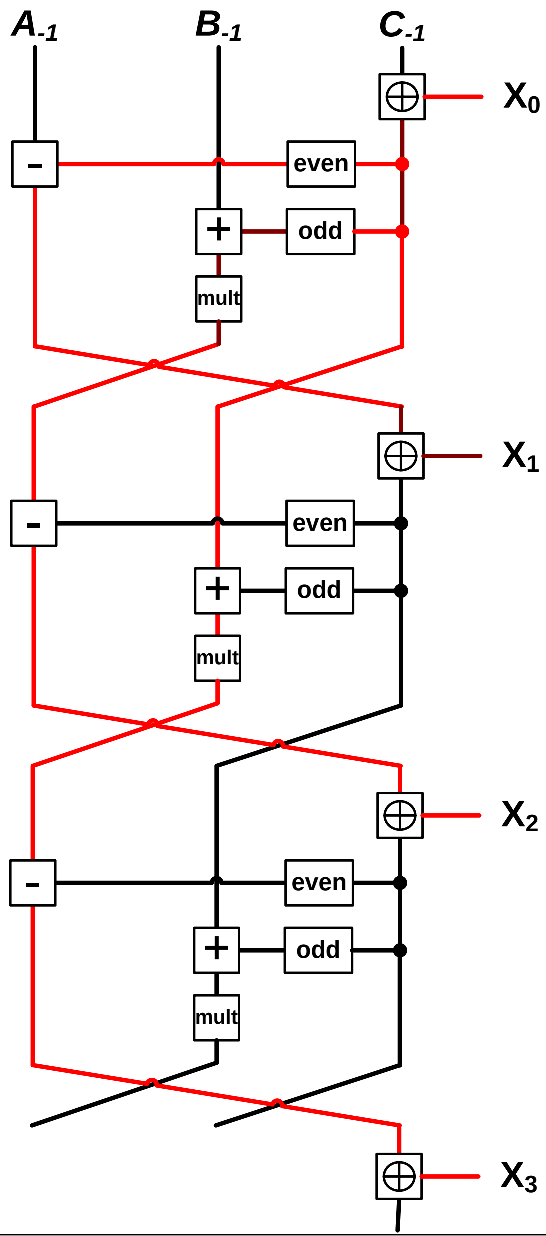
- Choose arbitrary values for A_2, B_2, C_2 in round 3
- Employ the attack on 16 rounds, to find message words $X_3, \ldots, X_7$ and $X_8[\mathbf{even}], X_9[\mathbf{even}]$ such that the outputs collide after 19 rounds
- Compute the message words X_0, X_1, X_2 such that $X_8[\mathbf{even}], X_9[\mathbf{even}]$ are correct after computing the Key Schedule

Outline of the Attack

- Choose arbitrary values for A_2, B_2, C_2 in round 3
- Employ the attack on 16 rounds, to find message words $X_3, \ldots, X_7$ and $X_8[\mathbf{even}], X_9[\mathbf{even}]$ such that the outputs collide after 19 rounds
- Compute the message words X_0, X_1, X_2 such that $X_8[\mathbf{even}], X_9[\mathbf{even}]$ are correct after computing the Key Schedule
- Run the rounds 2,1 and 0 backward to get the initial values A_{-1}, B_{-1} and C_{-1}

Collision in Tiger-19

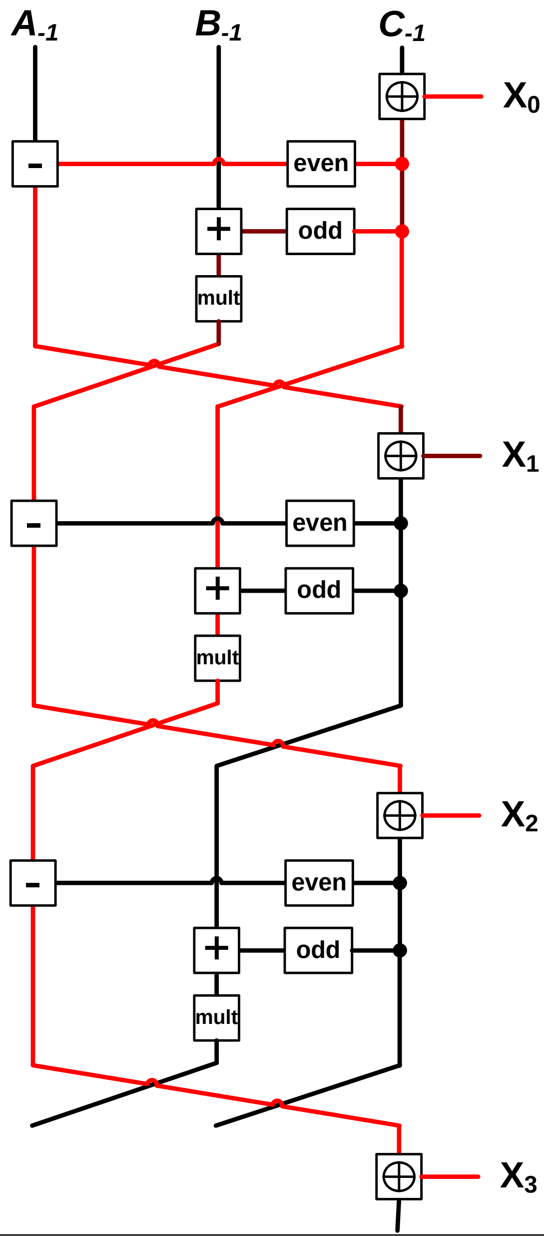
- Use the degree of freedom we have in the choice of the message words X_0, X_1, X_2, X_3 to guarantee that the message words $X_8[\mathbf{even}], X_9[\mathbf{even}]$ are correct after computing the Key Schedule of Tiger



Collision in Tiger-19

- Use the degree of freedom we have in the choice of the message words X_0, X_1, X_2, X_3 to guarantee that the message words $X_8[\text{even}], X_9[\text{even}]$ are correct after computing the Key Schedule of Tiger

This leads to a **collision** in **Tiger-19** with complexity of about $\mathbf{2^{62}}$



Summary

rounds	type	complexity	$\Delta \rightarrow \Delta$
Tiger-16	collision	2^{44}	
Tiger-19	collision	2^{62}	
Tiger-19	pseudo-collision	2^{44}	
Tiger-21	pseudo-near-collision	2^{44}	$(I, 0, 0) \rightarrow (I, 0, 0)$
Tiger-22	pseudo-near-collision	2^{44}	$(0, I, 0) \rightarrow (0, I, 0)$

Summary and Future Work

- Extending the method to find a collision in full Tiger hash function seems to be difficult
- By using a weaker attack scenario (pseudo-collisions, pseudo-near-collisions, etc.) it seems to be more likely that the attacks can be extended to full Tiger
- Future Work
  - Consider also characteristics for the Key Schedule with lower probability (not only probability 1)
  - Use of non-linear characteristics in the KS of Tiger

Conclusion

- Recent results in cryptanalysis show weaknesses in many commonly used hash functions
  - MD4, MD5, RIPEMD
  - SHA-1
  - …
- Hash functions that appear to be immune against existing attacks
  - SHA-2 family, RIPEMD-160
    - based on MD4 (!)
  - Whirlpool

Thank you for your Attention

Institute for Applied Information Processing and Communications (IAIK) - Krypto Group

Faculty of Computer Science
Graz University of Technology