Duplexing the sponge: single-pass authenticated encryption and other applications

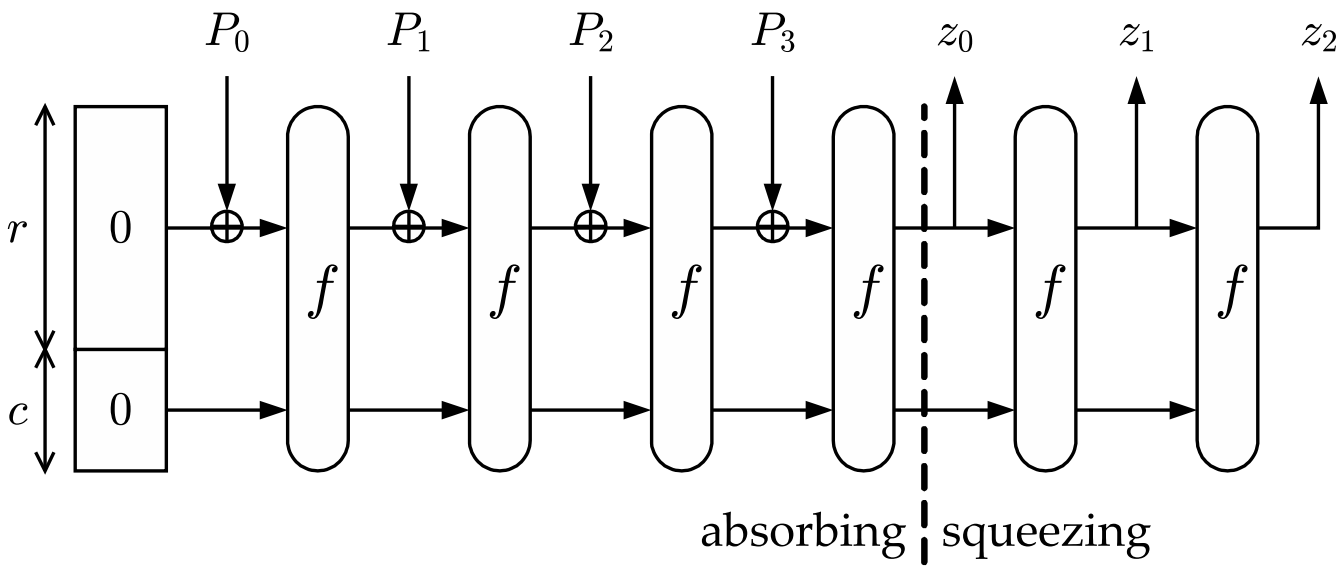
Guido Bertoni[1] Joan Daemen[1]
Michaël Peeters[2] Gilles Van Assche[1]

[1]STMicroelectronics [2]NXP Semiconductors

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

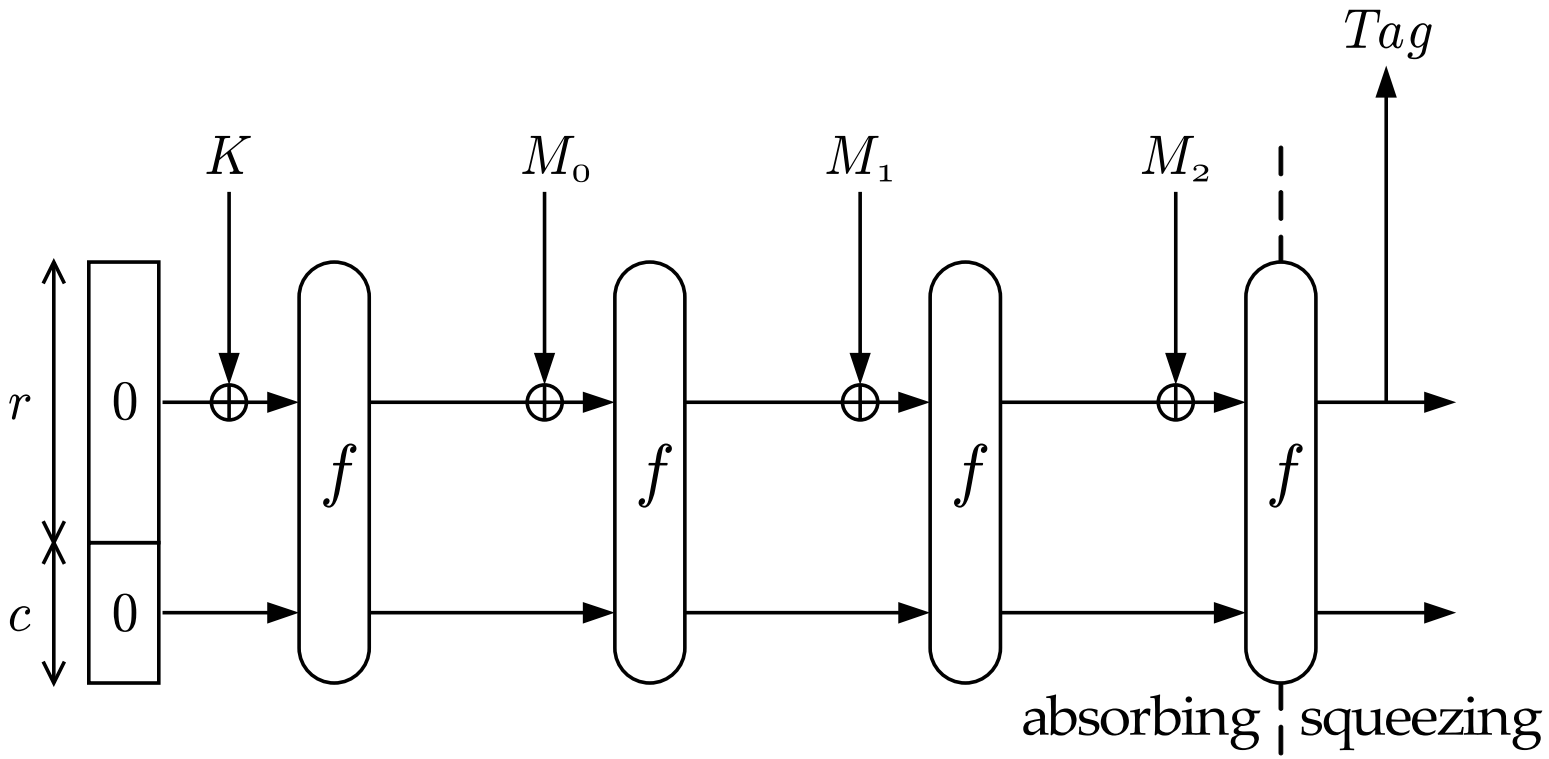
Outline

The sponge construction

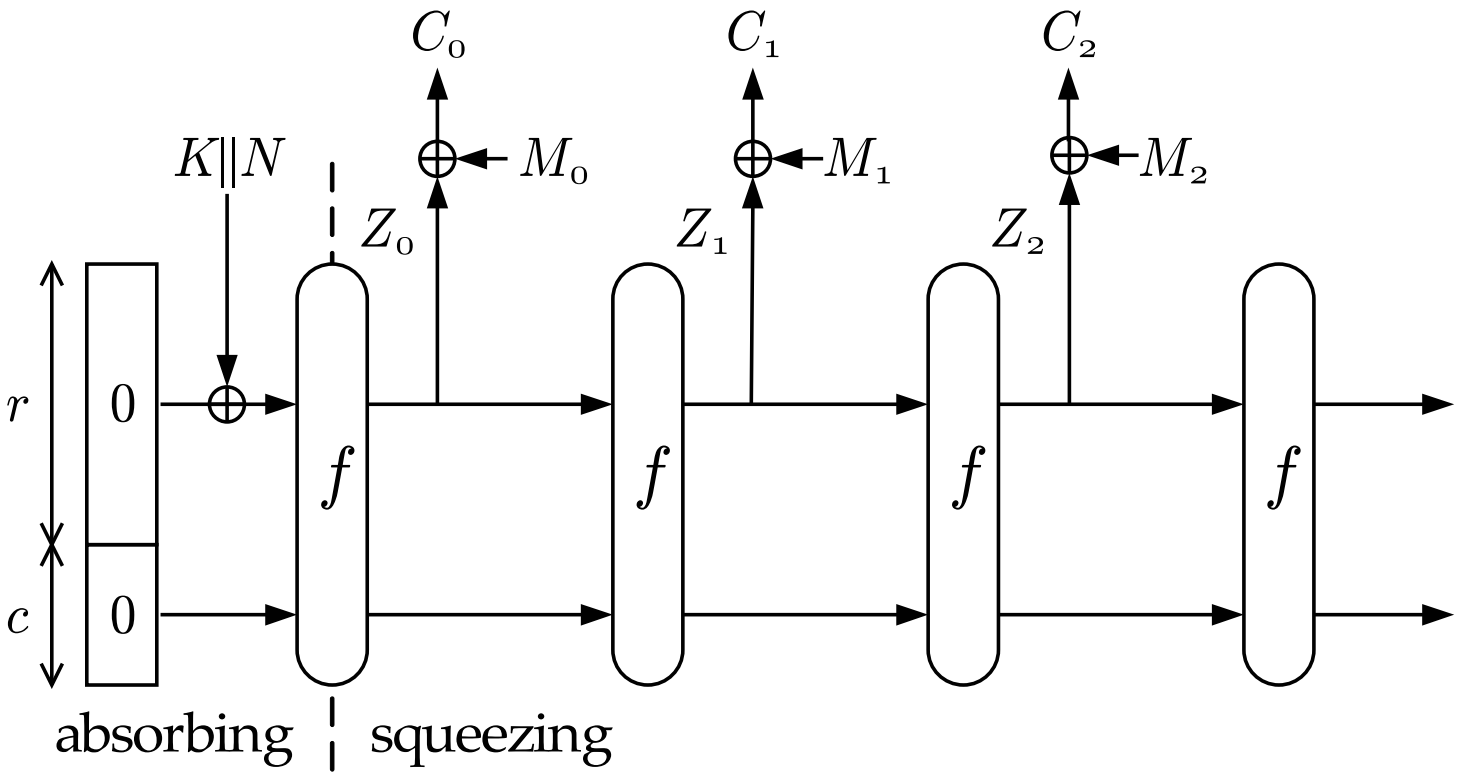


Provably secure if permutation f has no exploitable properties

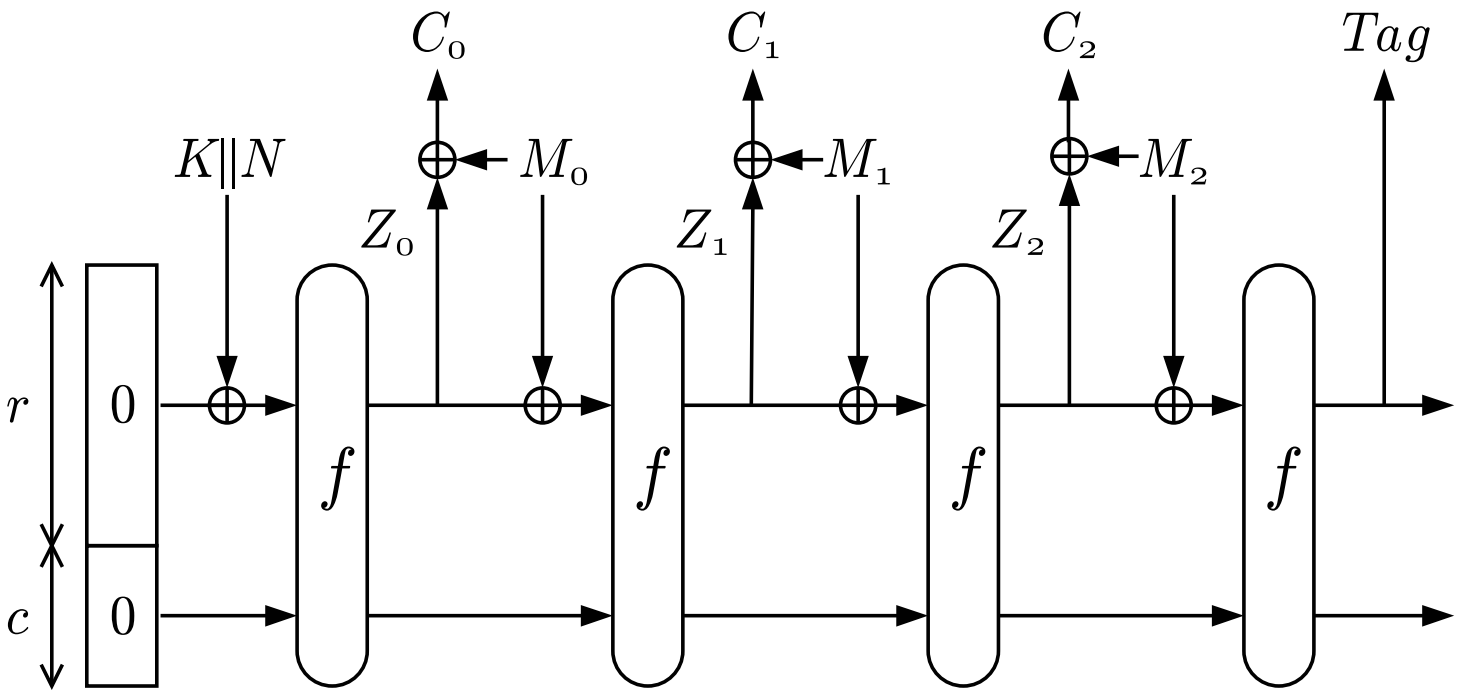
MAC generation with a sponge



Encryption with a sponge

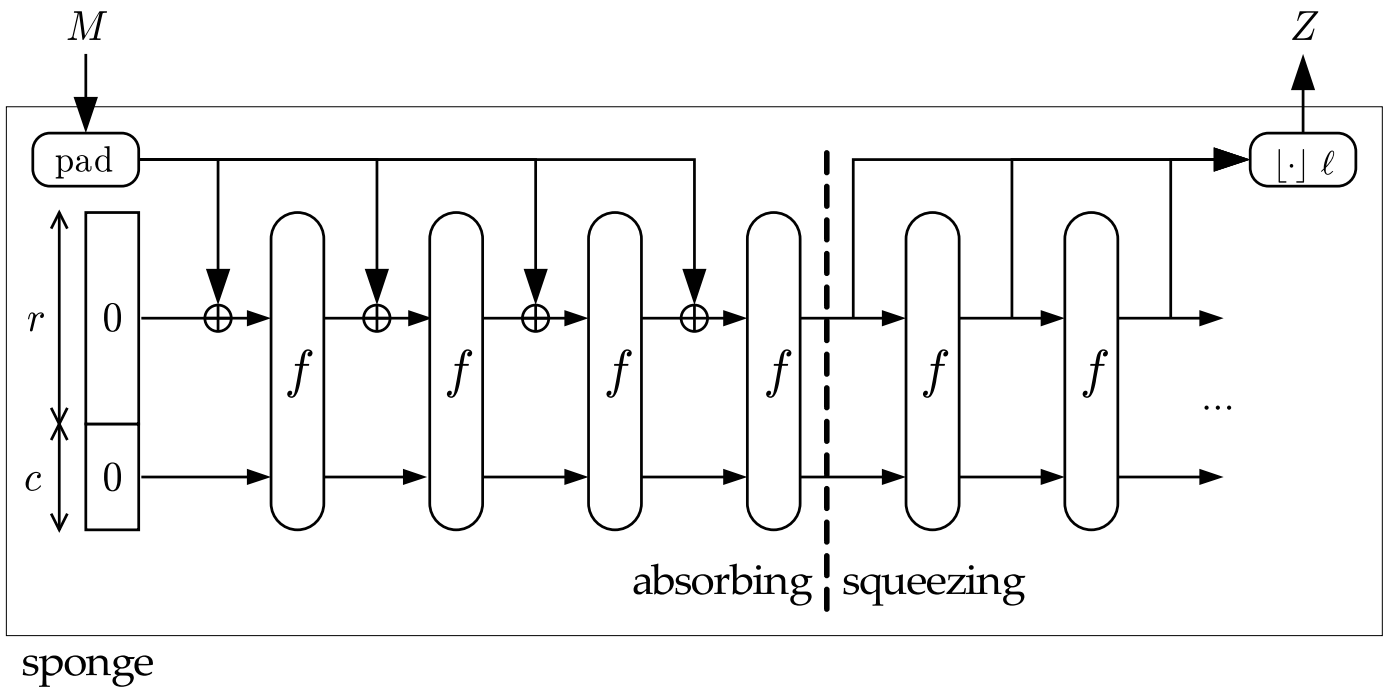


Both encryption and MAC?



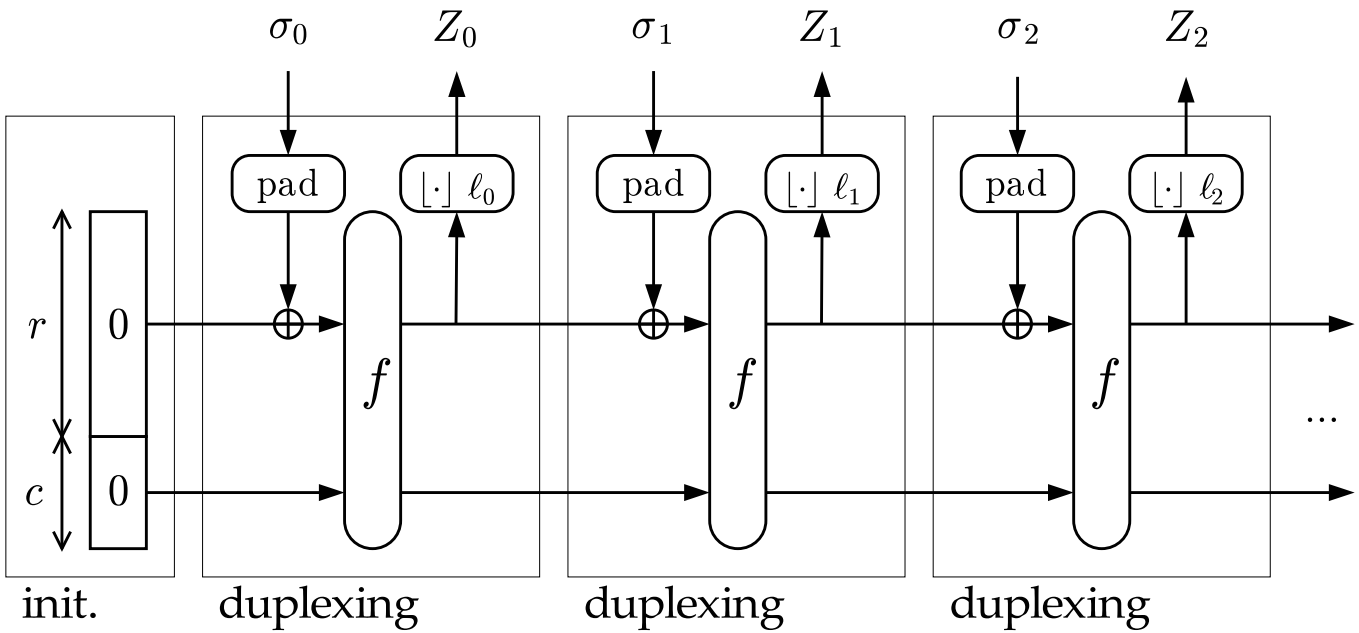
Absorbing and squeezing

The sponge construction, formally defined



- Definition: SPONGE$[f, \text{pad}, r]$
- Requesting ℓ-bit output $Z = \text{sponge}(M, \ell)$

The duplex construction



- Object: $D = \textsc{duplex}[f, \text{pad}, r]$
- Requesting ℓ-bit output $Z = D.\text{duplexing}(\sigma, \ell)$
  - input σ and output Z limited in length
  - Z depends on all previous inputs

Properties of duplex construction

- New type of *cryptographic object*
  - Input can be provided in each call
  - Output can be requested for each call
  - Memory: output to a call depends on all previous inputs
- Almost as efficient as the sponge construction itself
- Opens up new applications …

Security of the duplex construction

Duplexing-sponge lemma

Every output block of a duplex object DUPLEX$[f, \text{pad}, r]$ is a valid output of SPONGE$[f, \text{pad}, r]$

Proof is trivial

Corollary

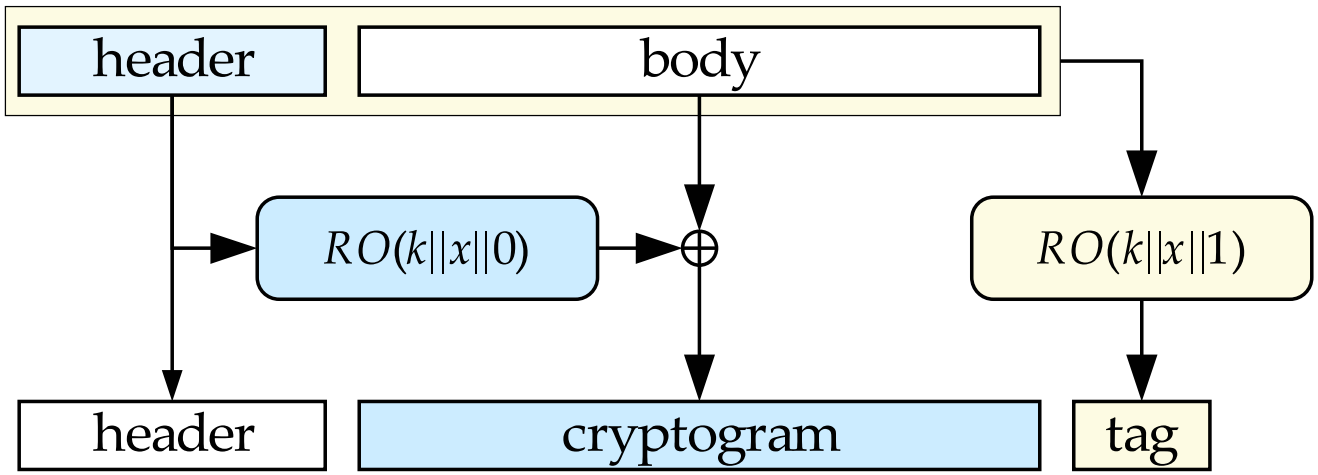
The security of DUPLEX$[f, \text{pad}, r]$ is equivalent to that of SPONGE$[f, \text{pad}, r]$

Authenticated encryption

- Functionality:
  - Tag computation over data header and data body
  - Encryption of body into cryptogram, *diversified by* header
- Wrapping:
  - Input: key, data header and body
  - Output: tag and cryptogram
- Unwrapping
  - Input: key, data header and cryptogram, tag
  - Output: data body or error message if tag is invalid
- Security requirements
  - Key recovery infeasibility
  - Tag forgery infeasibility
  - Plaintext recovery infeasibility

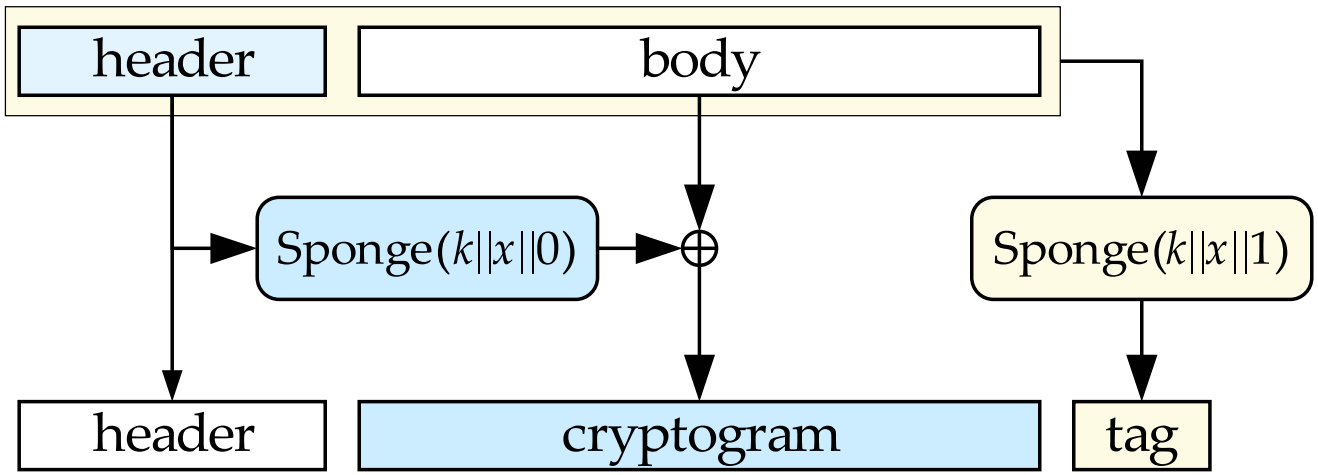
Wrapping with random oracles

Random oracle with *domain separation*

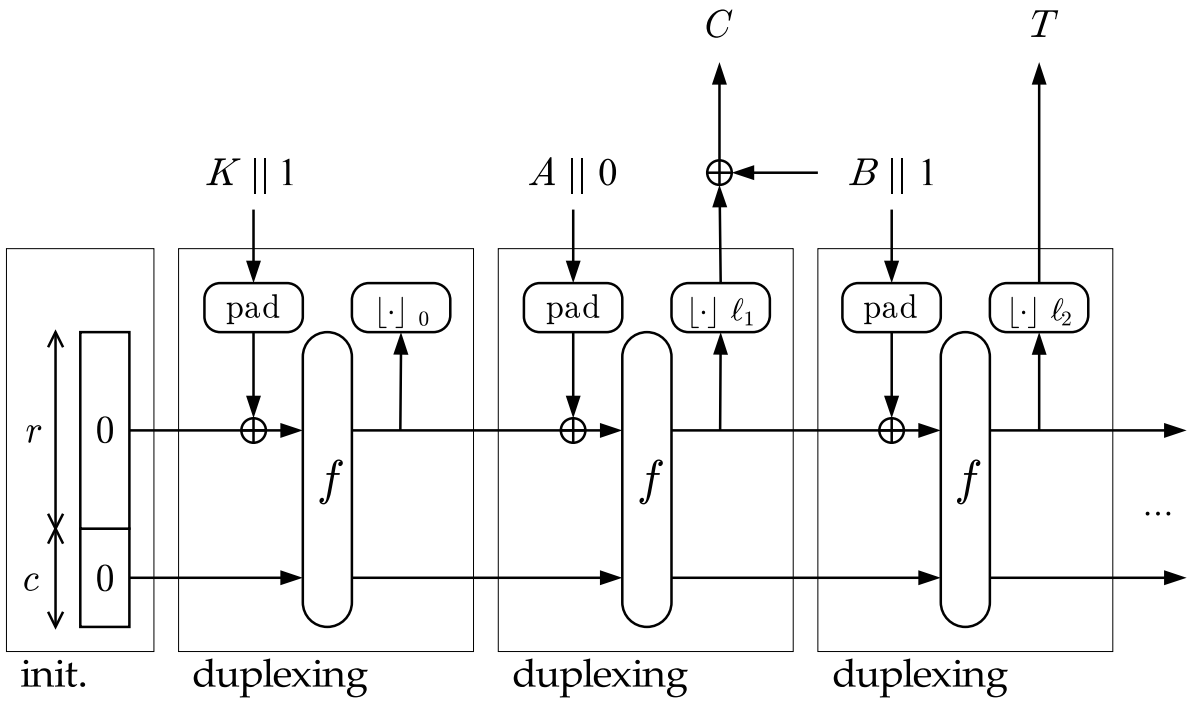


Wrapping with sponges

Sponge with *domain separation*: two passes



The SpongeWrap mode: a simple case



- Features:
  - Single-pass, unlike GCM and CCM
  - Unpatented, unlike OCB

Other applications

- Reseedable pseudorandom sequence generator
  - Seeding and reseeding
  - Pseudo-random output depends on all past seeds
  - Forward secrecy
- Overwrite mode
  - Sponge state: $b = r + c$ bits
  - Overwrite mode: in between calls to f state is only $c + 1$ bits

Duplexing other hash function constructions?

Possible but in general less efficient:

- Output transformation: in each call to duplex
- Blank iterations: in each call to duplex
- Last block domain separation: 2 CF per duplex call
- Digest shorter than input blocks: MGF in duplex
- Long padding: reduces available rate