Keccak

An update

Guido Bertoni[1] Joan Daemen[1]
Michaël Peeters[2] Gilles Van Assche[1]

[1]STMicroelectronics

[2]NXP Semiconductors

Second SHA-3 candidate conference, Santa Barbara, CA
August 23-24, 2010

Outline

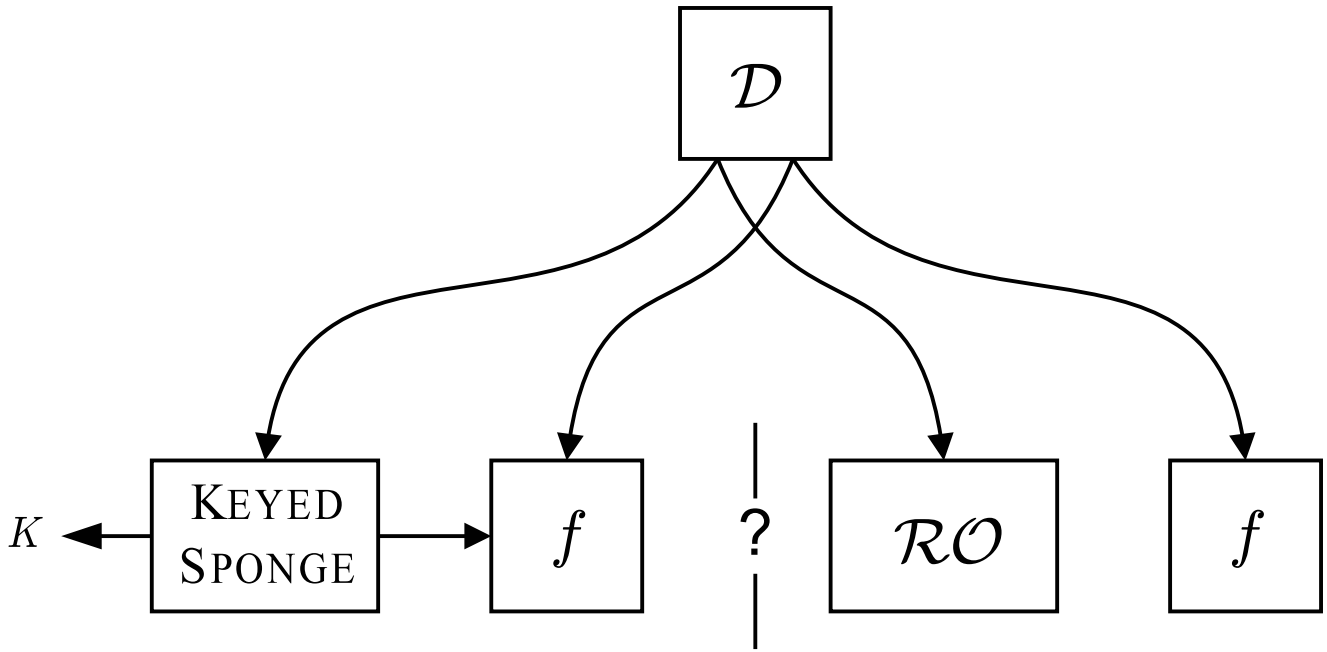
The sponge construction

- The sponge construction with capacity c
  - No generic attacks below $2^{c/2}$
  - Covers *all* attacks, unless easy on a random oracle

Collision	$\min(2^{c/2}, 2^{n/2})$
Preimage	$\min(2^{c}, 2^{n})$ (new)
Second preimage	$\min(2^{c/2}, 2^{n})$

- New results for:
  - Preimage: $\min(2^{c}, 2^{n})$ [Bertoni et al., CHES 2010]
  - Keyed modes (key derivation, authentication/encryption)...

Security of keyed sponge functions



- Security beyond $2^{c/2}$ time (offline) complexity, if data (online) complexity is limited
- Indistinguishable if data $\times$ time $\leq 2^{c-1}$
- See [Bertoni et al., On the security of the keyed sponge construction]

The design strategy

For Keccak: **flat sponge claim**

It must resist any attack with complexity up to $2^{c/2}$

(unless easier on a random oracle)

For Keccak f: the **hermetic sponge strategy**

Design f without structural distinguisher

- A **conservative** approach...

A conservative approach

Design f without structural distinguisher

no distinguisher on KECCAK-f
$\Rightarrow$ no distinguisher on KECCAK
$\Rightarrow$ no attack on KECCAK

- Safety margin on the applicability axis:
  distinguisher on KECCAK-f
  $\not\Rightarrow$ distinguisher on KECCAK
  $\not\Rightarrow$ attack on KECCAK
- Safety margin on the complexity axis:
  distinguisher on KECCAK-f with $\leq 2^{1600}$ complexity
  $\not\Rightarrow$ distinguisher on KECCAK-f with $\leq 2^{800}$ complexity
  $\not\Rightarrow$ distinguisher on KECCAK$[r > 0, c < 1600]$

Best cryptanalysis results

- Attacks on KECCAK
  - 3 rounds: preimage [Morawiecki, Srebrny]
  - 4 rounds: key recovery [Lathrop]
- Distinguishers on KECCAK-f[1600]
  - 3 rounds: CICO problem [Aumasson, Khovratovich]
  - 4 rounds: cube testers [Aumasson, Khovratovich]
  - 16 rounds: (2^{1024}) zero-sum [Aumasson, Meier]
  - 18 rounds: (2^{1370}) zero-sum [Boura, Canteaut]
  - 20 rounds: (2^{1586}) zero-sum [Boura, Canteaut]
  - 24 rounds: (2^{15XX}) zero-sum
    [Boura, Canteaut, De Cannière, Rump Crypto 2010]
- Three cryptanalysis prizes awarded!
- And our analysis [KECCAK main document]

Zero-sum distinguishers on Keccak-f

- 24 rounds: complexity 2^{15XX}
- Putting in perspective
  - Block ciphers have many (smaller) zero-sum sets by nature

$$\sum_{x \in \mathbb{Z}_2^m} E_k(x) = \sum_{x \in \mathbb{Z}_2^m} x = 0, \forall k$$

  - Only $\times 2$ faster than generic (zero-sum set only)
- But we consider them valid structural distinguishers!
  - Do they compromise the *hermetic sponge strategy*?

Zero-sums and the hermetic sponge strategy

- Applicability axis: could not be extended to KECCAK itself
  - No control over inner (c-bit) parts, both at input and output
  - Do not help solve problems relevant in attacks (e.g., CICO)
- Complexity axis: $2^{15^{XX}}$ versus 2^{800} versus $2^{\text{reasonable}}$
  - Above 2^{800}: no impact on the indifferentiability bound
    (and no need for complex proofs)
- Safe to stick to 24 rounds

What is currently the safety margin of Keccak?

- Currently best attack on Keccak: 4 rounds
- Sufficient #rounds for security claim on Keccak: 13 rounds
  Estimation from [Keccak main document]
- Keccak has 24 rounds

Software performance

- Good performance on 64-bit CPUs (e.g., Intel Core 2 Duo)

October 2008	18 rounds	12.5 cycles/byte
Now	24 rounds	12.6 cycles/byte

[eBASH]

  - Thanks to better C code and GCC $\geq$ 4.4
- Decent performance on small 32-bit processors
  - Thanks to bit interleaving
    - Implementation using 32-bit words only
    - 32-bit rotations, no carry to propagate
  - Example on ARM Cortex M3 (for Keccak[r=1088, c=512])

Code in sphlib 2.1	268 cycles/byte
Our round-2 code	153 cycles/byte

Other performance aspects

- Excellent suitability on hardware [Tillich et al.] and [CHES 2010]
  - With speed/area trade-offs
  - Low energy consumption per bit
- Secure implementations much cheaper than other designs
  - Against timing attacks
  - Against power analysis attacks
  - See [Bertoni et al., Note on side-channel attacks...]
- Flat memory usage ($c + r$ bits)
  - No memory for feed-forwards
  - No memory for message queue
  - Compact storage of intermediate state if needed ($c + \epsilon$ bits)
    [Bertoni et al., Duplexing the sponge...]

Keccak has strong (and sometimes unique) features

- Design and security
  - Thick safety margin
  - Matryoshka principle: cryptanalysis from small to large
  - Provable security against generic attacks
- Flexibility inherent in the sponge construction
  - Simple security claim, disentangled from output length
  - Arbitrary output length (for, e.g., MGF, stream cipher)
  - Single permutation for all output lengths
  - Performance-security (rate-capacity) trade-offs
  - No output transformation (e.g., efficient duplexing)
- Implementation
  - Good software performance
  - Excellent suitability on hardware with speed/area trade-offs
  - Secure implementations much cheaper than other designs

Our references

- *Sponge functions*, comment to NIST and Ecrypt Hash Workshop 2007
- *On the indifferentiability of the sponge construction*, Eurocrypt 2008
- KECCAK *sponge function family main document* (version 2.1 or later)
- *The road from* PANAMA *to* KECCAK *via* RADIOGATÚN, Dagstuhl 2009
- *Note on side-channel attacks and their counterm...*, NIST hash forum 2009
- *Note on zero-sum distinguishers of* KECCAK-*f*, NIST hash forum 2010
- *Note on* KECCAK *parameters and usage*, NIST hash forum 2010
- KECCAKTOOLS, Ecrypt Workshop on Tools for Cryptanalysis 2010
- *Sponge-based pseudo-random number generators*, CHES 2010
- *Duplexing the sponge: authenticated encryption and other applications*
- *On the security of the keyed sponge construction*
- *Building power analysis resistant implementations of* KECCAK

http://keccak.noekeon.org/