Generic security of the Keyed Sponge

Joan Daemen[1,2]

based on joint work with
Guido Bertoni[1], Michaël Peeters[1], Gilles Van Assche[1],
Elena Andreeva[3] and Bart Mennink[3]

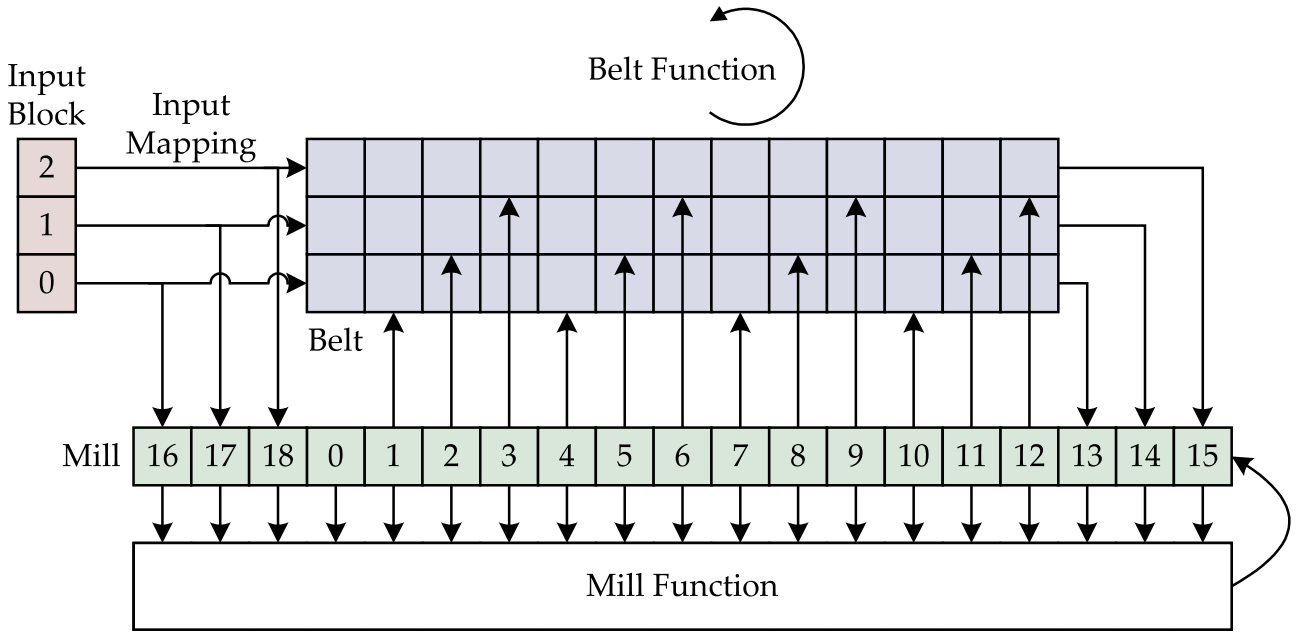
[1]STMicroelectronics [2]Radboud University [3]COSIC KULeuven

ArcticCrypt
Longyearbyen
July 19, 2016

Outline

Outline

RadioGatún [Keccak team, NIST 2nd hash workshop 2006]



- XOF: eXtendable Output Function
- Problem: **expressing security claim**
- Search for random oracle but then with inner collisions

(Early) Sponge at Dagstuhl, January 2007

Screenshot:

- Description:
  - Internal state $S = (S_A, S_G) \in Z_2 \times Z_2^c$ with initial value $S = (0,0)$
  - Absorbing: for each bit p of the input:

    $S = f(S_A + p,\ S_G)$
  - Resting:

    $S = f(S_A + 1,\ S_G)$
  - Squeezing: for each bit z of the output:

    $z = S_A$

    $S = f(S_A + 0,\ S_G)$
- We call c: the *sponge capacity*

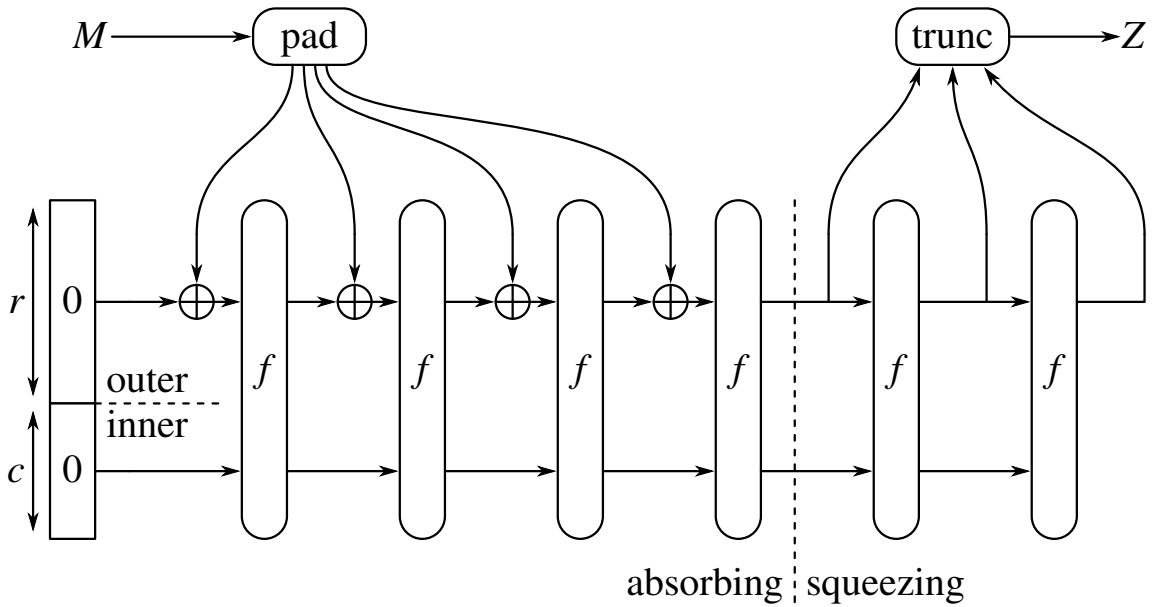
Generic security of Sponge [KT, Ecrypt hash, September 2007]

- Random sponges:
  - T-sponge: f is random transformation
  - P-sponge: f is random permutation
- Theorem: if no inner collisions, output is uniformly random
  - inner collision: different inputs leading to same inner state
  - Probability of inner collision:

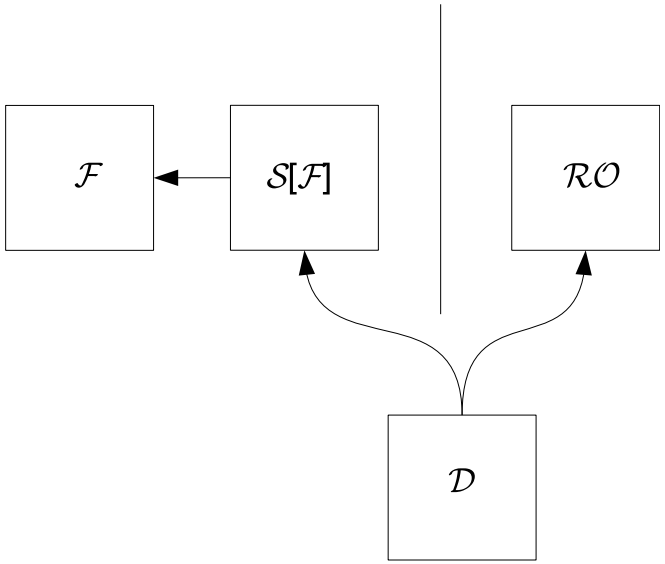
$$2^{-c-1}M^2 \text{ with } M : \# \text{ calls to } f$$

Promoting sponge from reference to usage (2007-2008)

- RadioGatún cryptanalysis (1st & 3rd party): not promising
- NIST SHA-3 deadline approaching ...U-turn
- Sponge with *strong* permutation f: Keccak [KT, SHA-3, 2008]

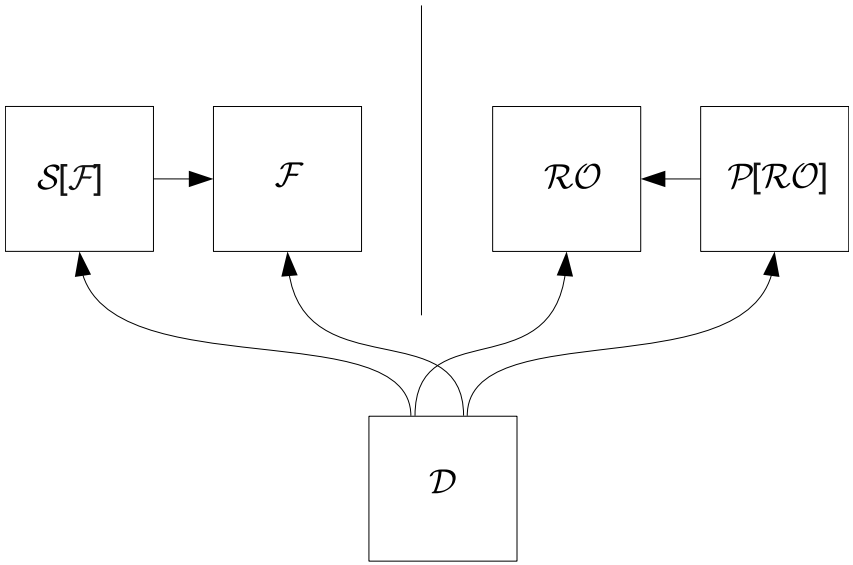


Distinguishing random sponge from random oracle



- Distinguishing advantage: $2^{-c-1}M^2$
- Problem: in real world, adversary has access to f

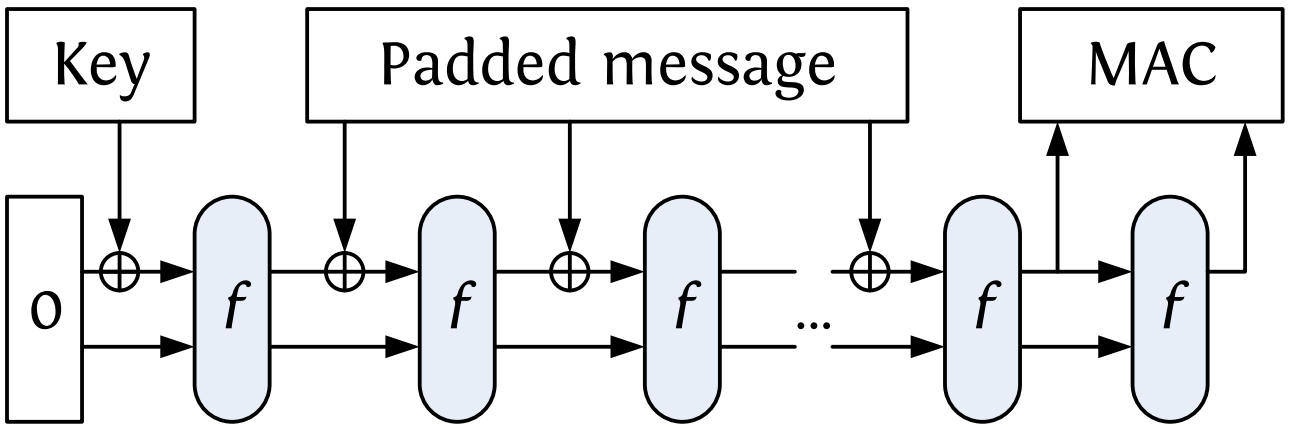
Differentiating random sponge from random oracle



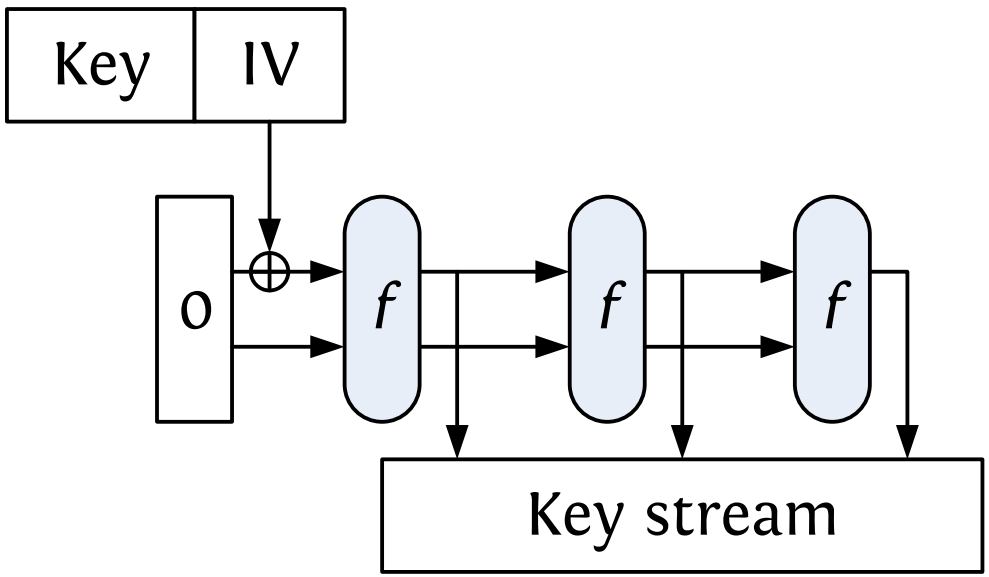
- Indifferentiability framework [Maurer, Renner & Holenstein, 2004]
- Applied to hashing [Coron, Dodis, Malinaud & Puniya, 2005]
- Random oracle augmented with *simulator* for sake of proof
- Differentiating advantage: $2^{-c-1}M^2$ [KT, Eurocrypt 2008]

Outline

Message authentication codes

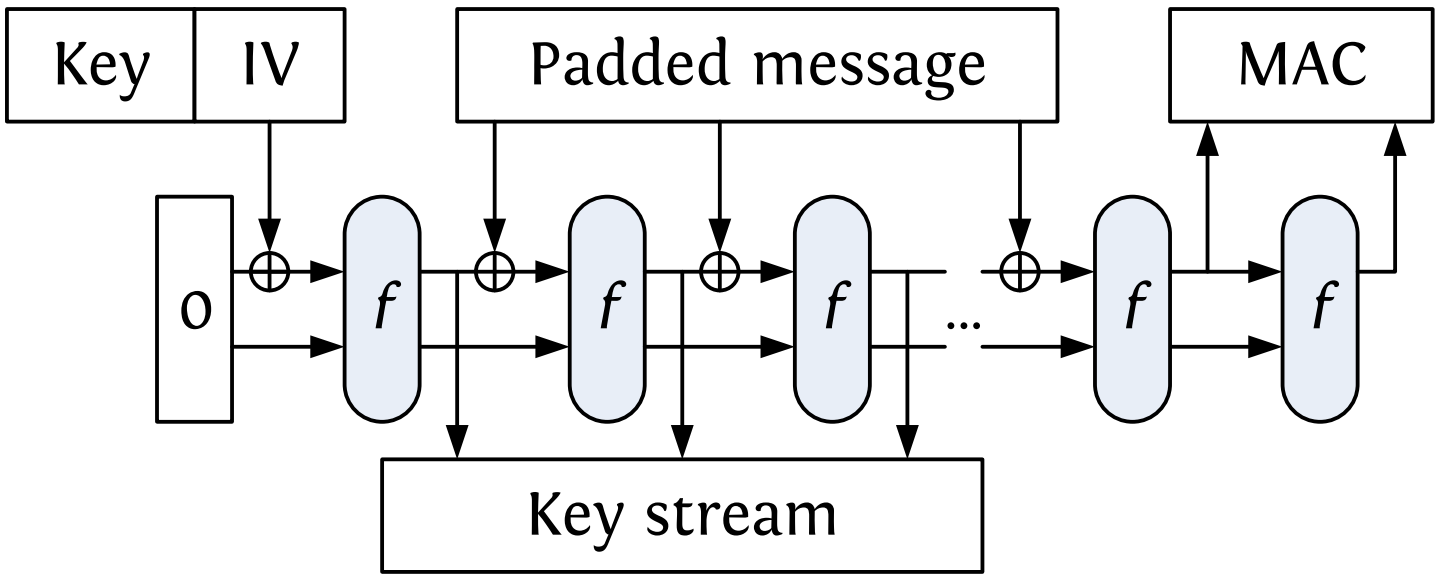


Stream encryption



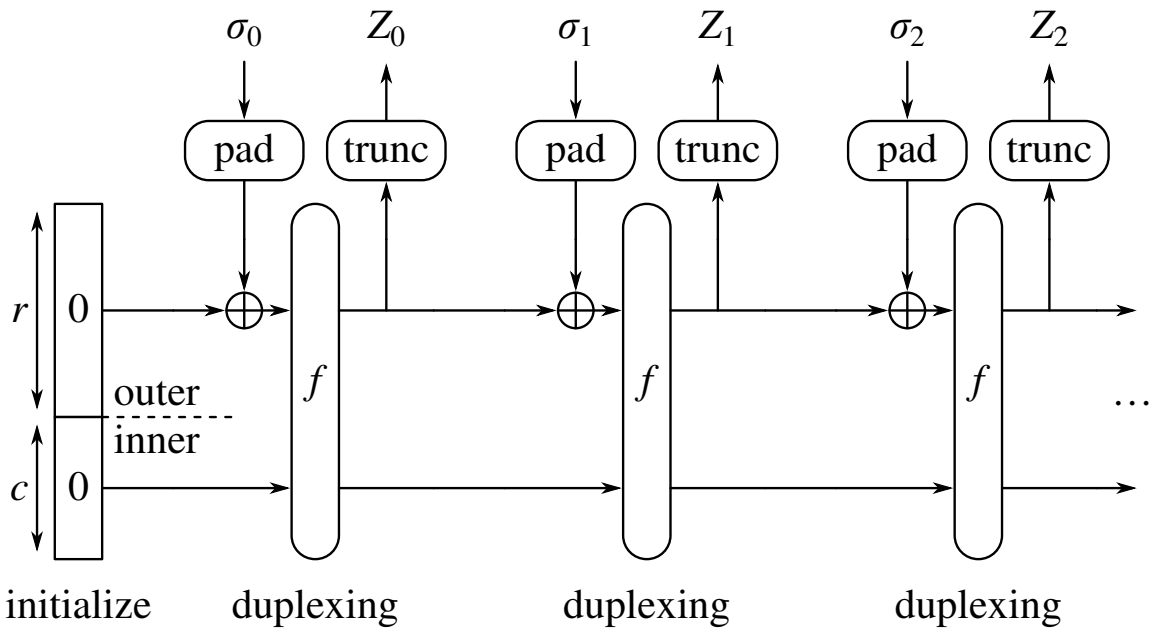
- Long output stream per IV: similar to OFB mode
- Short output stream per IV: similar to counter mode

Authenticated encryption: spongeWrap [KT, SAC 2011]



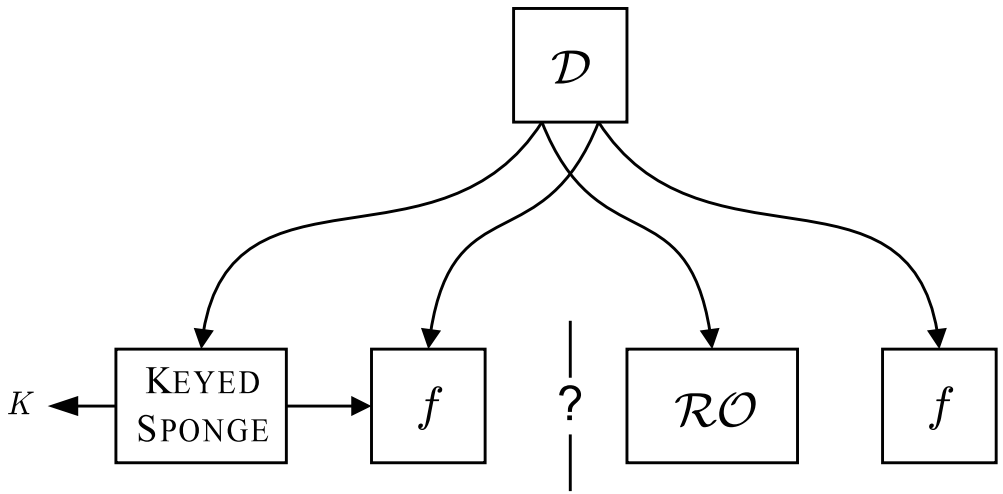
- Adopted by several CAESAR candidates
- But this is no longer sponge

The duplex construction [KT, SAC 2011]



Generic security equivalent to that of sponge

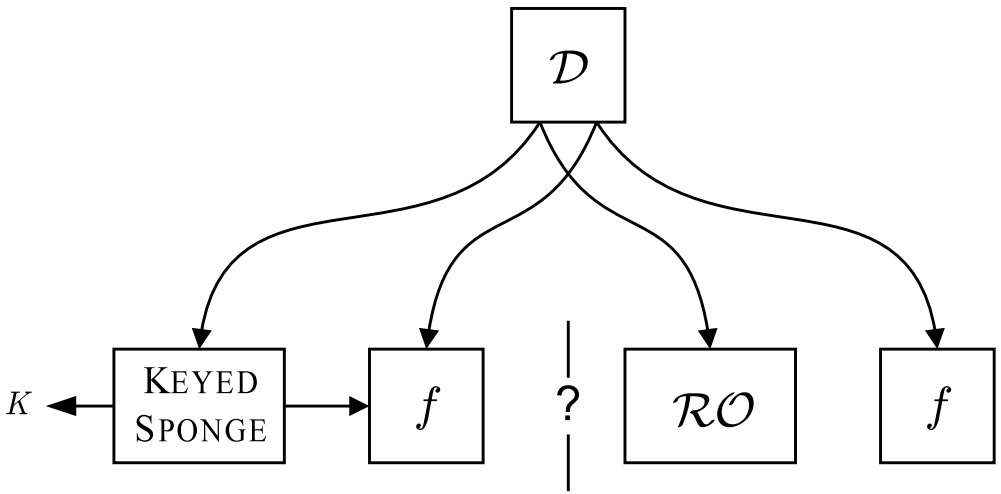
Keyed sponge: distinguishing setting



- Straightforward bound: $2^{-c-1}M^2 + 2^{-k}M$
- Security strength s: expected complexity of succesful attack
  - strength s means attack complexity 2^s
  - bounds can be converted to security strength statements
- Here: $s \geq \min(c/2, k)$
  - e.g., $s = 128$ requires $c = 256$ and $k = 128$
  - $c/2$: birthday bound

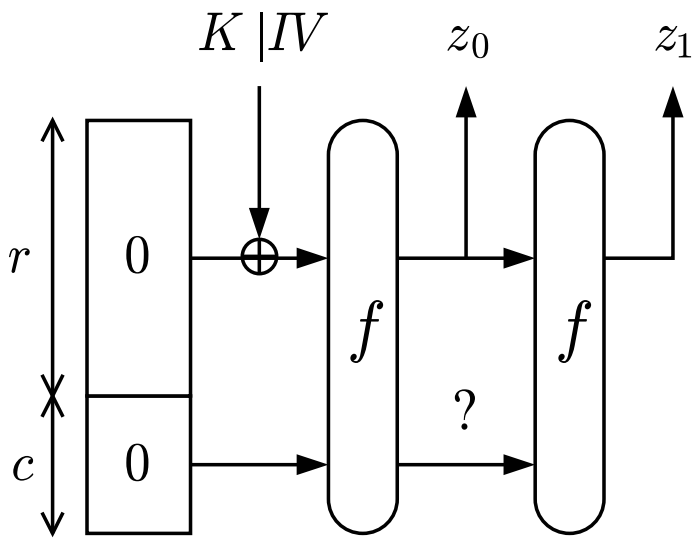
Outline

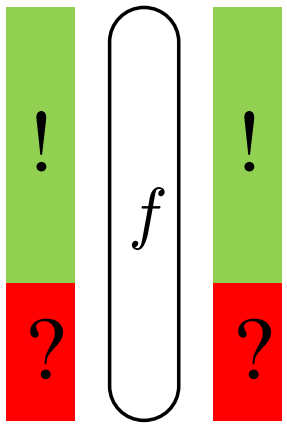
More fine-grained attack complexity



- Splitting attack complexity:
  - queries to construction: data complexity M
  - queries to f or f^{-1}: computational complexity N
- Our ambition around 2010: $2^{-c-1}M^2 + 2^{-c}NM + 2^{-k}N$
- If we limit data complexity $M \leq 2^a \lll 2^{c/2}$:
  - $s \geq \min(c - a, k)$
  - e.g., $s = 128$ and $a = 64$ require $c = 192$ and $k = 128$

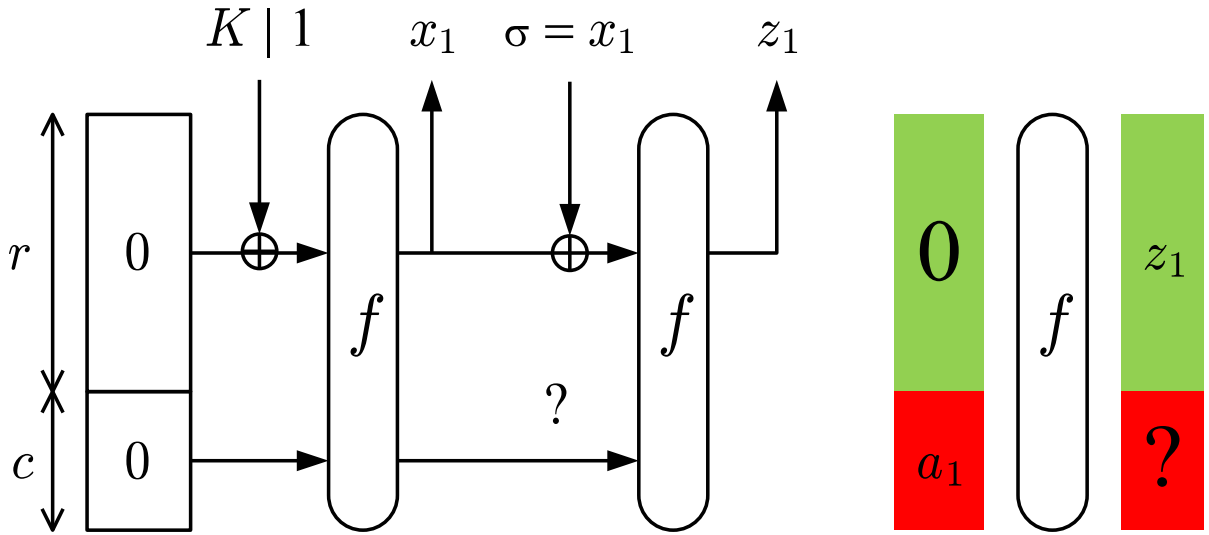
Intuition behind $2^{-c}NM$





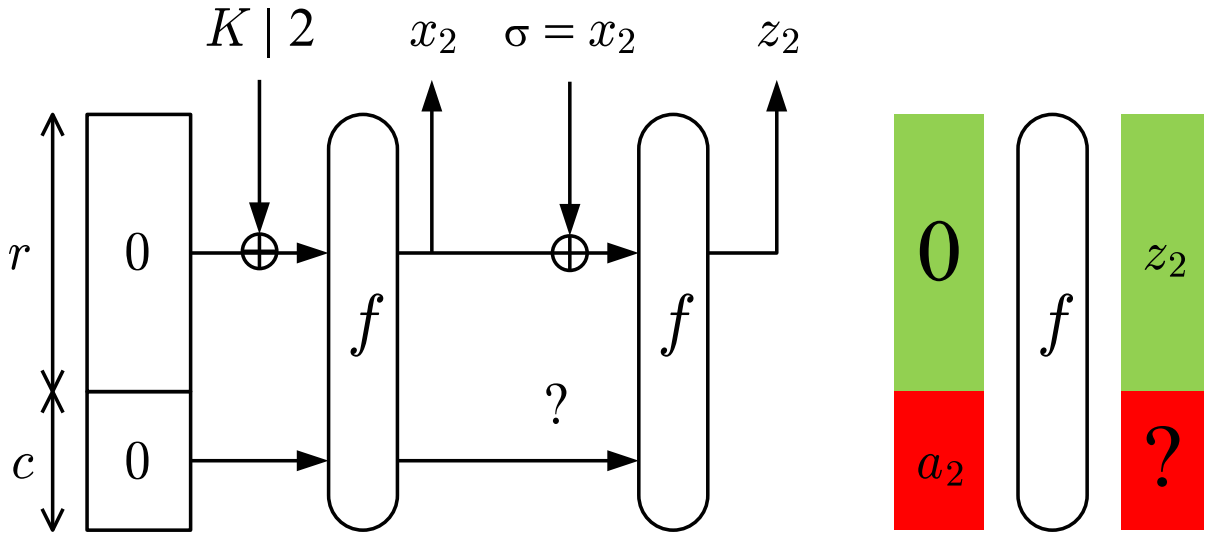
- success probability per guess: 2^{-c}

Intuition behind $2^{-c}NM$



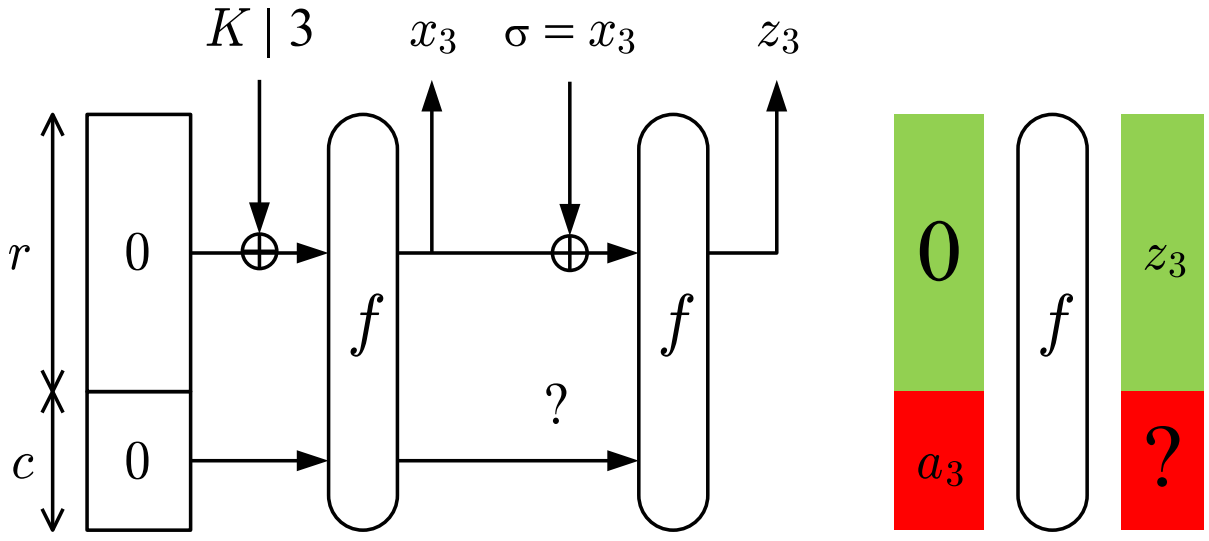
- $\mu \leq M$ instances with same partial r-bit input
- success probability per guess: $\mu 2^{-c}$

Intuition behind $2^{-c}NM$



- $\mu \leq M$ instances with same partial r-bit input
- success probability per guess: $\mu 2^{-c}$

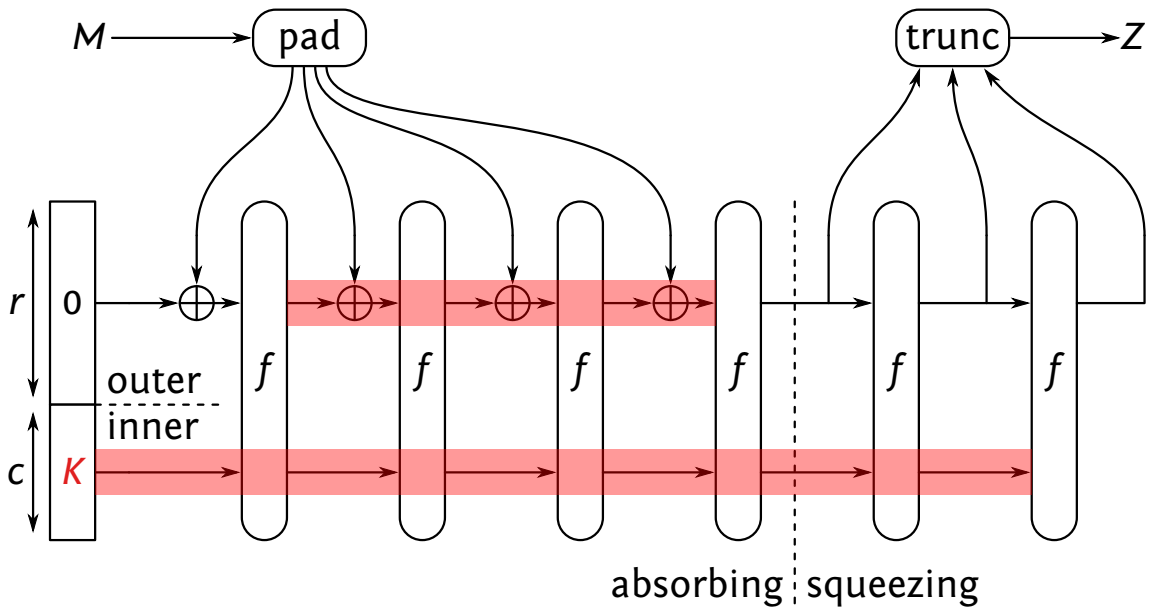
Intuition behind $2^{-c}NM$



- $\mu \leq M$ instances with same partial r-bit input
- success probability per guess: $\mu 2^{-c}$

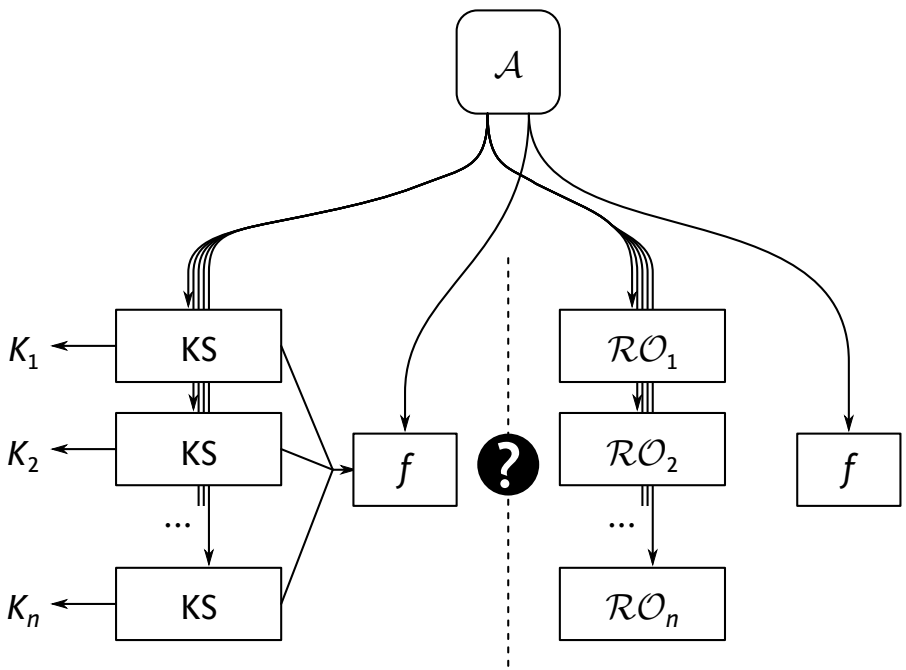
An initial attempt [KT, SKEW 2011]

- bound: $2^{-c-1}M^2 + 2^{-c+1}NM + 2^{-k}N$
- Problems and limitations
  - bound did not cover multi-target (key) attacks
  - proof did not convince reviewers
  - new variant (a.o. in CAESAR): inner-keyed sponge:

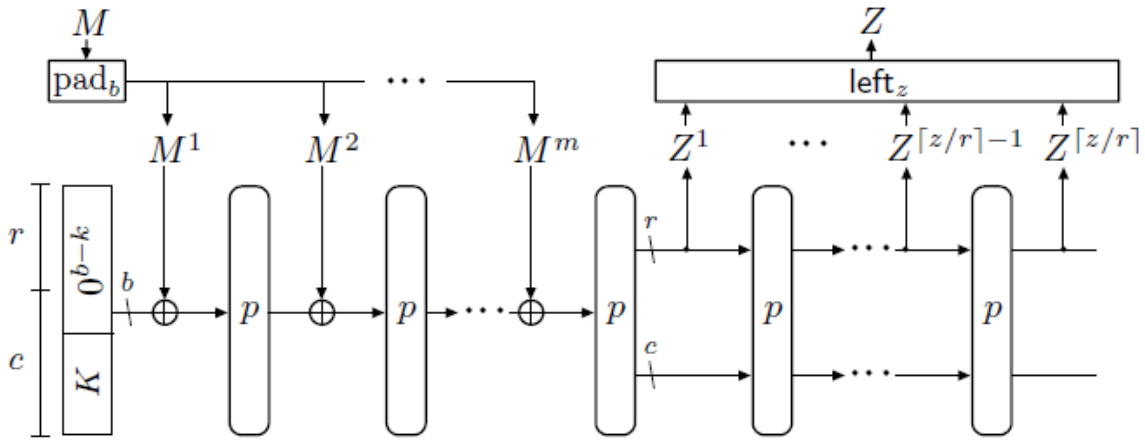


[Andreeva, Daemen, Mennink, Van Assche, FSE 2015]

- Inner/outer-keyed, multi-target (n), multiplicity μ
- Modular proof using Patarin's H-coefficient technique
- Bound: $2^{-c-1}M^2 + 2^{-c+1}\mu N + 2^{-k}nN + \ldots$

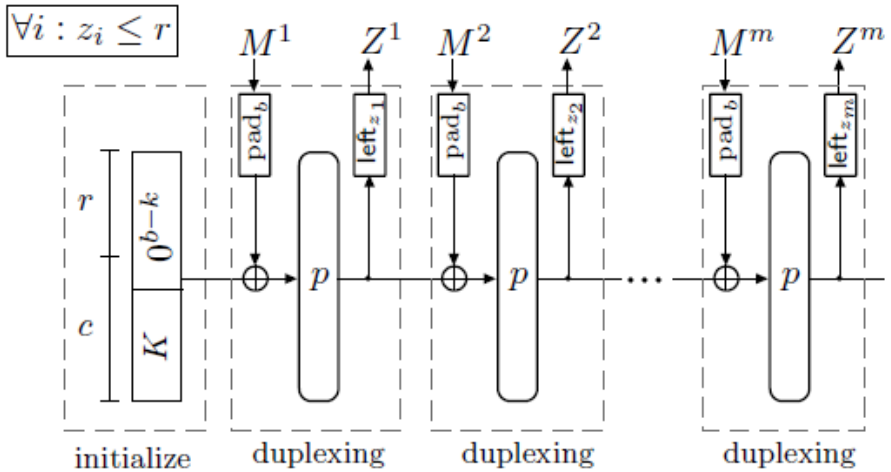


Full-state absorbing! [Mennink, Reyhanitabar and Vizár, Asiacrypt 2015]



- Absorbing on full permutation width does not degrade bounds
- We decided to use that insight in KEYAK v2
- But proven bounds had some limitations and problems:
  - term $2^{-k}\mu N$ rather than $2^{-c}\mu N$
  - no multi-key security
  - multiplicity μ only known a posteriori

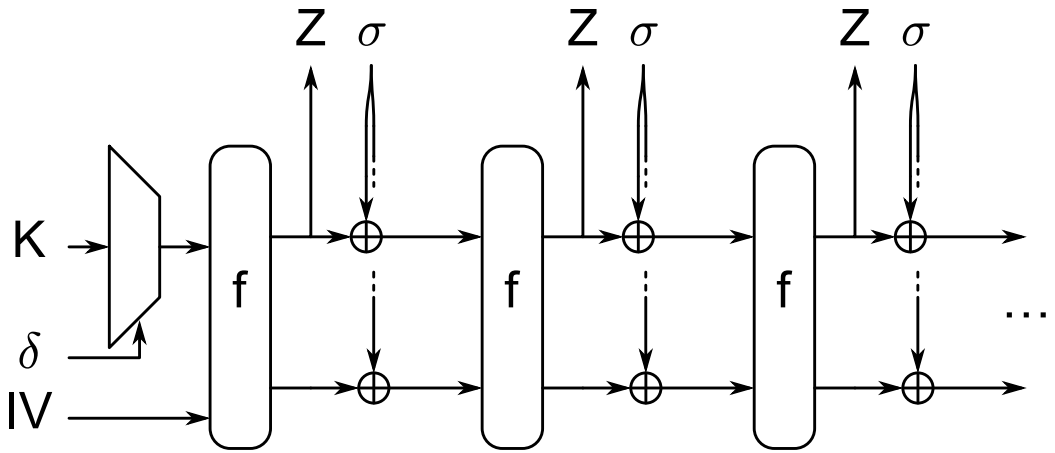
Full-state absorbing! [Mennink, Reyhanitabar and Vizár, Asiacrypt 2015]



- Absorbing on full permutation width does not degrade bounds
- We decided to use that insight in KEYAK v2
- But proven bounds had some limitations and problems:
  - term $2^{-k}\mu N$ rather than $2^{-c}\mu N$
  - no multi-key security
  - multiplicity μ only known a posteriori

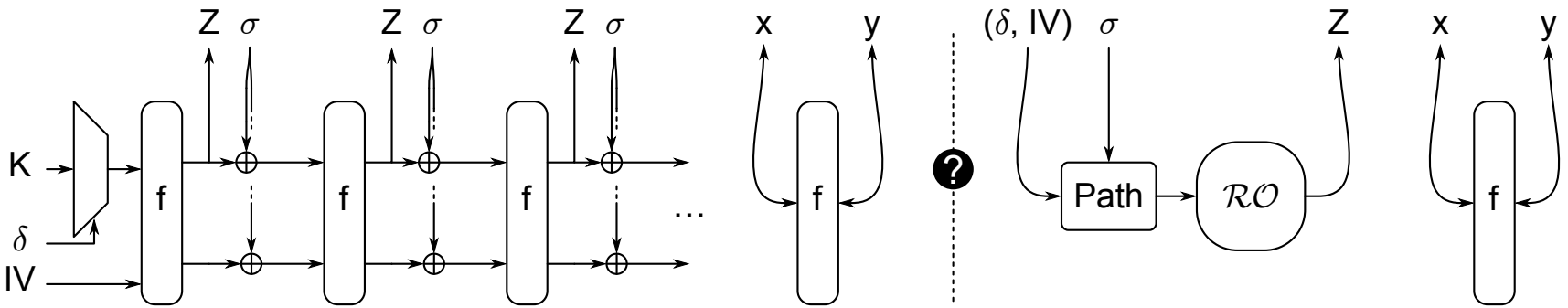
Outline

The new core: (full-state) keyed duplex



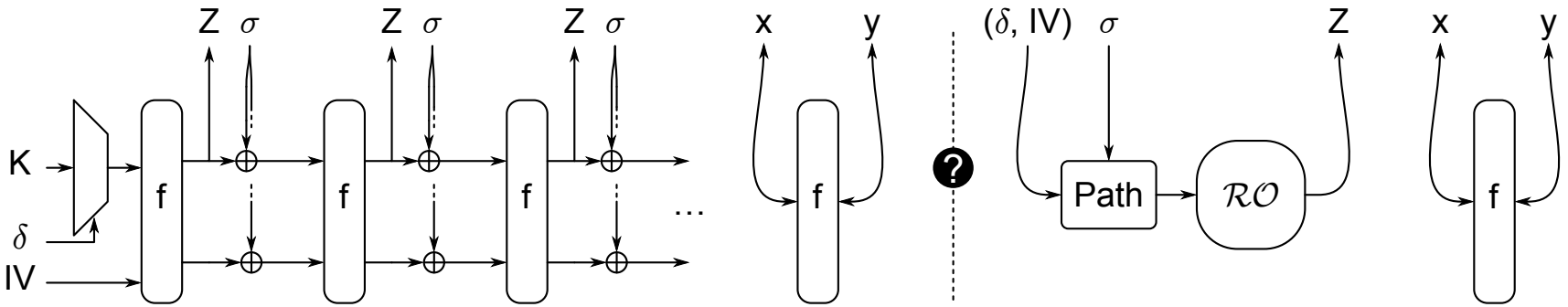
- Full-state absorbing, no padding: $|\sigma| = b$
- Initial state: concatenation of key k and IV
- Multi-key: k selected from an array $\mathbf{K}$ with index δ
- Re-phased: f, Z, σ instead of σ, f, Z
- $\approx$ all keyed sponge functions are modes of this

Generic security of keyed duplex: the setup



- Ideal function: Ideal eXtendable Input Function (IXIF)
  - $\mathcal{RO}$-based object with duplex interface
  - Independent outputs Z for different paths
- Further refine adversary's capability
  - L: # queries to keyed duplex/$\mathcal{RO}$ with repeated path
  - q_{IV} : $\max_{IV}$ # init queries with different keys

Generic security of keyed duplex: the bound



$$2^{-c-1}L^2 + 2^{-c}(L + 2\nu)N + 2^{-k}q_{\mathrm{IV}}N + \ldots$$

with ν: chosen such that probability of ν-wise multi-collision in set of M r-bit values is negligible

Application: counter-like stream cipher

- Only init calls, each taking Z as keystream block
- IV is nonce, so $L = 0$
- Assume $M \lll 2^{r/2}$: $\nu = 1$

Bound:

$$2^{-c}(2\nu)N + 2^{-k}q_{\mathrm{IV}}N + \ldots$$

Strength:

$$s \geq \min(c - 1, k - \log_2(q_{\mathrm{IV}}))$$

Application: lightweight MAC

- Message padded and fed via IV and σ blocks
- t-bit tag, squeezed in chunks of r bits: $c = b - r$
- adversary chooses IV so $L \approx M = 2^a$
- q_{IV} is total number of keys n

Bound:

$$2^{-c-1}M^2 + 2^{-c+1}MN + 2^{-k}nN + \ldots$$

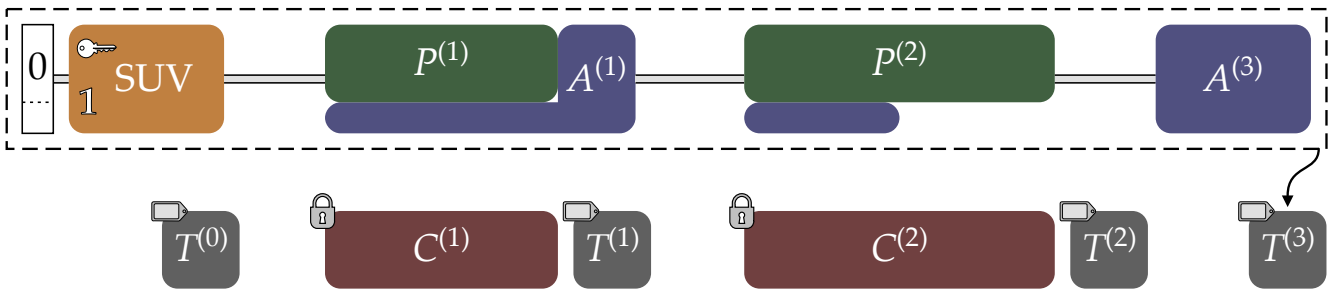
Strength:

$$s \geq \min(b - a - r - 1, k - \log_2(n))$$

Imposes a minimum width of the permutation:

$$b > s + a + r$$

Application: Motorist AE session mode



Used in KEYAK v2 [KT & Ronny Van Keer, 2015]

- Plaintext absorbed in outer part, AD in inner part also
- Used in KEYAK with $c = 256$ and $b = 1600$ or $b = 800$
- Rate 544 or 1344 so we can take $\nu = 1$
- bounds:
  - nonce-respecting: $2^{-c+1}N + 2^{-k}q_{\mathrm{IV}}N + \ldots$
  - nonce-violating: $2^{-c}MN + 2^{-k}q_{\mathrm{IV}}N + \ldots$

Conclusions

- Quite some evolution in keyed sponge
- New results (*in submission*)
  - appropriate keyed-sponge primitive: (full-state) keyed duplex
  - flexible bound covering many use cases
  - makes life easier for sponge mode designer

Thanks for your attention!