Collision-Correlation Attack against a First-Order Masking Scheme for MAC based on SHA-3

Luk Bettale   Emmanuelle Dottax   Laurie Genelle   Gilles Piret

Oberthur Technologies Crypto Group

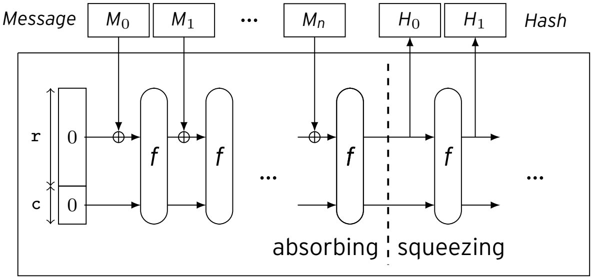
COSADE 2014
April 14th and 15th, Paris

The Keccak Hash Function

- Bertoni, Daemen, Peeters, Van Assche
- SHA-3 winner
- Sponge construction

The Keccak Hash Function

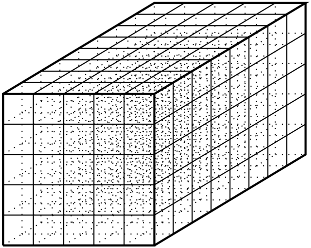
- Bertoni, Daemen, Peeters, Van Assche
- SHA-3 winner
- Sponge construction



Security relies on the function f.

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

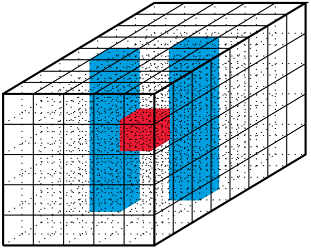
$$A \leftarrow \iota_i \circ \chi \circ \pi \circ \rho \circ \theta(A)$$

end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows
- ι xor with cst

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

$A \leftarrow \iota_i \circ \chi \circ \pi \circ \rho \circ \theta(A)$

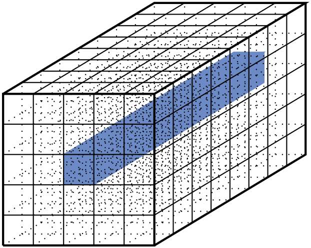
end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows
- ι xor with cst

oberthur TECHNOLOGIES THE M COMPANY

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

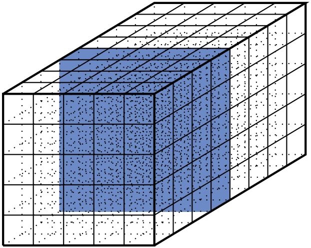
$A \leftarrow \iota_i \circ \chi \circ \pi \circ \rho \circ \theta(A)$

end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows
- ι xor with cst

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

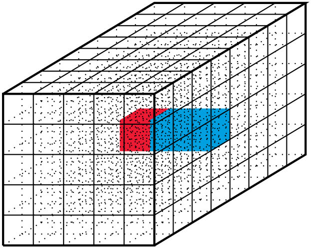
$$A \leftarrow \iota_i \circ \chi \circ \pi \circ \rho \circ \theta(A)$$

end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows
- ι xor with cst

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

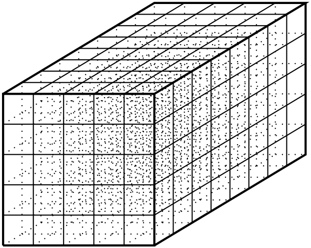
$A \leftarrow \iota_i \circ \chi \circ \pi \circ \rho \circ \theta(A)$

end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows, **non linear**
- ι xor with cst

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

$A \leftarrow \pi \circ \rho \circ \theta(A)$

$A \leftarrow \chi(A)$

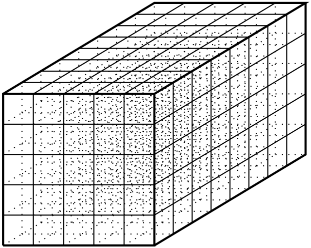
$A \leftarrow \iota_i(A)$

end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows, **non linear**
- ι xor with cst

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**

$A \leftarrow \lambda(A)$

$A \leftarrow \chi(A)$

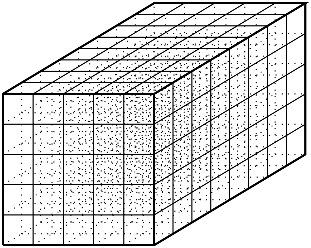
$A \leftarrow A + K_i$

end for

- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows, **non linear**
- ι xor with cst

Keccak f function

State A: $5 \times 5 \times w$ bits.



for $i = 1$ **to** 24 **do**
$A \leftarrow \lambda(A)$
$A \leftarrow \chi(A)$
$A \leftarrow A + K_i$
end for

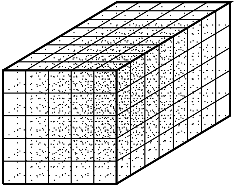
- θ xors columns
- ρ mix lanes
- π mix slices
- χ mix rows, **non linear**
- ι xor with cst

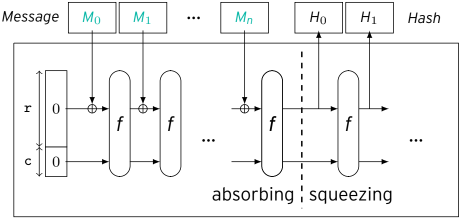
Practical Instantiation

$w = 64$ bits lane $\Rightarrow 1600$ bits state.

MAC based on Keccak

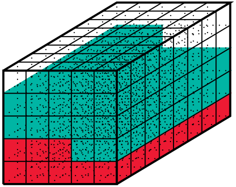
- Sponge construction $\Rightarrow$ no need for HMAC
- $\mathrm{MAC} = \lfloor H(K \| M) \rfloor_{\mathtt{p}}$

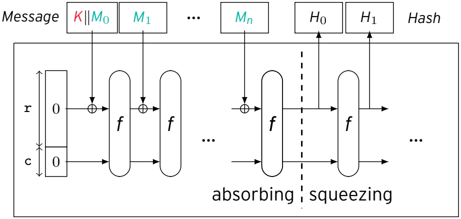




MAC based on Keccak

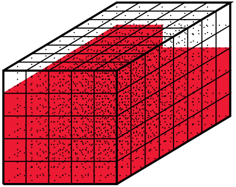
- Sponge construction $\Rightarrow$ no need for HMAC
- $\mathrm{MAC} = \lfloor H(K \| M) \rfloor_{\mathtt{p}}$

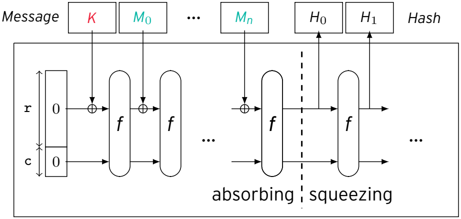




MAC based on Keccak

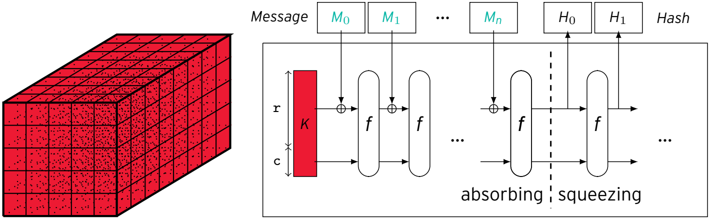
- Sponge construction $\Rightarrow$ no need for HMAC
- $\mathrm{MAC} = \lfloor H(K \| M) \rfloor_{\mathtt{p}}$





MAC based on Keccak

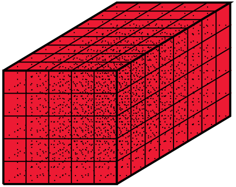
- Sponge construction $\Rightarrow$ no need for HMAC
- $\mathrm{MAC} = \lfloor H(K \| M) \rfloor_{\mathtt{p}}$

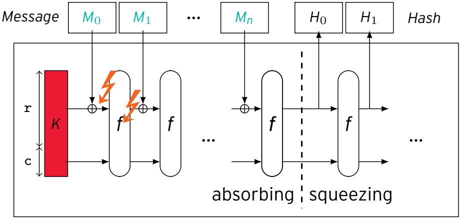


Always possible to consider a secret initial state.

MAC based on Keccak

- Sponge construction $\Rightarrow$ no need for HMAC
- $\mathrm{MAC} = \lfloor H(K \| M) \rfloor_{\mathrm{p}}$





Always possible to consider a secret initial state.

DSCA on MAC based on Keccak

DSCA on keyed Keccak:

- Zohner, Kasper, Stöttinger, Huss, [DATE 2012]: brief analysis among other SHA-3 candidates.
- Taha, Schaumont, [HOST 2013]: attack paths in function of key length.

DSCA on MAC based on Keccak

DSCA on keyed Keccak:

- Zohner, Kasper, Stöttinger, Huss, [DATE 2012]: brief analysis among other SHA-3 candidates.
- Taha, Schaumont, [HOST 2013]: attack paths in function of key length.

All prevented by first order masking.

Keccak First Order Masking Scheme

Already considered in SHA-3 submission:

- Boolean masking: $A = R \oplus S$
- Linear part: $\lambda(A) = \lambda(R) \oplus \lambda(S)$.
- Non-linear part: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$

$$r_x \leftarrow r_x + (r_{x+1} + 1) \cdot r_{x+2} + r_{x+1} \cdot s_{x+2}$$

$$s_x \leftarrow s_x + (s_{x+1} + 1) \cdot s_{x+2} + s_{x+1} \cdot r_{x+2}$$

Keccak First Order Masking Scheme

Already considered in SHA-3 submission:

- Boolean masking: $A = R \oplus S$
- Linear part: $\lambda(A) = \lambda(R) \oplus \lambda(S)$.
- Non-linear part: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$

$$r_x \leftarrow r_x + (r_{x+1} + 1) \cdot r_{x+2} + r_{x+1} \cdot s_{x+2}$$

$$s_x \leftarrow s_x + (s_{x+1} + 1) \cdot s_{x+2} + s_{x+1} \cdot r_{x+2}$$

Masking Scheme Improvement (Keccak team, 2012)

- Precompute $Y = S \oplus \lambda(S)$,
- Never change S.

Keccak First Order Masking Scheme

Already considered in SHA-3 submission:

- Boolean masking: $A = R \oplus S$
- Linear part: $\lambda(A) = \lambda(R) \oplus \lambda(S)$.
- Non-linear part: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$

$$\begin{aligned} r_x &\leftarrow r_x + (r_{x+1} + 1) \cdot r_{x+2} + r_{x+1} \cdot s_{x+2} \\ &\quad + (s_{x+1} + 1) \cdot s_{x+2} + s_{x+1} \cdot r_{x+2} \\ s_x &\leftarrow s_x \end{aligned}$$

Masking Scheme Improvement (Keccak team, 2012)

- Precompute $Y = S \oplus \lambda(S)$,
- Never change S.

Keccak First Order Masking Scheme

Already considered in SHA-3 submission:

- Boolean masking: $A = R \oplus S$
- Linear part: $\lambda(A) = (\lambda(R) \oplus Y) \oplus S$.
- Non-linear part: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$

$$\begin{aligned} r_x &\leftarrow r_x + (r_{x+1} + 1) \cdot r_{x+2} + r_{x+1} \cdot s_{x+2} \\ &\quad + (s_{x+1} + 1) \cdot s_{x+2} + s_{x+1} \cdot r_{x+2} \\ s_x &\leftarrow s_x \end{aligned}$$

Masking Scheme Improvement (Keccak team, 2012)

- Precompute $Y = S \oplus \lambda(S)$,
- Never change S.

Keccak First Order Masking Scheme

Already considered in SHA-3 submission:

- Boolean masking: $A = R \oplus S$
- Linear part: $\lambda(A) = (\lambda(R) \oplus Y) \oplus S$.
- Non-linear part: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$

$$\begin{aligned} r_x &\leftarrow r_x + (r_{x+1} + 1) \cdot r_{x+2} + r_{x+1} \cdot s_{x+2} \\ &\quad + (s_{x+1} + 1) \cdot s_{x+2} + s_{x+1} \cdot r_{x+2} \\ s_x &\leftarrow s_x \end{aligned}$$

Masking Scheme Improvement (Keccak team, 2012)

- Precompute $Y = S \oplus \lambda(S)$,
- Never change S.

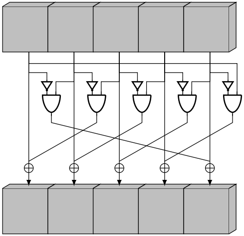
$\rightsquigarrow$ Possibility for collision-correlation.

Outline

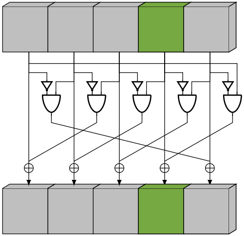
1. Algebraic Collision Attack
2. Collision Detection
3. Experiments
4. Conclusion

Collisions in the χ function

χ function: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$ $\qquad (r_x = a_x + s_x)$.



χ function: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$ $(r_x = a_x + s_x)$.

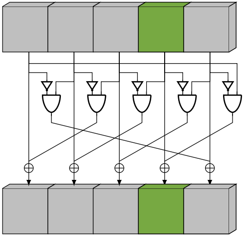


Bit collision:

$$a_x + s_x = a_x + (a_{x+1} + 1) \cdot a_{x+2} + s_x,$$

Collisions in the χ function

χ function: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$ $\quad (r_x = a_x + s_x)$.

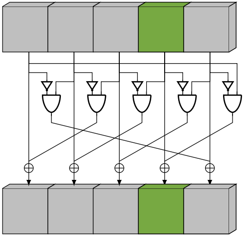


Bit collision:

$$a_x + s_x = a_x + (a_{x+1} + 1) \cdot a_{x+2} + s_x,$$
$$\Rightarrow \quad (a_{x+1} + 1) \cdot a_{x+2} = 0,$$

Collisions in the χ function

χ function: $a_x \leftarrow a_x + (a_{x+1} + 1) \cdot a_{x+2}$ $\quad$ $(r_x = a_x + s_x)$.



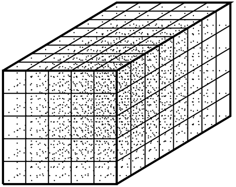
Bit collision first round:

$$a_x + s_x = a_x + (a_{x+1} + 1) \cdot a_{x+2} + s_x,$$
$$\Rightarrow \quad (a_{x+1} + 1) \cdot a_{x+2} = 0,$$

$$(\lambda(K \oplus M)_{x+1} + 1) \cdot \lambda(K \oplus M)_{x+2} = 0$$

First round: Each equation depend on 33 key bits.

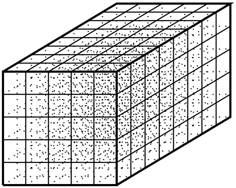
Building an Algebraic System



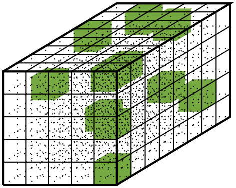
System of equations

Collisions in message

Building an Algebraic System

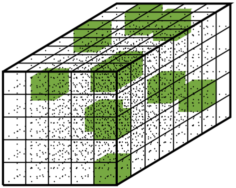


System of equations



Collisions in message

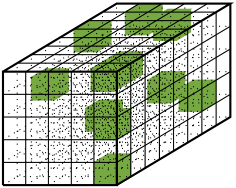
Building an Algebraic System



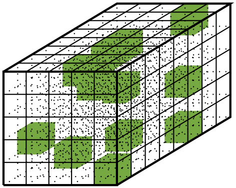
System of equations

Collisions in message

Building an Algebraic System

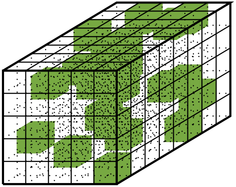


System of equations



Collisions in message

Building an Algebraic System

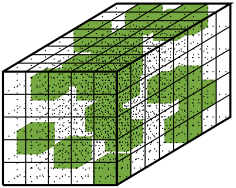


System of equations

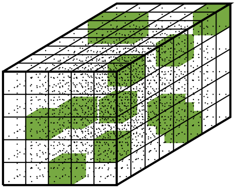
Collisions in message

oberthur TECHNOLOGIES THE M COMPANY

Building an Algebraic System

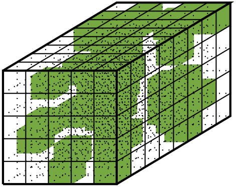


System of equations



Collisions in message

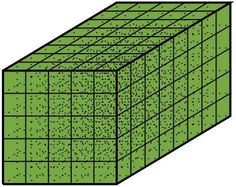
Building an Algebraic System



System of equations

Collisions in message

Building an Algebraic System

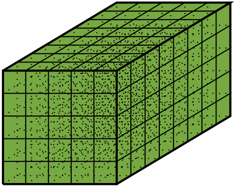


System of equations

Collisions in message

- 1600 variables and 1600+ equations...
- would take $\approx 2^{962}$ ops. for a random system

Building an Algebraic System

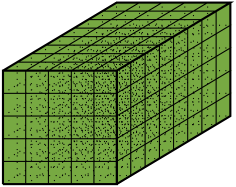


System of equations

Collisions in message

- 1600 variables and 1600+ equations...
- would take $\approx 2^{962}$ ops. for a random system
- solved in few minutes (Gröbner basis in Magma).

Building an Algebraic System



System of equations

Collisions in message

- 1600 variables and 1600+ equations...
- would take $\approx 2^{962}$ ops. for a random system
- solved in few minutes (Gröbner basis in Magma).

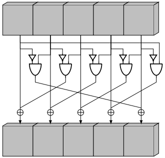
$\rightsquigarrow$ The system is heavily structured.

Considering Smaller Systems

Linearity of λ

By linearity, $\lambda(K \oplus M) = \lambda(K) \oplus \lambda(M) = K' \oplus M'$ where M' can be computed.

Linear change of variable $\Rightarrow$ reveals the structure: $5 \times w$ independent systems of 5 equations in 5 variables.

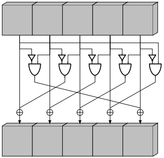


Considering Smaller Systems

Linearity of λ

By linearity, $\lambda(K \oplus M) = \lambda(K) \oplus \lambda(M) = K' \oplus M'$ where M' can be computed.

Linear change of variable $\Rightarrow$ reveals the structure: $5 \times w$ independent systems of 5 equations in 5 variables.



Easy to solve, even with exh. search.

oberthur TECHNOLOGIES THE M COMPANY

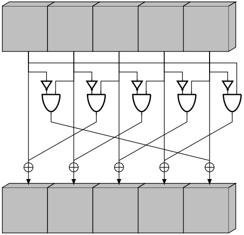
Attack Roadmap

1. Detect collisions from n different messages.
2. Build $5 \times w$ small systems $\mathcal{F}_{y,z}$.
3. Solve each system $\mathcal{F}_{y,z} \Rightarrow \mathcal{V}_{y,z}$.
4. Build all candidates K' from $\mathcal{V}_{y,z}$.
5. Compute $K = \lambda^{-1}(K')$ for all K'. Exhaustively search the correct key.

Attack Roadmap

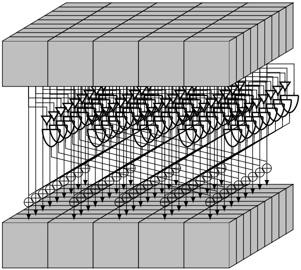
1. Detect collisions from n different messages.
2. Build $5 \times w$ small systems $\mathcal{F}_{y,z}$.
3. Solve each system $\mathcal{F}_{y,z} \Rightarrow \mathcal{V}_{y,z}$.
4. Build all candidates K' from $\mathcal{V}_{y,z}$.
5. Compute $K = \lambda^{-1}(K')$ for all K'. Exhaustively search the correct key.

Collision Detection by Correlation



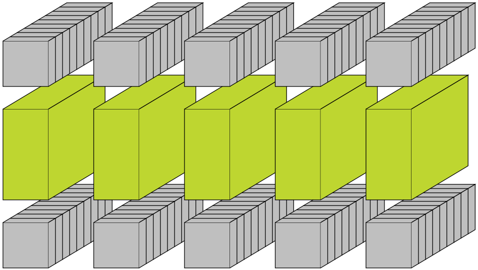
- Difficult to detect a collision on a bit;

Collision Detection by Correlation



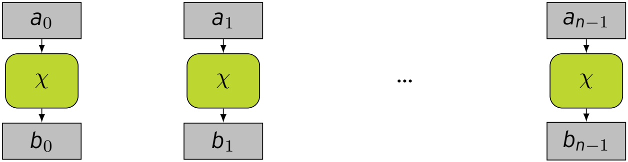
- Difficult to detect a collision on a bit;
- χ is usually processed by machine word (ℓ bits);

Collision Detection by Correlation

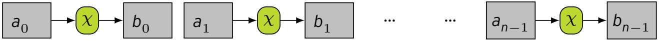


- Difficult to detect a collision on a bit;
- χ is usually processed by machine word (ℓ bits);
- Consider each word independently.

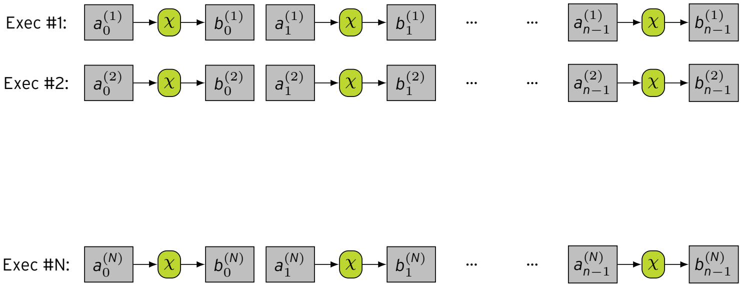
Collision Detection by Correlation (cont'd)



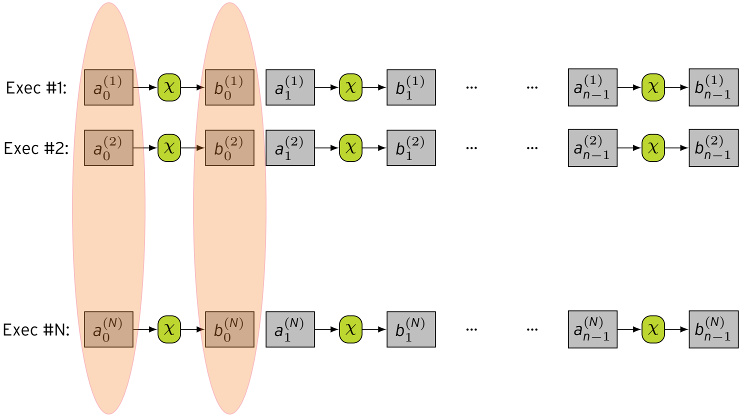
Collision Detection by Correlation (cont'd)



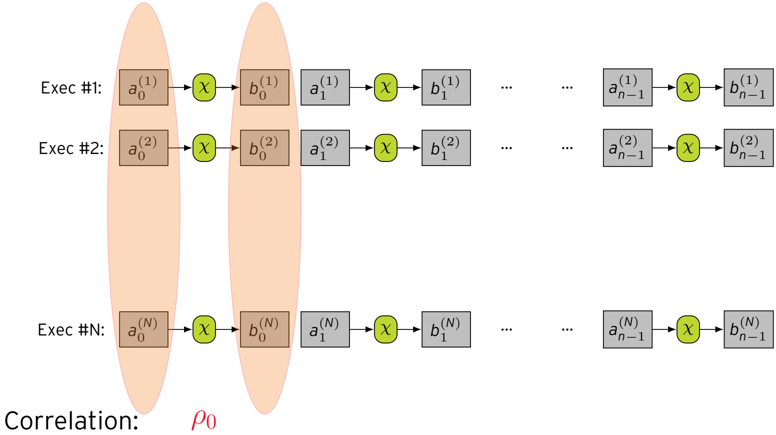
Collision Detection by Correlation (cont'd)



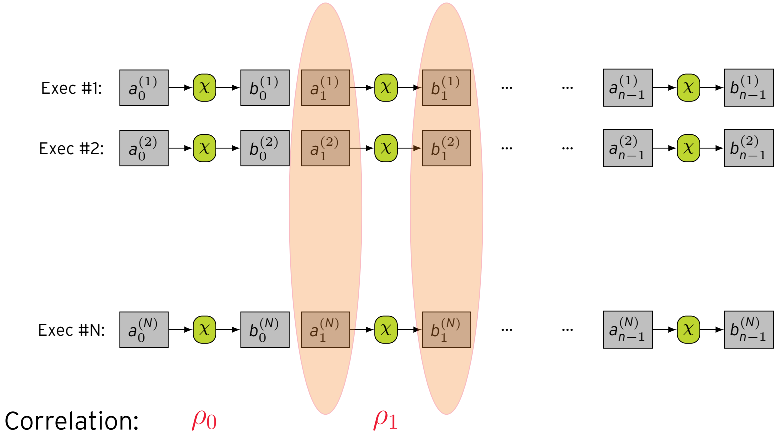
Collision Detection by Correlation (cont'd)



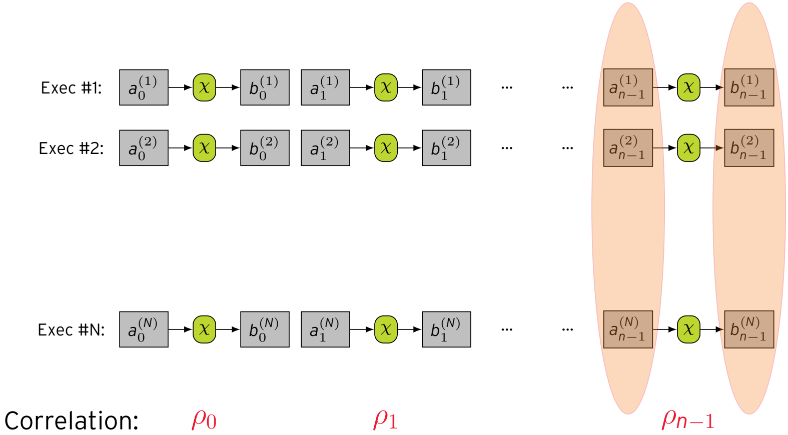
Collision Detection by Correlation (cont'd)



Collision Detection by Correlation (cont'd)

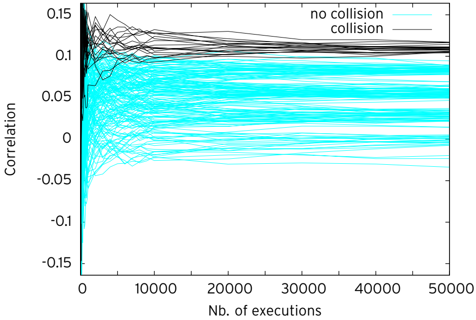


Collision Detection by Correlation (cont'd)



Recognize a Collision (HW leakage model)

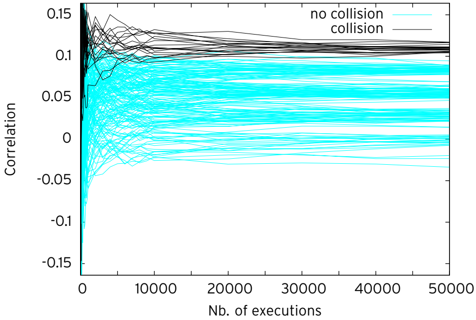
Correlations for one message $\ell = 8$, noise $\sigma = 4$.



- Correlations appear by packs
- More top correlations

Recognize a Collision (HW leakage model)

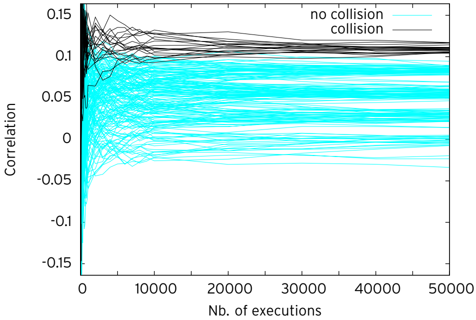
Correlations for one message $\ell = 8$, noise $\sigma = 4$.



- Correlations appear by packs $\Rightarrow$ related to HW
- More top correlations

Recognize a Collision (HW leakage model)

Correlations for one message $\ell = 8$, noise $\sigma = 4$.



- Correlations appear by packs $\Rightarrow$ related to HW
- More top correlations $\Rightarrow$ High collision probability

Required Number of Messages

Attack Behavior depending on #messages, $\ell = 8$ (average):

#messages	0	1	20	50	60
#collisions	0	20.4	177.3	199.0	199.7
#equations	0	0.5	7.4	12.2	13.1
#candidates	2^{1600}	2^{1558}	2^{589}	2^{78}	2^{38}
#messages	70	80	90	140	170
#collisions	199.8	199.9	200	200	200
#equations	13.6	14.0	14.3	14.9	15.0
#candidates	2^{20}	379.0	19.4	1.1	1

Required Number of Messages

Attack Behavior depending on #messages, $\ell = 8$ (average):

#messages	0	1	20	50	60
#collisions	0	20.4	177.3	199.0	199.7
#equations	0	0.5	7.4	12.2	13.1
#candidates	2^{1600}	2^{1558}	2^{589}	2^{78}	2^{38}

70	80	90	140	170
199.8	199.9	200	200	200
13.6	14.0	14.3	14.9	15.0
2^{20}	379.0	19.4	1.1	1

- Only 70 *different* messages are needed;

Required Number of Messages

Attack Behavior depending on #messages, $\ell = 8$ (average):

#messages	0	1	20	50	60
#collisions	0	20.4	177.3	199.0	199.7
#equations	0	0.5	7.4	12.2	13.1
#candidates	2^{1600}	2^{1558}	2^{589}	2^{78}	2^{38}
#messages	70	80	90	140	170
#collisions	199.8	199.9	200	200	200
#equations	13.6	14.0	14.3	14.9	15.0
#candidates	2^{20}	379.0	19.4	1.1	1

- Only 70 *different* messages are needed;
- Each message hashed several times.

Conclusion

Summary

- Combining collision correlation and algebraic attack;
- High collision probability helps our attack;
- Efficient detection in random leakage model.

Conclusion

Summary

- Combining collision correlation and algebraic attack;
- High collision probability helps our attack;
- Efficient detection in random leakage model.

- Other collision detection techniques ?
- Find an *efficient* masking scheme ?

Collision Probability

Each input bit of χ has a probability $\frac{3}{4}$ to collide with its output bit.

$$a_x = a_x + (a_{x+1} + 1) \cdot a_{x+2} \,.$$

Collision Probability

Each input bit of χ has a probability $\frac{3}{4}$ to collide with its output bit.

$$(a_{x+1} + 1) \cdot a_{x+2} = 0\,.$$

Collisions in Keccak

Collision Probability

Each input bit of χ has a probability $\frac{3}{4}$ to collide with its output bit.

$$(a_{x+1} + 1) \cdot a_{x+2} = 0 \,.$$

Probability of (at least one) collision in a message:

bit-size ℓ	8	16	32
prob. collision	≈ 1	0.635	0.005

High probability of collision $\Rightarrow$ no threshold for detection.

Considering Other Leakage Models

Experimental Framework

Leakage function L, let $a = \sum_{i=0}^{\ell-1} a_i$:

- *quad* leakage: $L(a) = f_{\text{quad}}(a_0, \ldots, a_{\ell-1})$, $\deg f_{\text{quad}} = 2$;
- *full*: leakage: $L(a) = f_{\text{full}}(a_0, \ldots, a_{\ell-1})$, $\deg f_{\text{full}} = \ell$.

Noise level: keep constant Signal to Noise Ratio (SNR).

Considering Other Leakage Models

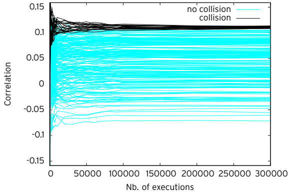
Experimental Framework

Leakage function L, let $a = \sum_{i=0}^{\ell-1} a_i$:

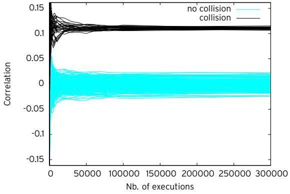
- *quad* leakage: $L(a) = f_{\mathrm{quad}}(a_0, \ldots, a_{\ell-1})$, $\deg f_{\mathrm{quad}} = 2$;
- *full*: leakage: $L(a) = f_{\mathrm{full}}(a_0, \ldots, a_{\ell-1})$, $\deg f_{\mathrm{full}} = \ell$.

Noise level: keep constant Signal to Noise Ratio (SNR).

Correlations for one message $\ell = 8$, SNR=0.125.



quad.



full.

Comparison to 2O DSCA

2O-DSCA on Keccak

- Correlation between χ input/output ℓ bits words;
- Normalized product as combination function;
- HW leakage hypothesis.

Comparison to 2O DSCA

2O-DSCA on Keccak

- Correlation between χ input/output ℓ bits words;
- Normalized product as combination function;
- HW leakage hypothesis.

Comparison to 2O-DSCA for $\ell = 8$, SNR=0.125:

	HW	*quad*	*full*
This attack	$315\,000 \times 70$	$200\,000 \times 70$	$5\,000 \times 70$
2O-DSCA	$600\,000$	$> 1\,500\,000$	$> 1\,500\,000$