Side-Channel Analysis of the SHA-3 Finalists on SASEBO

CryptArchi 2012

Julien Francq, Jean-Baptiste Rigaud and Antoine Wurcker

antoine.wurcker.external@cassidian.com

CASSIDIAN CyberSecurity, ENSMSE

2012, June the 20th

What is SHA?

Cryptographic Hash Functions

- play a fundamental role in modern cryptography
  - data integrity, message authentication, *etc.*
- map an arbitrary finite length bitstring to a fixed length digest
- should have desirable properties
  - preimage and collision resistance

- NIST Hash Function Standard = Secure Hash Algorithm
- Cryptanalysis of previous SHA families $\Rightarrow$ SHA-3 Contest

SHA-3 Contest

- Similar to the past AES one.
- 11/2/2007: kick-off.
- 11/31/2008: 64 candidates submitted.
- 12/10/2008: 51 accepted in the 1st round.
- 07/24/2009: 14 semifinalists.

Selection Criteria

- Security
  - against mathematical cryptanalysis
  - against Side-Channel Attacks (SCA), *e.g.* ElectroMagnetic Analysis (EMA)
- Software/hardware cost (ASIC, FPGA)
- Flexibility

- 12/09/2010: 5 finalists
  - BLAKE, Grøstl, JH, Keccak, Skein
- Summer 2012: and the winner is?

Litterature and Remaining Questions

Only a few studies of SCA on SHA primitives

- SHA-1 [Lemke *et al.*, CHES 2004], SHA-2 [McEvoy *et al.*, WISA 2007]
- SHA-3
  - [Benoît *et al.*, CHES 2010], [Zohner *et al.*, TrustED 2011], [Boura *et al.*, TrustED 2012]: SCA on software implementations.

Remaining Questions

For each finalist:

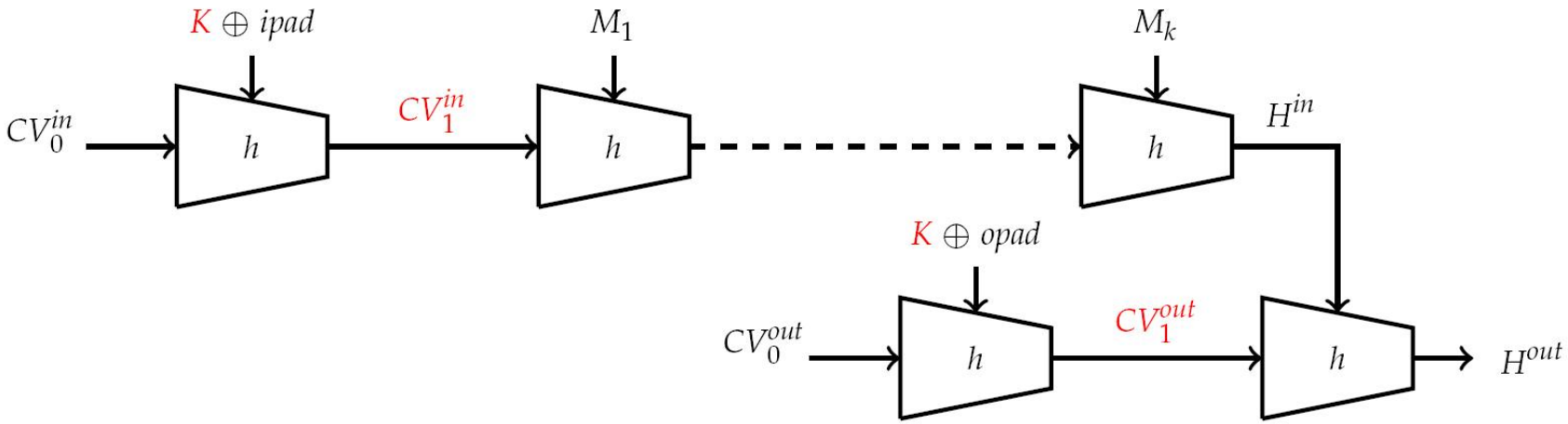
- What is the cost of HMAC-SHA-3 implementations?
- What is the cost of a 1^{st} (and higher)-order masking scheme?
- Does it protect well? $\Rightarrow$ SASEBO
- EMA $\neq$ Power Analysis (PA)? Software $\neq$ Hardware?
- What are the consequences for SHA-3 Contest?

Outline

1. Our HMAC-SHA-3 Implementations
2. Our 1^{st}-Order Masking Schemes
3. Check on SASEBO
4. Conclusion and Future Works

HMAC and SCA

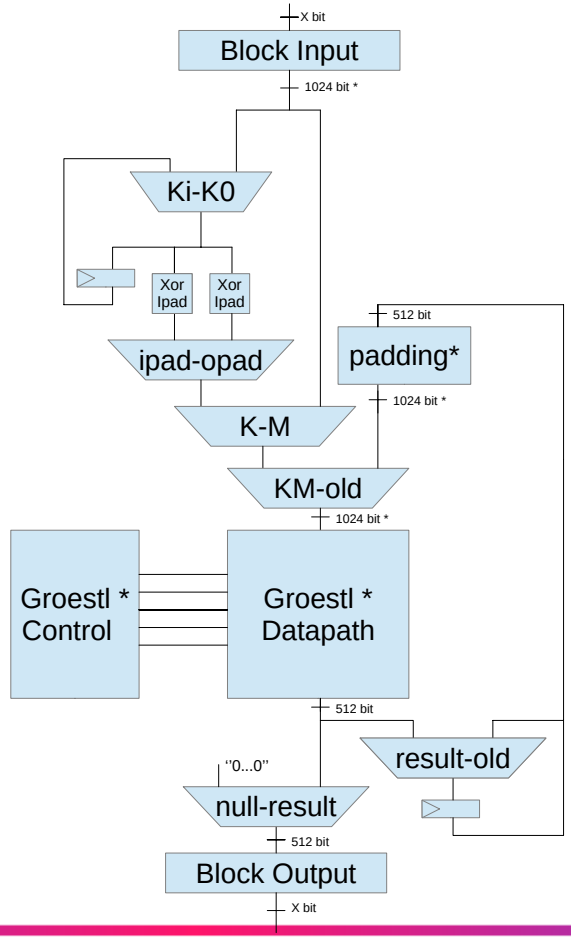
- HMAC processes a key $\rightarrow$ Retrieve it: forge correct MAC
- $HMAC(K, M) = H((K \oplus opad) \| H((K \oplus ipad) \| M))$



- Possible targets of a SCA: K, CV_1^{in} and CV_1^{out}
- Before SCA study, we implemented SHA-3 finalists in HMAC mode
  - Starting point: [Francq *et al.*, CryptArchi 2011]

Overhead for HMAC

- Area Overhead:
  - Registers for K, CV, $opad = 0x5c5c\ldots$ and $ipad = 0x3636\ldots$
  - XORs, Muxes and more complex FSM



Implementation Results for HMAC

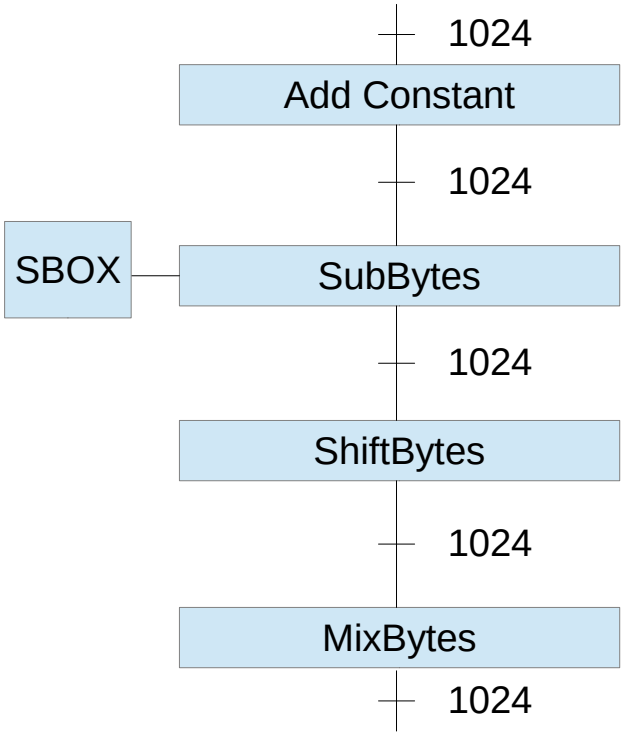
- VHDL, SASEBO GII (Xilinx Virtex-5 VLX-30), ISE Tools (v14.1i)
- Post-place-and-route results

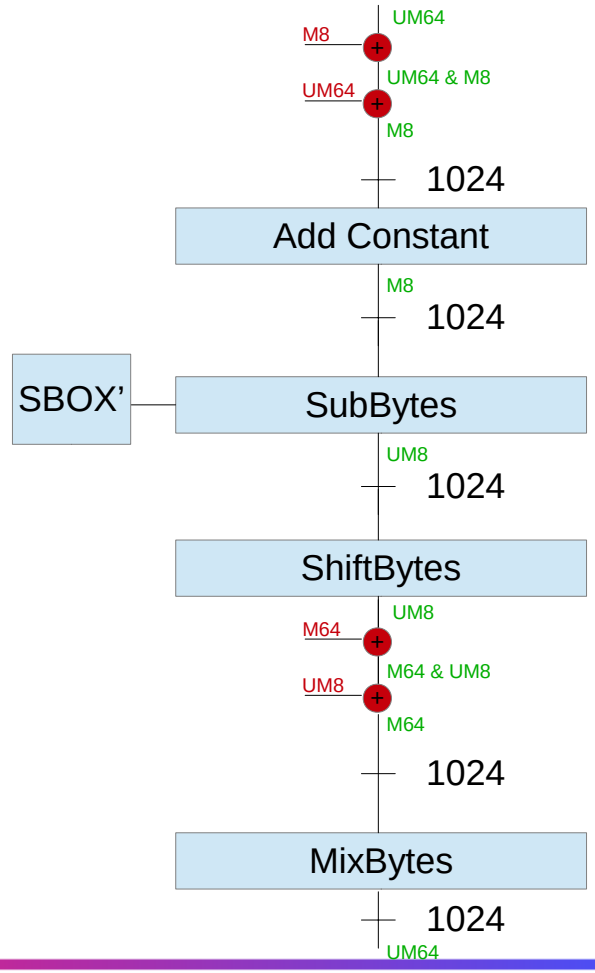
Circuit	Area (slices)	Freq. (MHz)
BLAKE-4G/6	1911	125
HMAC-BLAKE-4G/6	2538	125
Grøstl-256-512 ($P + Q$)	3248	167
HMAC-Grøstl-256-512 ($P + Q$)	3817	167
Grøstl-256 ($P + Q$)	1551	202
HMAC-Grøstl-256 ($P + Q$)	1751	180
JH_u2	2378	142
HMAC-JH_u2	2636	142
Keccak	−−	−−
Skein-256	996	69
HMAC-Skein-256	1021	62

Outline

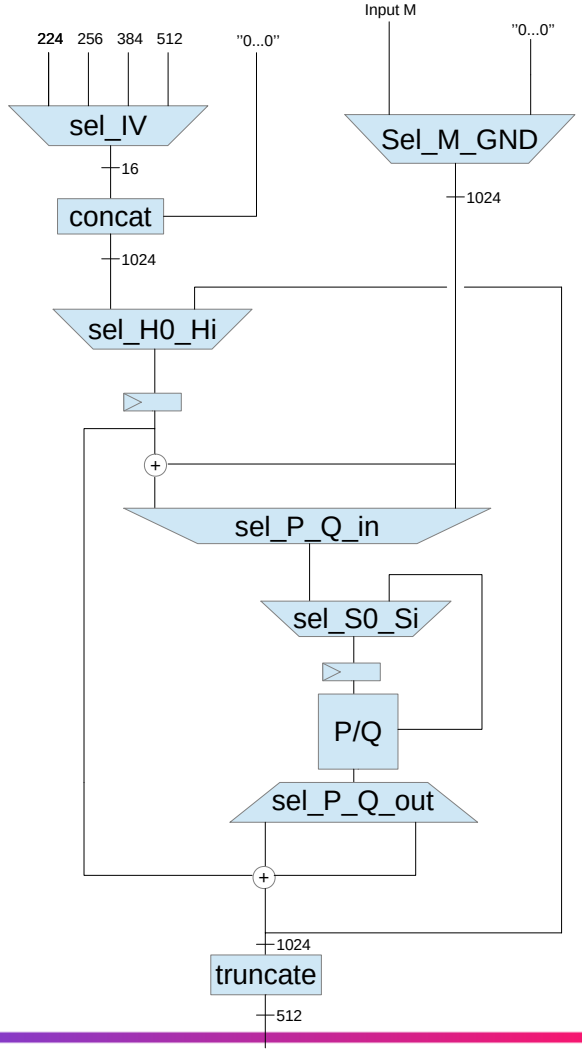
1. Our HMAC-SHA-3 Implementations
2. Our 1st-Order Masking Schemes
3. Check on SASEBO
4. Conclusion and Future Works

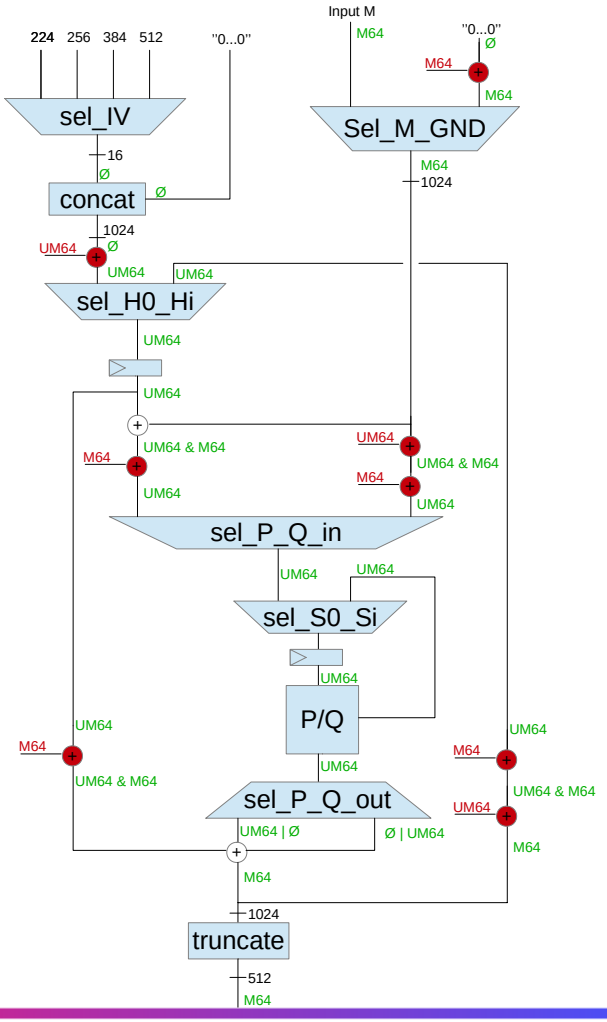
Masked Grøstl Round



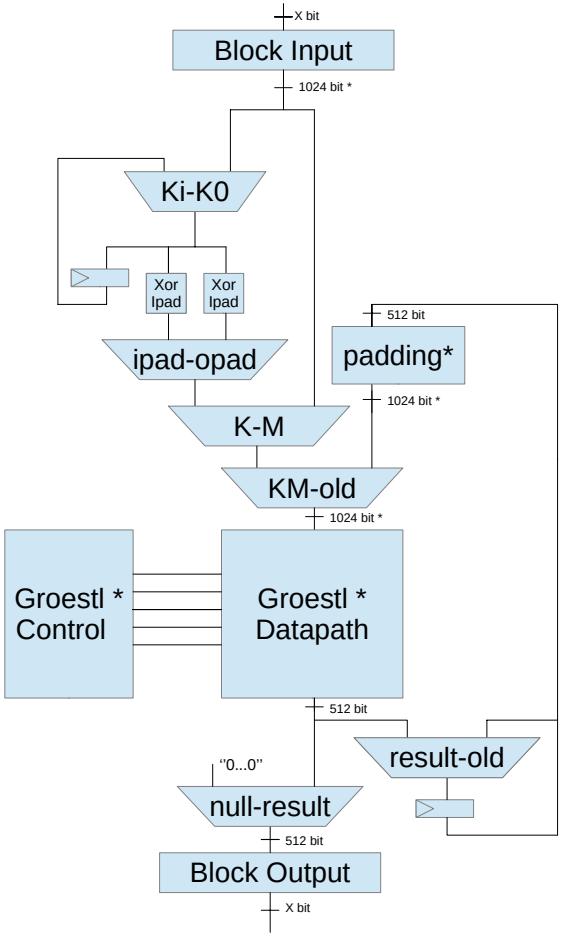


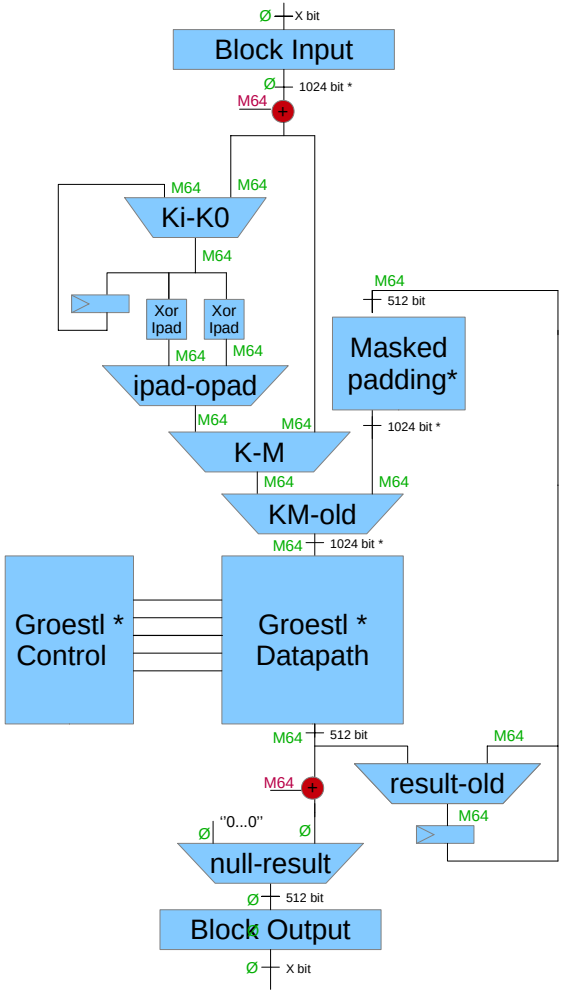
Masked Grøstl Datapath





Masked HMAC SHA-3 Finalist





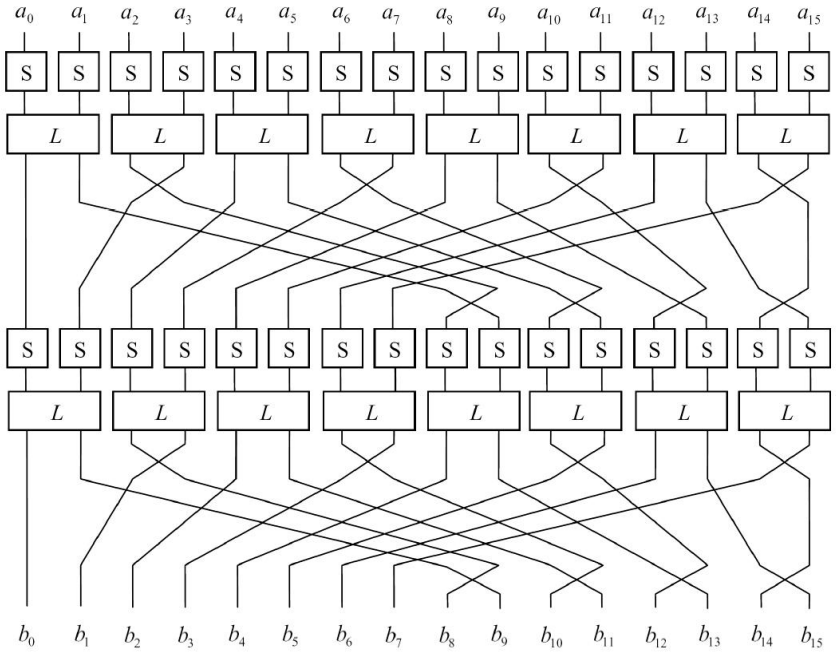
Implementation Results for Grøstl

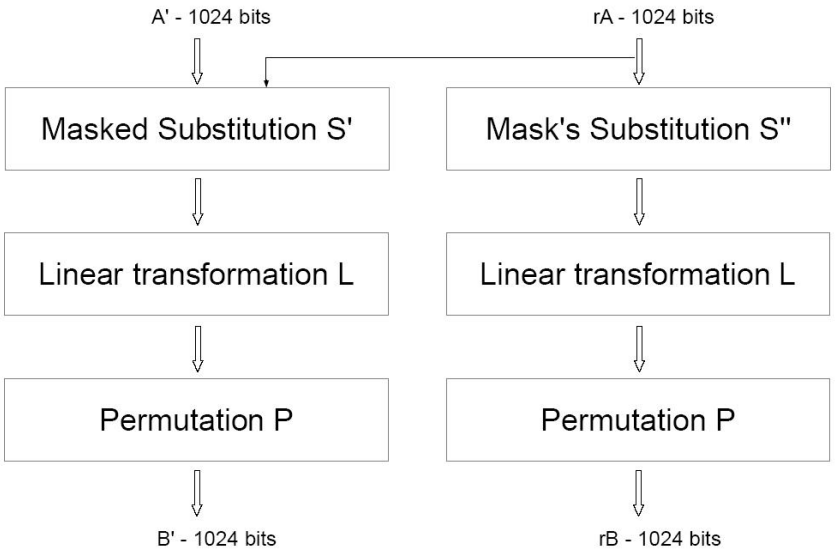
- VHDL, SASEBO GII (Xilinx Virtex-5 VLX-30), ISE Tools (v14.1i)
- Only 4800 slices available!
- Post-place-and-route results

Circuit	Area (slices)	Freq. (MHz)
Grøstl-256-512 $(P+Q)$	3248	167
HMAC-Grøstl-256-512 $(P+Q)$	3817	167
Masked-HMAC-Grøstl-256-512 $(P+Q)$	$>$ 4800	−−
Grøstl-256 $(P+Q)$	1551	202
HMAC-Grøstl-256 $(P+Q)$	1751	180
Masked-HMAC-Grøstl-256 $(P+Q)$	2895	167

- Area overhead: 65%, Time overhead: 7%

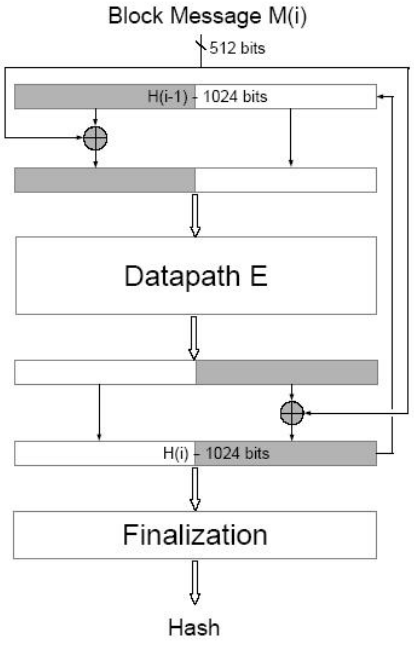
Masked JH Round

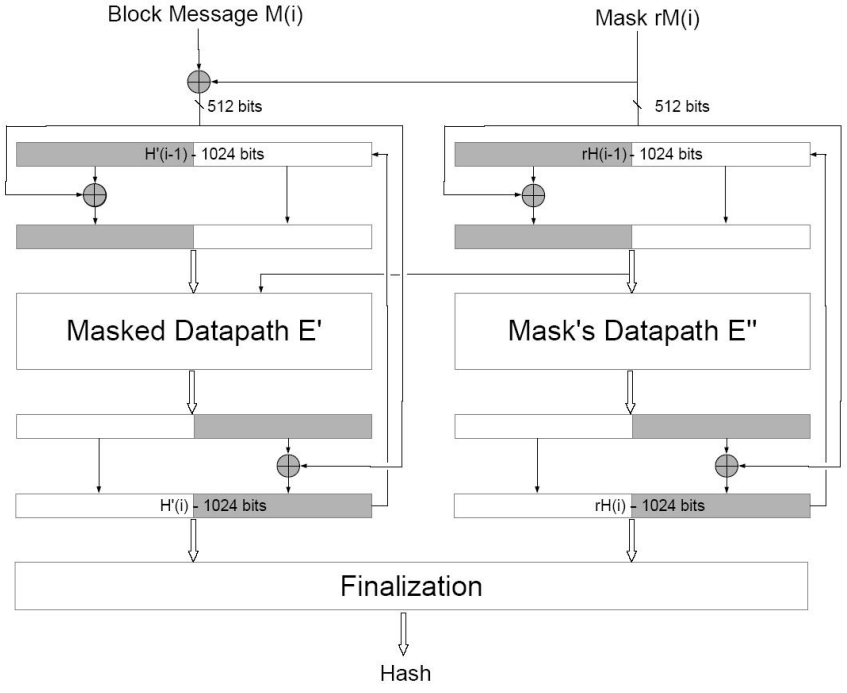




- JH bitsliced version, masked logical operations [Golić, IEEE 2007]
  - *e.g.*: $z = x \wedge y = z' \oplus r_x$
    $\Rightarrow z' = \wedge'(x', y', r_x, r_y)$
    $= \bar{y'} \wedge (\bar{r_y} \wedge r_x \vee r_y \wedge x') \vee y' \wedge (r_y \wedge r_x \vee \bar{r_y} \wedge x')$

Masked JH Datapath





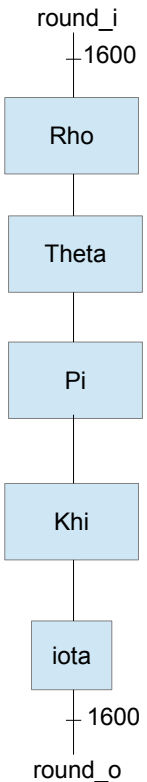
Implementation Results for JH

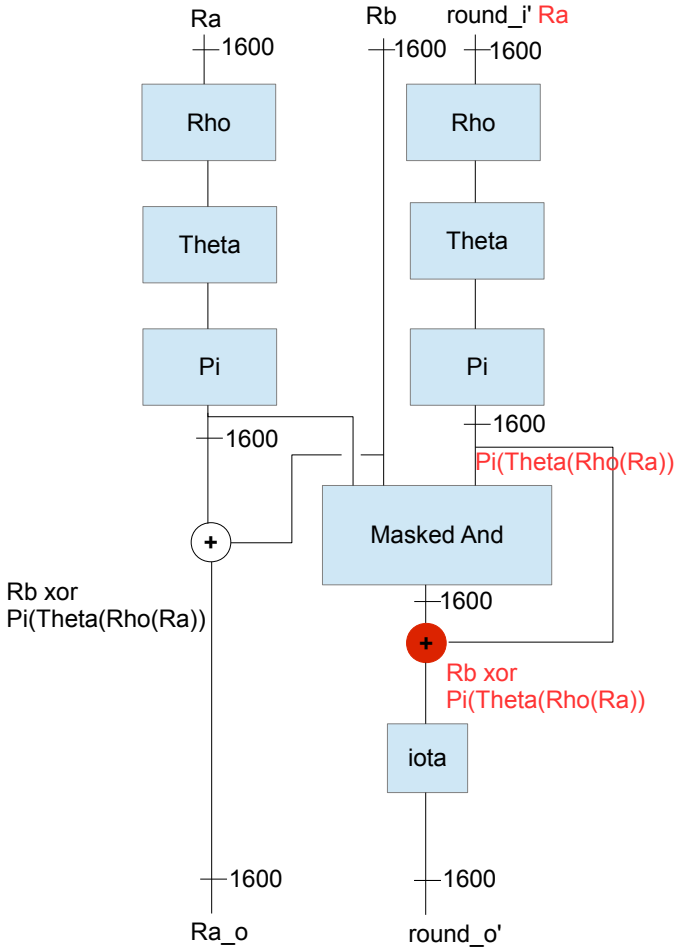
- VHDL, SASEBO GII (Xilinx Virtex-5 VLX-30), ISE Tools (v14.1i)
- Post-place-and-route results

Circuit	Area (slices)	Freq. (MHz)
JH_u2	2378	142
HMAC-JH_u2	2636	142
Masked-HMAC-JH_u2	4774	116

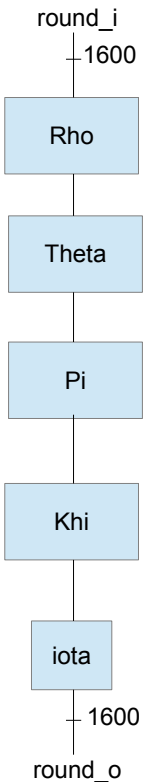
- 4774/4800 slices (phew!)
- Area overhead: 81%, Time overhead: 18%

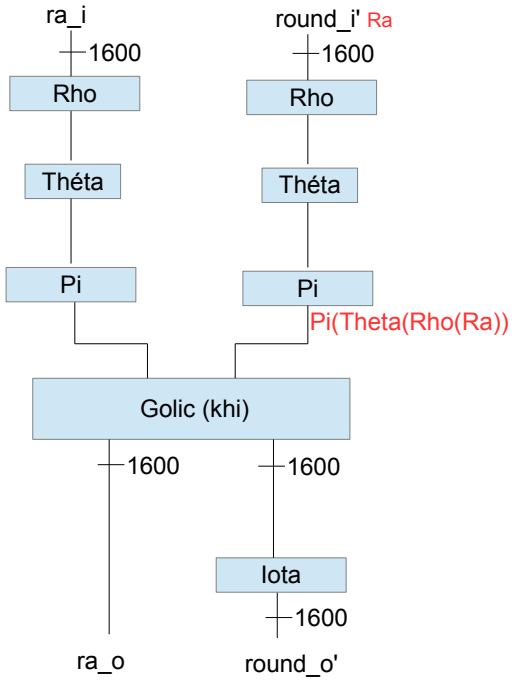
Masked Keccak Round (1st Version)



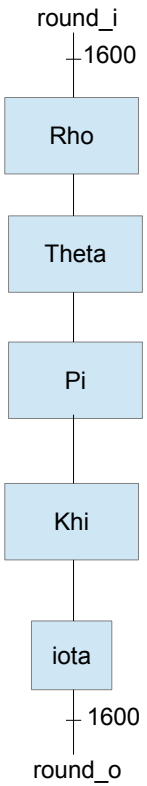


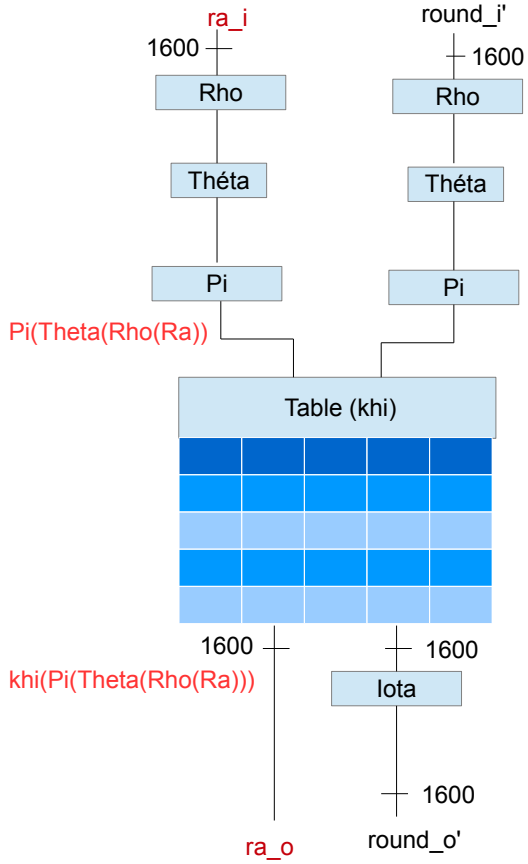
Masked Keccak Round (Golić)



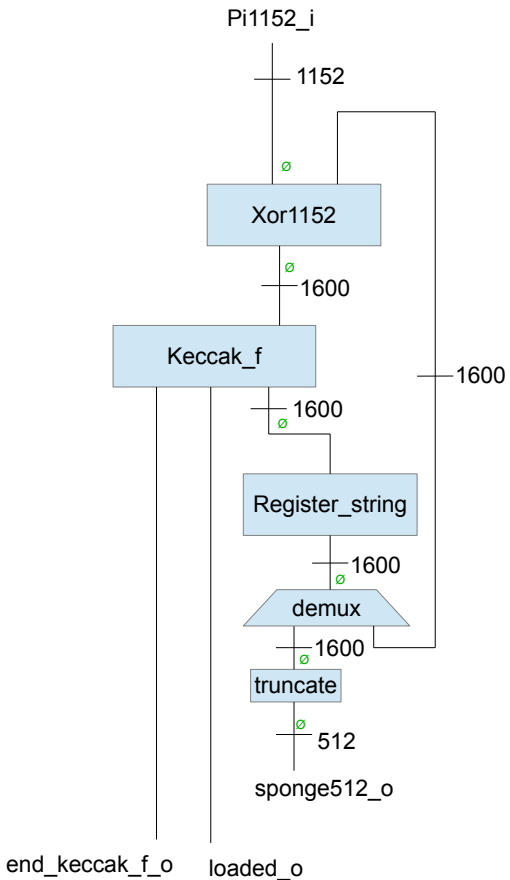


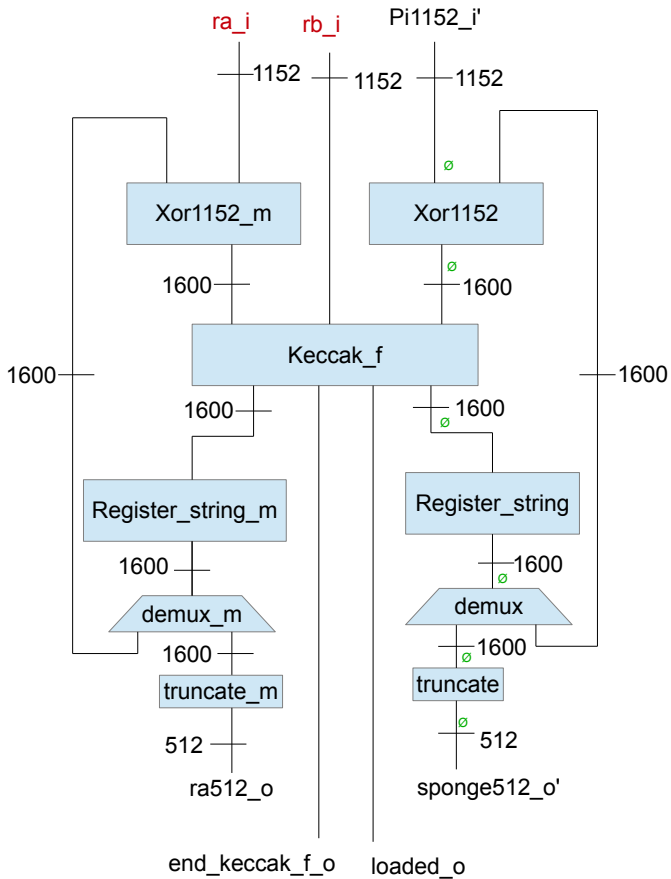
Masked Keccak Round (Tables)





Masked Keccak Datapath





Implementation Results for Keccak

- VHDL, SASEBO GII (Xilinx Virtex-5 VLX-30), ISE Tools (v14.1i)
- Post-place-and-route results

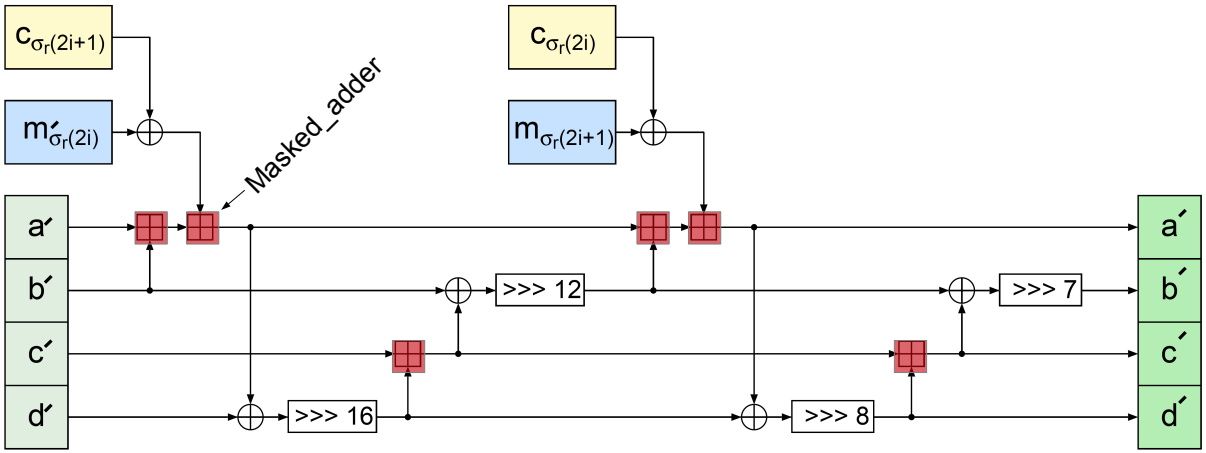
Circuit	Area (slices)	Freq. (MHz)
Keccak	2561	105
Keccak_v1.0	4124 (+61%)	95 (+9%)
Keccak_Golić	4789 (+86%)	92 (+12%)
Keccak_Tables	4709 (+83%)	104 (∼0%)

- 4709, 4789/4800 slices (phew again!)

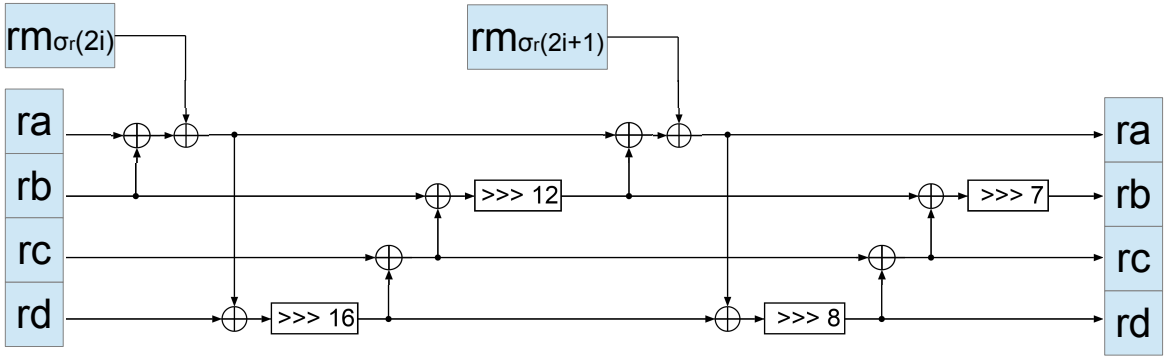
Masking Addition-Rotation-XOR schemes

- Arithmetic Masking (A2B) $\longleftrightarrow$ Boolean (B2A)
- [Messerges, FSE 2000] attacked by [Coron *et al.*, CHES 2000]
- [Goubin, CHES 2001]
  - B2A: 7 op.
  - A2B: $(5K + 5)$ op., where K is the size of the processor registers.
- [Coron *et al.*, CHES 2003]
  - Table-based method
  - A2B: 32-bit conversions on 8-bit CPUs 4.7 times faster than [Goubin, CHES 2001].
- [Neisse *et al.*, CHES 2004]
  - A2B: half the memory for tables compared to [Coron *et al.*, CHES 2003].
- [Golić, IEEE 2007], [McEvoy *et al.*, WISA 2007]
  - Masked circuits
  - More suitable for hardware implementations

Masked BLAKE Round

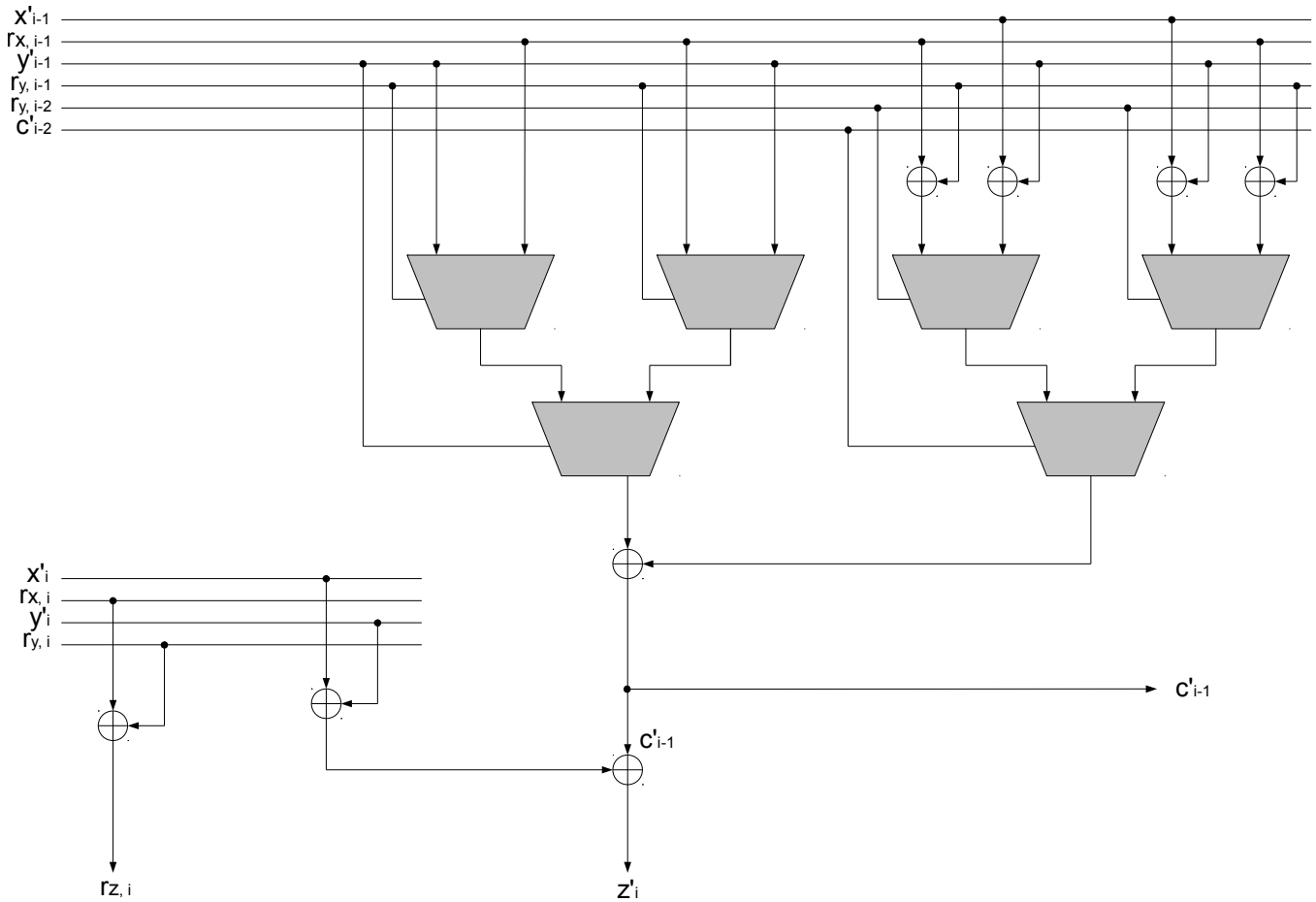


Blake masked Datapath

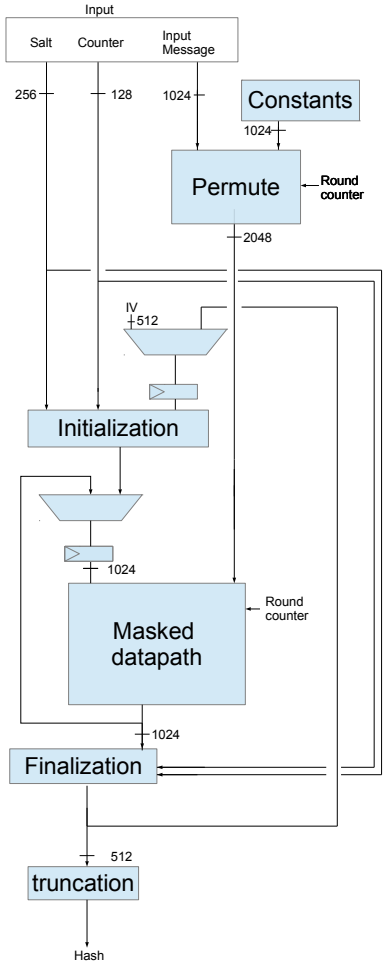


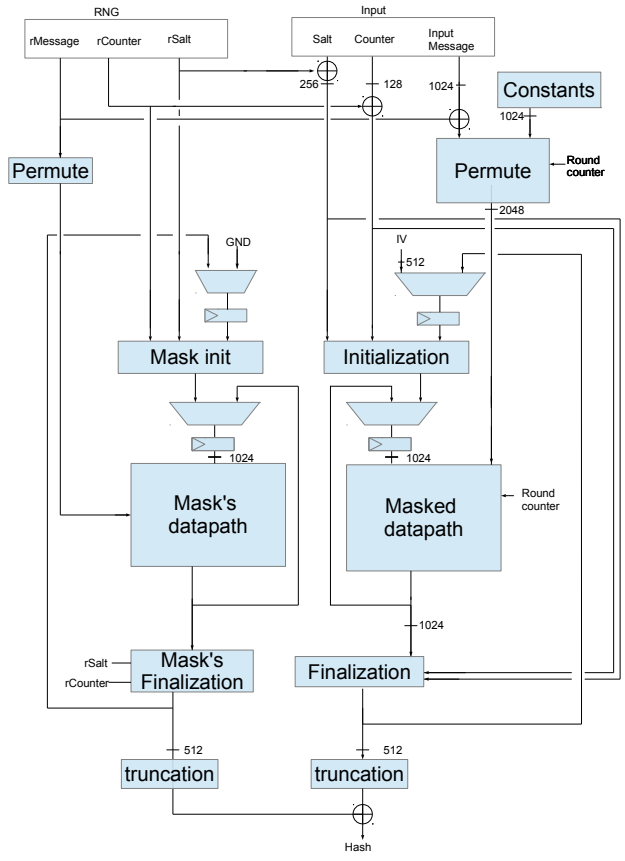
Blake mask's Datapath

Masked Adder [Golić, IEEE 2007]



Masked BLAKE





Implementation Results for BLAKE

- VHDL, SASEBO GII (Xilinx Virtex-5 VLX-30), ISE Tools (v14.1i)
- Only 4800 slices available!
- Post-place-and-route results

Circuit	Area (slices)	Freq. (MHz)
BLAKE-4G/6	1911	125
HMAC-BLAKE-4G/6	2538	125
Masked-HMAC-BLAKE-4G/6	3966 (+56%)	83 (+33%)

- Throughput divided by 6 compared to BLAKE-4G!

Outline

1. Our HMAC-SHA-3 Implementations
2. Our 1^{st}-Order Masking Schemes
3. Check on SASEBO
4. Conclusion and Future Works

Correlation Analysis

- Selection function: $w = f(cv, m)$
- The theoretical correlation between a data set x_i for a key guess j and the data set y_i for an arbitrary real key r is:

$$c(j, r) = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2} . \sqrt{\sum (y_i - \overline{y})^2}}$$

- Assuming a leakage in the Hamming Distance (HD) model:

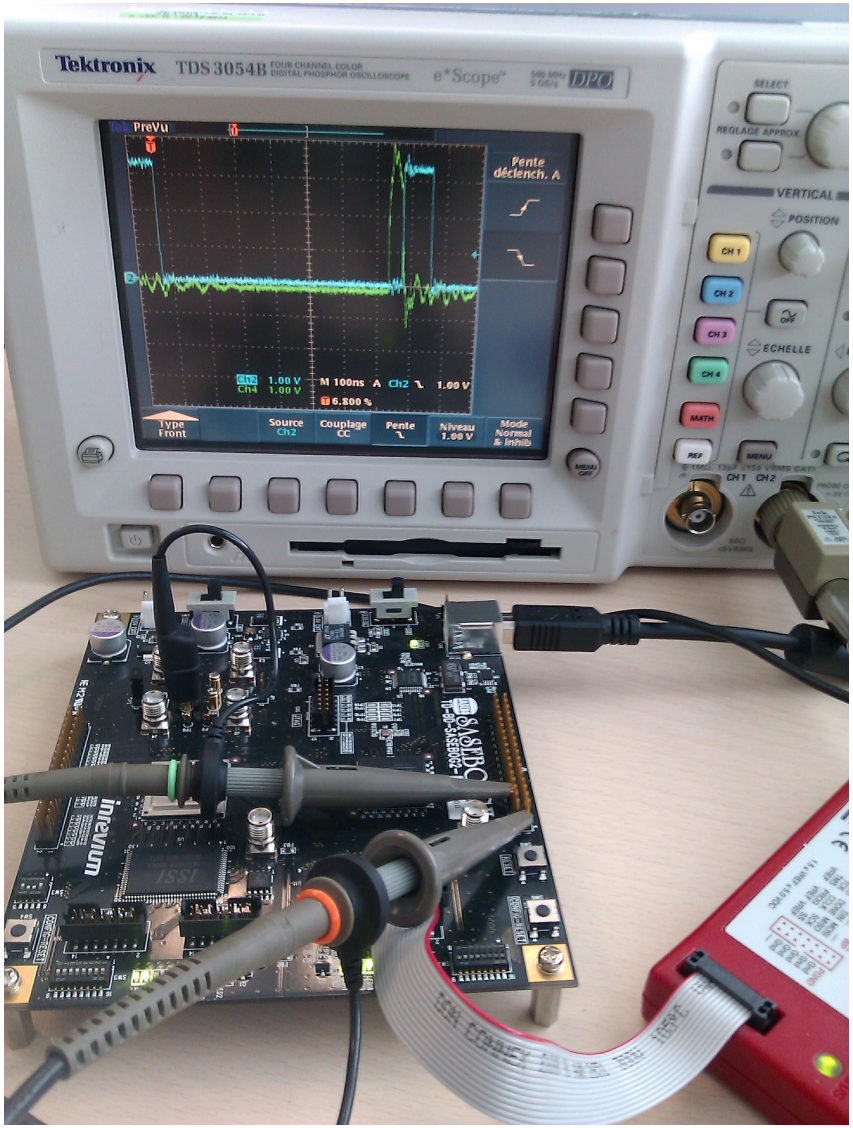
$$x_i = HD(f(j, m_i)) \text{ and } y_i = HD(f(r, m_i))$$

- Given a selection function, it is possible to compute $c(j, r)$ for all key guess and look at the correlation contrast between the real key and the wrong keys

SHA-3 Selection Functions

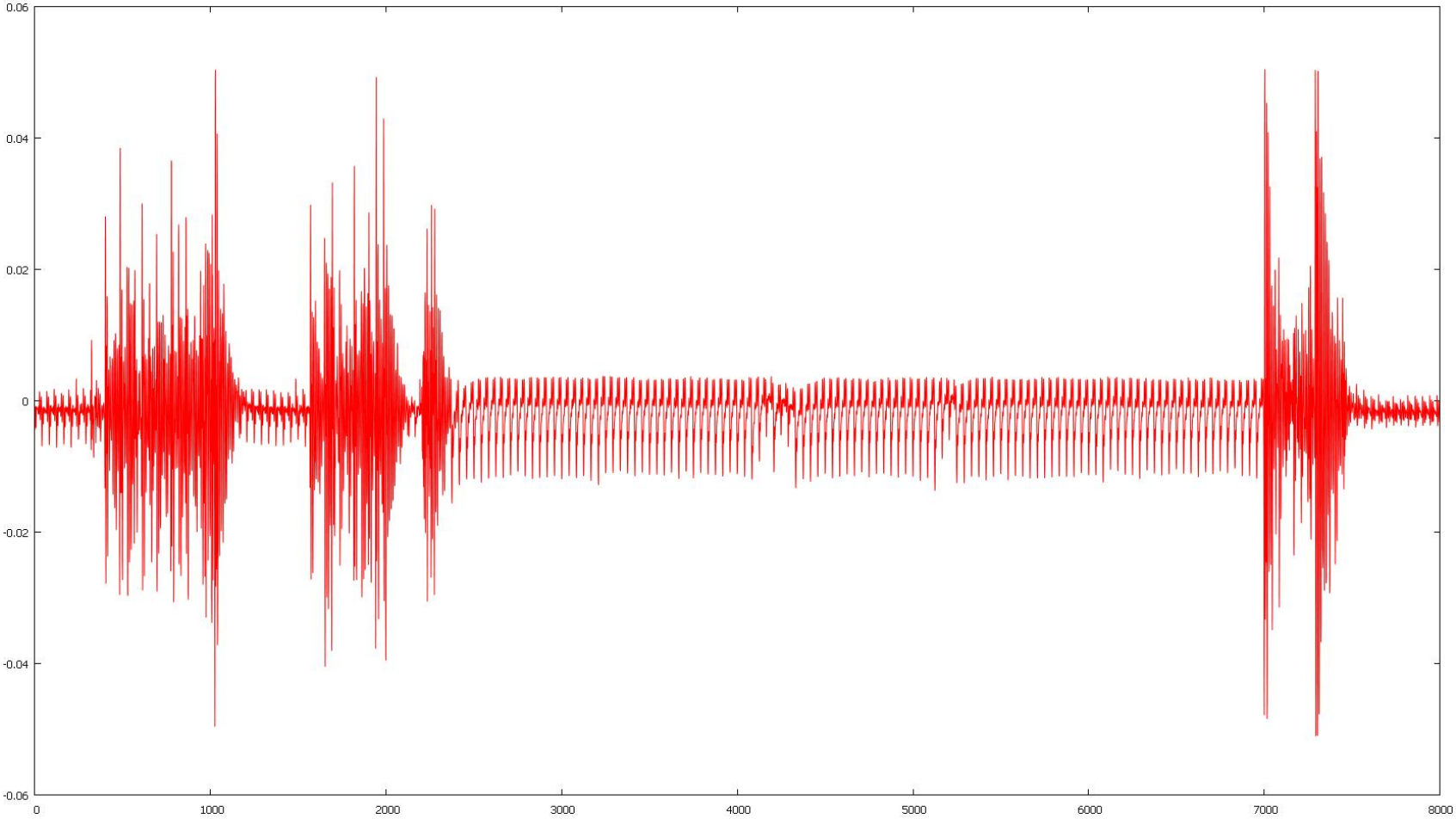
- BLAKE: modular addition
  - $w = (cv \boxplus m) \bmod 256$
- Grøstl: SBox ($256 \rightarrow 256$ substitution)
  - $w = SBOX_{Grøstl}(cv \oplus m)$
- JH: SBoxes ($16 \rightarrow 16$ substitution) S_0 and/or S_1
- Keccak
  - XOR in θ permutation
- Skein: modular addition
  - $w = K \boxplus m$

Measurement Platform



- SASEBO-GII
- Tektronix TDS3054B, 500 MHz, 5 GS/s
- Voltage probes: Tek P6139A, 500 MHz
- 128 curves/message
- 8640 averaged curves/day

SPA on HMAC-Grøstl



Grøstl Side-Channel Analysis

- Internal state after the AES SBox operation during first round of P_G

$$w'[b] = SBox(m[b] \oplus CV[b])$$

- 64 AES SBox SCAs to retrieve CV (straightforward)
- It is possible to speed up the attack by a factor 64 by choosing all $m[b]$ equals
- CryptArchi 2012 is coming 1 week too soon...
- ...We have the curves but not enough time to mount the attack

Outline

1. Our HMAC-SHA-3 Implementations
2. Our 1^{st}-Order Masking Schemes
3. Check on SASEBO
4. Conclusion and Future Works

Conclusion and Future Works

- For HMAC:
  - Reasonable overhead
  - Time consuming for implementers (often no KATs)
- For SCA protection of non-ARX finalists:
  - $65\% <$ Area overhead $< 86\%$
  - $0\% <$ Time overhead $< 18\%$
- For SCA protection of ARX finalists:
  - Less efficient to protect in FPGAs
  - Will it be taken into account by NIST for its choice?

Calendar

- July, 2012:
  - Implement HMAC-Keccak
  - Implement SCA-protected Skein,
  - Mount SCA on the 5 finalists (the bench with SASEBO now works!),
  - Publish our results.
- For allowing public scrutiny, VHDL sources and ISE projects and curves will be available on SAPHIR2 website
- March, 2013: End of SAPHIR2 project.
- Significant contribution in the benchmarking of the SHA-3 finalists

Thank you!

Thanks for your attention and your future questions, comments, suggestions, etc.!

SAPHIR2 Project

- SAPHIR2 project supported by the French Agence Nationale de la Recherche (ANR-08-VERS-014)
  - Security and Analysis of Primitives of Hashing Innovatory and Recent 2
- **Partners:** ANSSI, CASSIDIAN (CSCSC), Cryptolog International, France Télécom, Gemalto, INRIA, LIENS, Morpho, UVSQ.
- **Goals:**
  1. follow and participate to the NIST SHA-3 Contest (cryptanalysis, implementations, optimizations, etc.),
  2. to support the candidates proposed by the SAPHIR projects 1 and 2.

SAPHIR2 Partners

CRYPTOLOG
Créateur de confiance

ANSSI

EADS
DEFENCE
& SECURITY

Ens

france telecom

gemalto
security to be free

INRIA

Sagem Sécurité
SAFRAN Group

UNIVERSITÉ DE VERSAILLES
SAINT-QUENTIN-EN-YVELINES