KECCAK

A family of sponge functions

Guido BERTONI[1] Joan DAEMEN[1] Michaël PEETERS[2]
Gilles VAN ASSCHE[1]

[1]STMicroelectronics

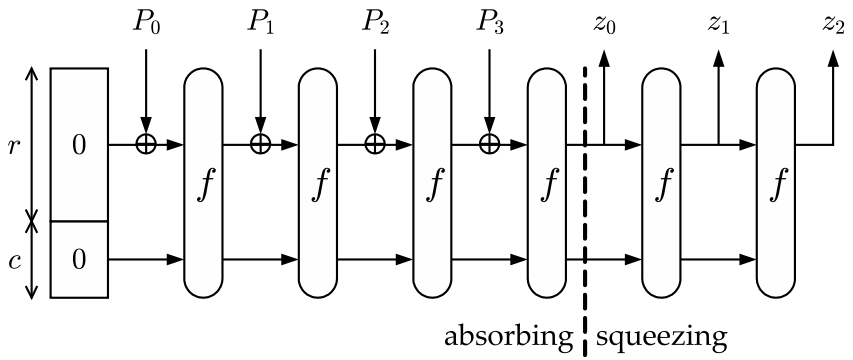
[2]NXP Semiconductors

First SHA-3 candidate conference, Leuven, Belgium
February 26, 2009

Outline

Outline

The sponge construction



- Variable-length input, indefinite-length output
- Secure against generic attacks with $< 2^{c/2}$ calls to f
  - Indifferentiability proof assumes f is random permutation
  - Attacks exploiting specific properties of f are not covered
- Provable security against generic attacks

KECCAK

- KECCAK follows the **hermetic sponge strategy**
  - Instantiation of a sponge function
  - Permutation f shall be designed such that it has no exploitable properties
- KECCAK uses a permutation KECCAK-$f[r + c]$
- Actually, seven permutations
  - $r + c \in \{25, 50, 100, 200, 400, 800, 1600\}$
  - Primary choice: KECCAK-$f[1600]$

Distinguishing features of the sponge construction

- Security-speed trade-offs using the same permutation
  - E.g., using KECCAK-f[1600], $r + c = 1600$:
  - $r = 1024$ and $c = 576$ for $2^{c/2} = 2^{288}$ security, faster
  - $r = 512$ and $c = 1088$ for $2^{c/2} = 2^{544}$ security, slower
- Usage of KECCAK: more than just hashing
  - Variable-length output
    - Stream cipher
    - Mask generating function
  - *Blank page* input, structure determined by usage scenario
    - Randomized/diversified hash function
    - MAC function
  - As component in tree hashing mode
  - *Slow* hash function
  - ...

Outline

The KECCAK-*f* permutation

- KECCAK-*f* is an iterated permutation
- Like a block cipher
  - Sequence of identical rounds
  - Round consists of sequence of simple step mappings
- . . . but not quite
  - No key schedule
  - Round constants instead of round keys
  - Inverse permutation need not be efficient

Structural distinguishers in KECCAK-*f*

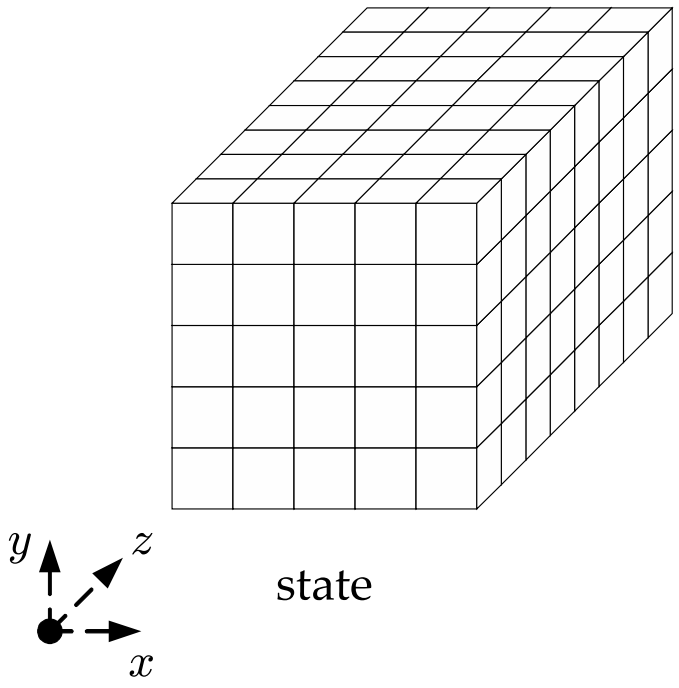
- Presence of large input-output correlations
- Ability to control propagation of differences
  - Differential/linear trail analysis
  - Strong bounds for 25-bit and 50-bit versions
  - This shaped the diffusion layer
- Algebraic Normal Form representation
  - Distribution of number of terms of certain degrees
  - Imposes lower bound to number of rounds
  - Relevant for, e.g., cube attacks and testers
- Symmetry properties
  - Both in space (state) as in time (rounds)
  - This shaped the round constants
- Ability of solving certain problems algebraically

Distinguishing features of KECCAK-*f*

- Use of a permutation: block cipher without key schedule
- Range of 7 permutations, from small to large
  - *Matryoshka* structure
  - Also possible: KECCAK-*f*[800] (32 bits)
  - Toy permutations: KECCAK-*f*[25] or KECCAK-*f*[50]
- Symmetry and parallelism
- Algebraic simplicity: writing equations is straightforward
  - KECCAKTOOLS available
- Overall good performance...

Outline

The state: an array of $5 \times 5 \times 2^\ell$ bits

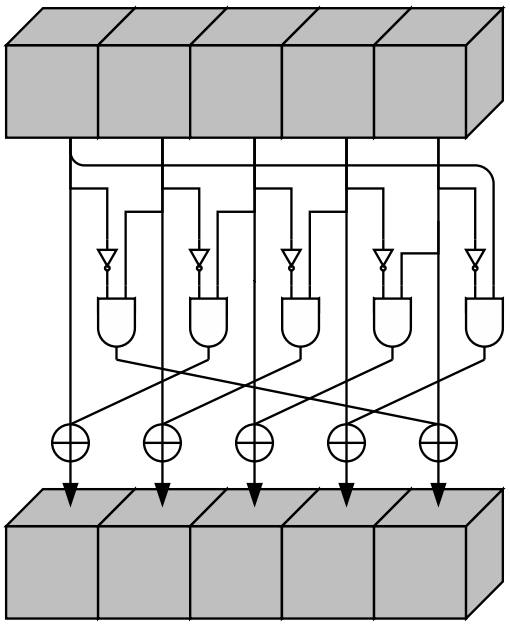


- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ (1, 2, 4, 8, 16, 32 or 64) of them

The rounds of KECCAK-f

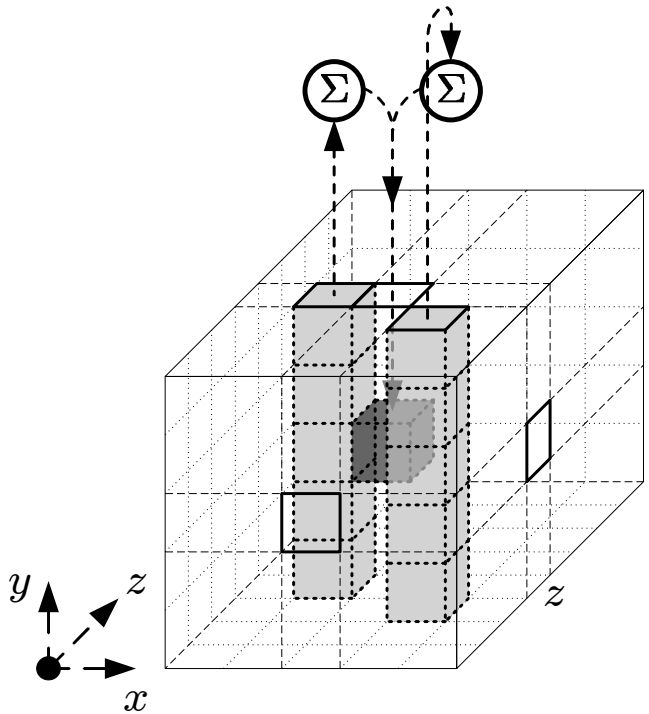
- A round consists of 5 invertible step mappings
  - θ for diffusion
  - ρ for inter-slice dispersion
  - π for disturbing horizontal/vertical alignment
  - χ for non-linearity
  - ι to break symmetry
- Number of rounds: $12 + \ell$
  - KECCAK-f[25] has 12 rounds
  - KECCAK-f[1600] has 18 rounds

χ for non-linearity



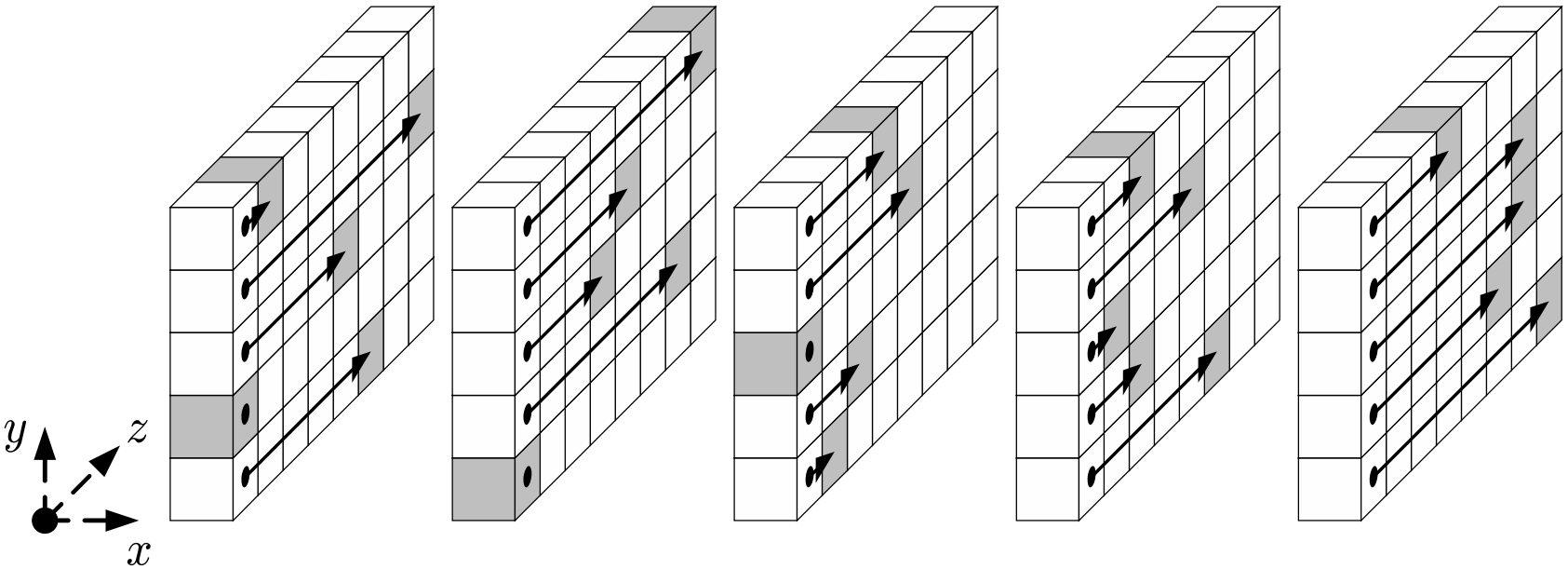
- Simple nonlinear mapping with well-understood properties
- Inherited from RADIOGATÚN and further back . . .

θ for diffusion



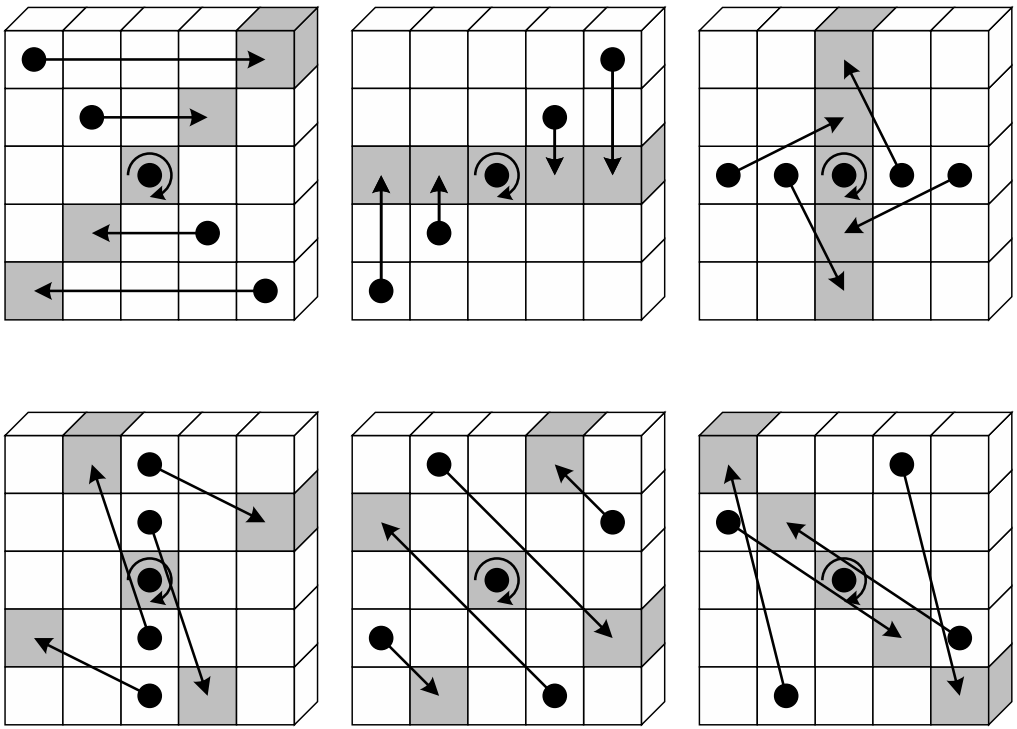
- Each input bit affects 11 output bits
- 50 bitwise XORs and 5 rotations

ρ for inter-slice dispersion



- Moves bits of a slice to 25 different slices
- 24 rotations

π for disturbing horizontal/vertical alignment



- Cycle with period 24 around a fixed origin
- Linear mapping of (x, y) coordinates in GF(5)

Outline

Implementation

- Written in C only (and no SIMD instructions [yet])
- KECCAK-f[1600] on a 64-bit CPU:
  - One lane = one CPU word
  - 10 cycles/byte for $r = 1024$
- KECCAK-f[1600] on a 32-bit CPU:
  - Two CPU words per lane
  - Bit interleaving technique: 32-bit rotations
  - 31 cycles/byte for $r = 1024$
- Well suited for hardware: speed/area trade-off
  - High-speed ASIC: 30 Gbit/s, 48 kGate, 526 MHz
  - Low-area ASIC: 53 Mbit/s, 5 kGate, 200 MHz
- Well suited for DPA protection: masking and secret sharing

Outline

Questions?

Thanks for your attention!

Any questions?

More information on
`http://keccak.noekeon.org/`