Radboud University

Introduction to permutation-based cryptography

Joan Daemen[1,2]

based on joint work with
Guido Bertoni[1], Michaël Peeters[1], Gilles Van Assche[1],
Ronny Van Keer[1] Bart Mennink[2] and Seth Hoffert

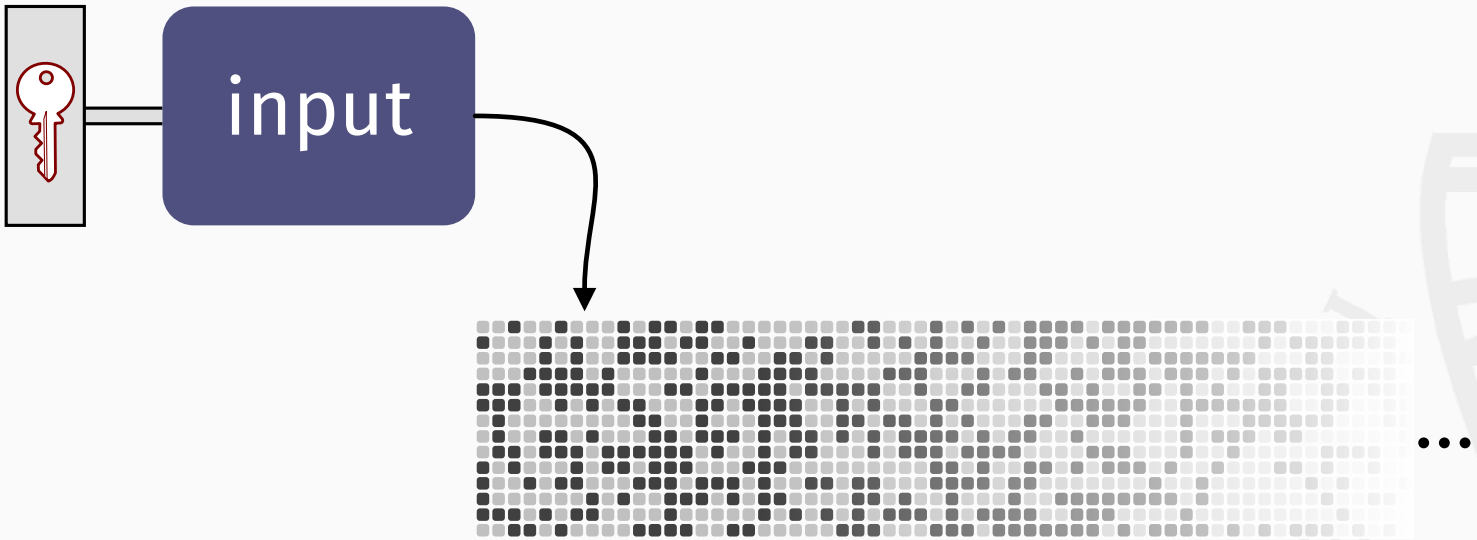
Croatia Summer School 2017

[1]STMicroelectronics [2]Radboud University

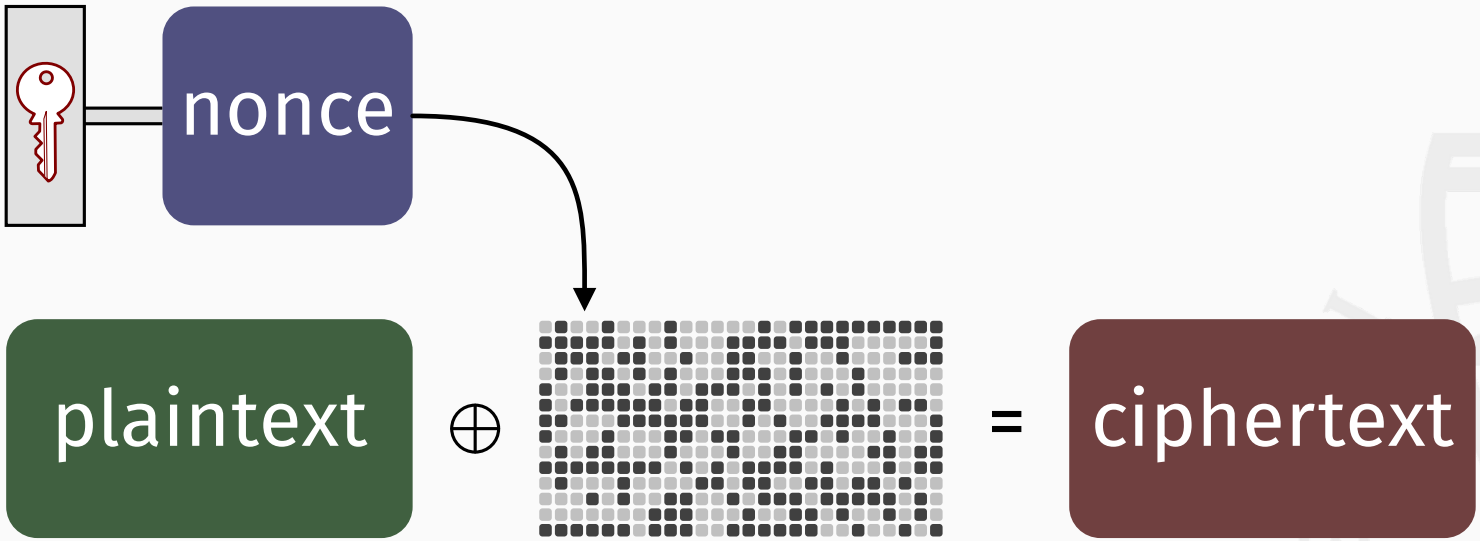
Outline

Outline

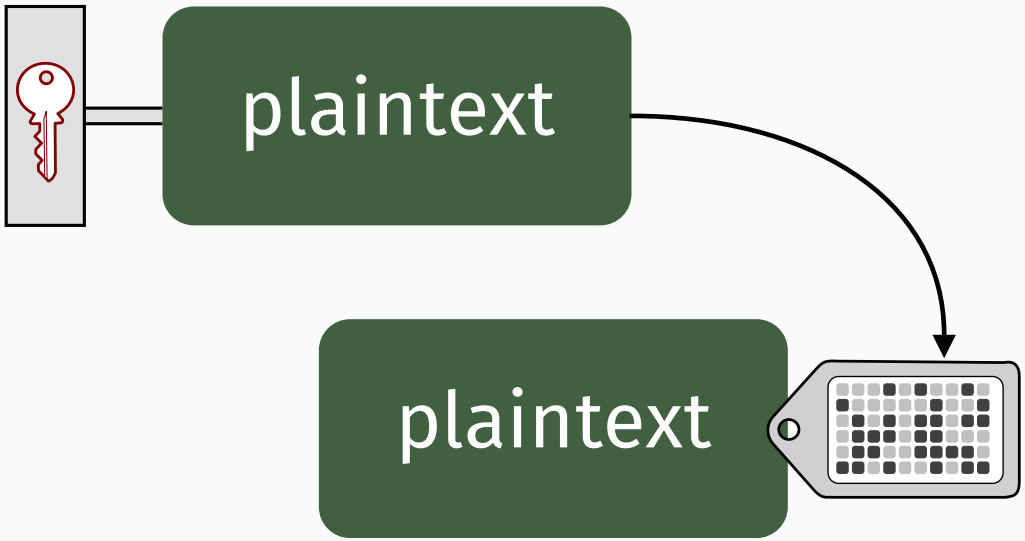
Pseudo-random function (PRF)



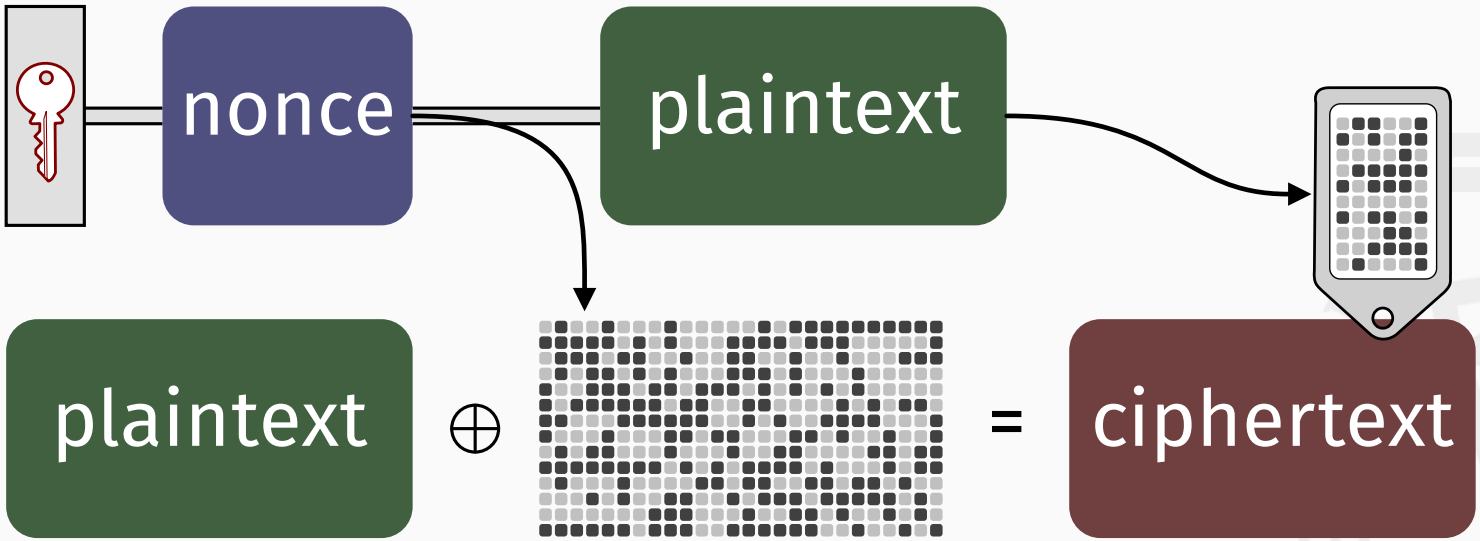
Stream encryption



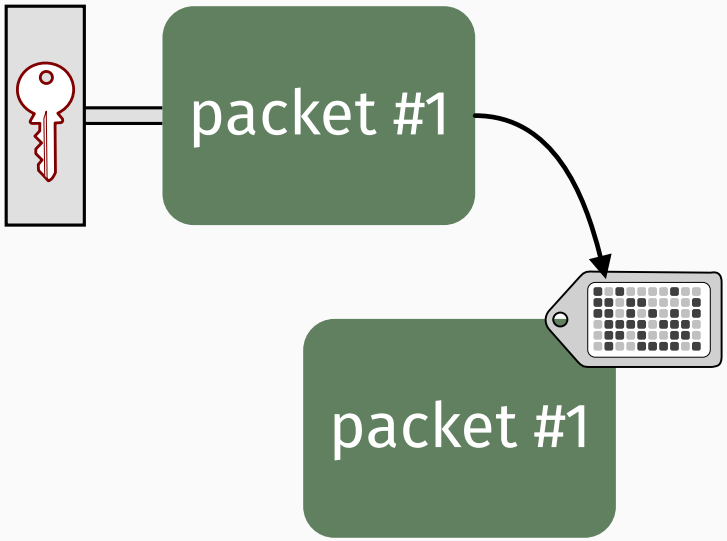
Message authentication (MAC)



Authenticated encryption

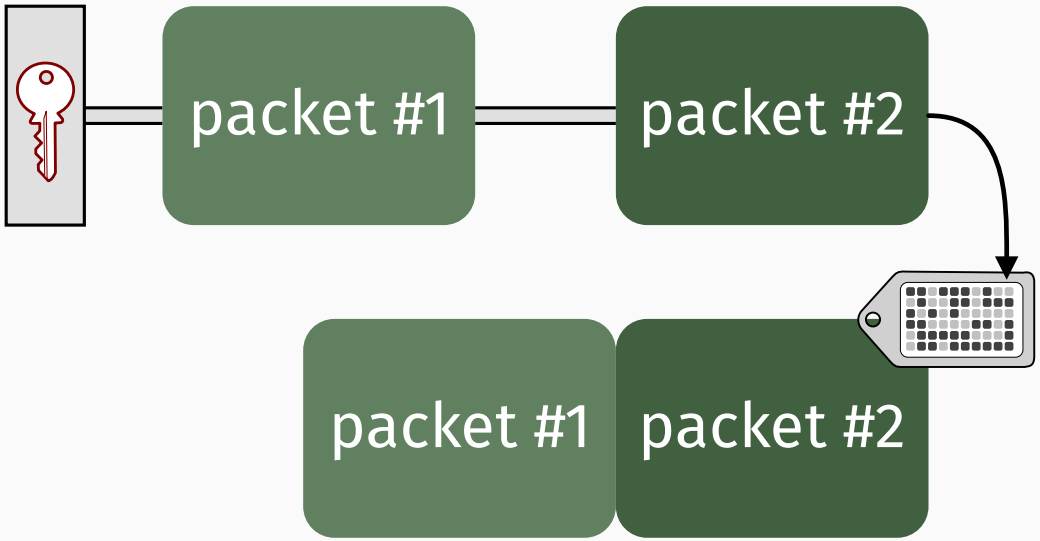


String sequence input and incrementality



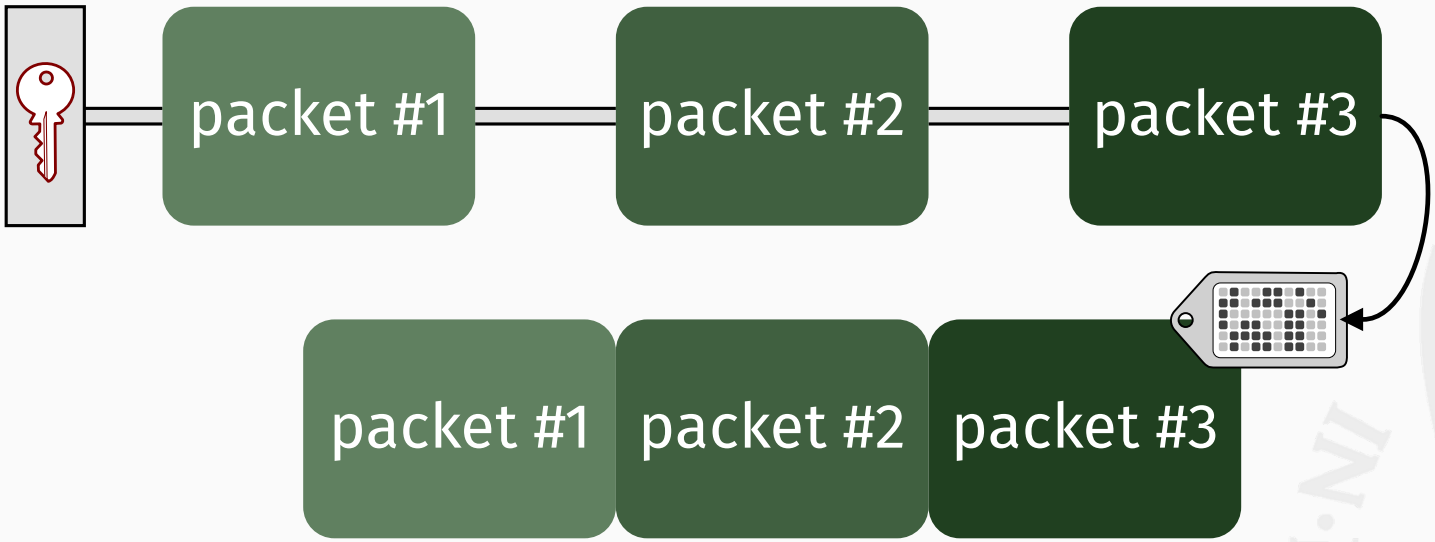
$$F_K\left(P^{(1)}\right)$$

String sequence input and incrementality



$$F_K\left(P^{(2)} \circ P^{(1)}\right)$$

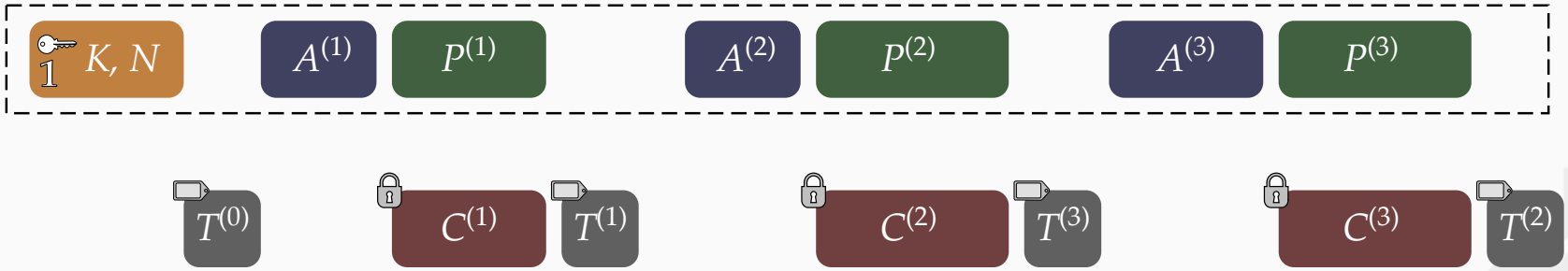
String sequence input and incrementality



$$F_K\left(P^{(3)} \circ P^{(2)} \circ P^{(1)}\right)$$

Outline

Session authenticated encryption (SAE) [Keccak Team, SAC 2011]



Initialization taking nonce N

$T \leftarrow 0^t + F_K(N)$

$\mathrm{history} \leftarrow N$

return tag T of length t

Wrap taking metadata A and plaintext P

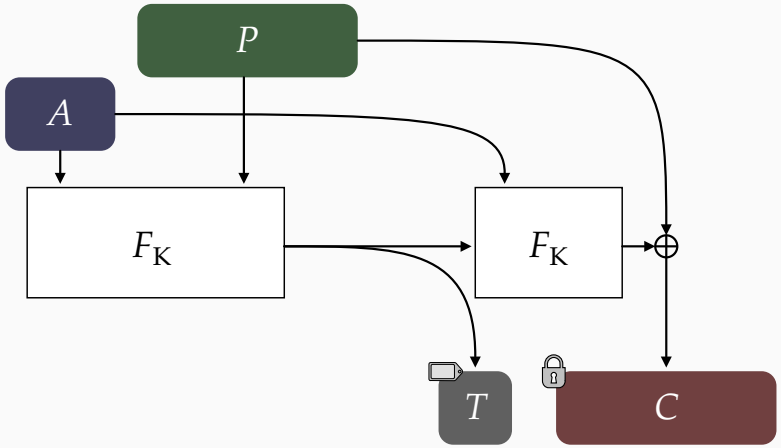
$C \leftarrow P + F_K(A \circ \mathrm{history})$

$T \leftarrow 0^t + F_K(C \circ A \circ \mathrm{history})$

$\mathrm{history} \leftarrow C \circ A \circ \mathrm{history}$

return ciphertext C of length $|P|$ and tag T of length t

Synthetic initialization value (SIV), as in [eprint 2016/1188]



Wrap taking metadata A and plaintext P
$T \leftarrow 0^t + F_K(P \circ A)$
$C \leftarrow P + F_K(T \circ A)$
return ciphertext C of length $|P|$ and tag T

Unwrap taking metadata A, ciphertext C and tag T
$P \leftarrow C + F_K(T \circ A)$
$\tau \leftarrow 0^t + F_K(P \circ A)$
if $\tau \neq T$ **then return** error!
else return plaintext P of length $|C|$

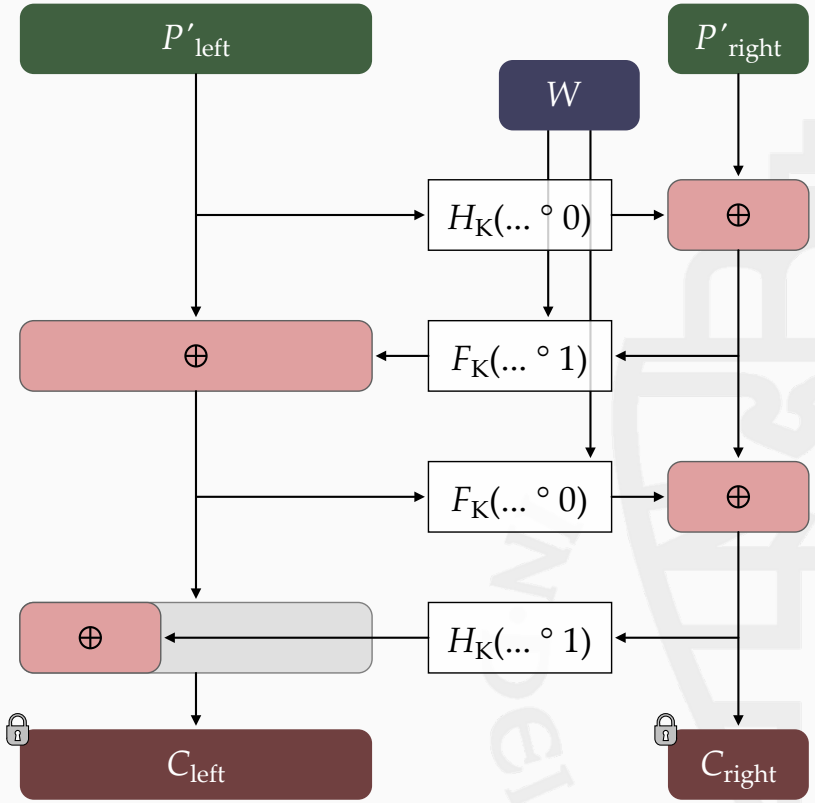
Variant of SIV of [Rogaway & Shrimpton, EC 2006]

Wide block cipher (WBC), as in [eprint 2016/1188]

Encipher P with K and tweak W

$$
\begin{array}{llll}
(L, R) & \leftarrow & & \mathrm{split}(P) \\
R_0 & \leftarrow R_0 & + & H_K(L \circ 0) \\
L & \leftarrow L & + & F_K(R \circ W \circ 1) \\
R & \leftarrow R & + & F_K(L \circ W \circ 0) \\
L_0 & \leftarrow L_0 & + & H_K(R \circ 1) \\
C & \leftarrow L & || & R
\end{array}
$$

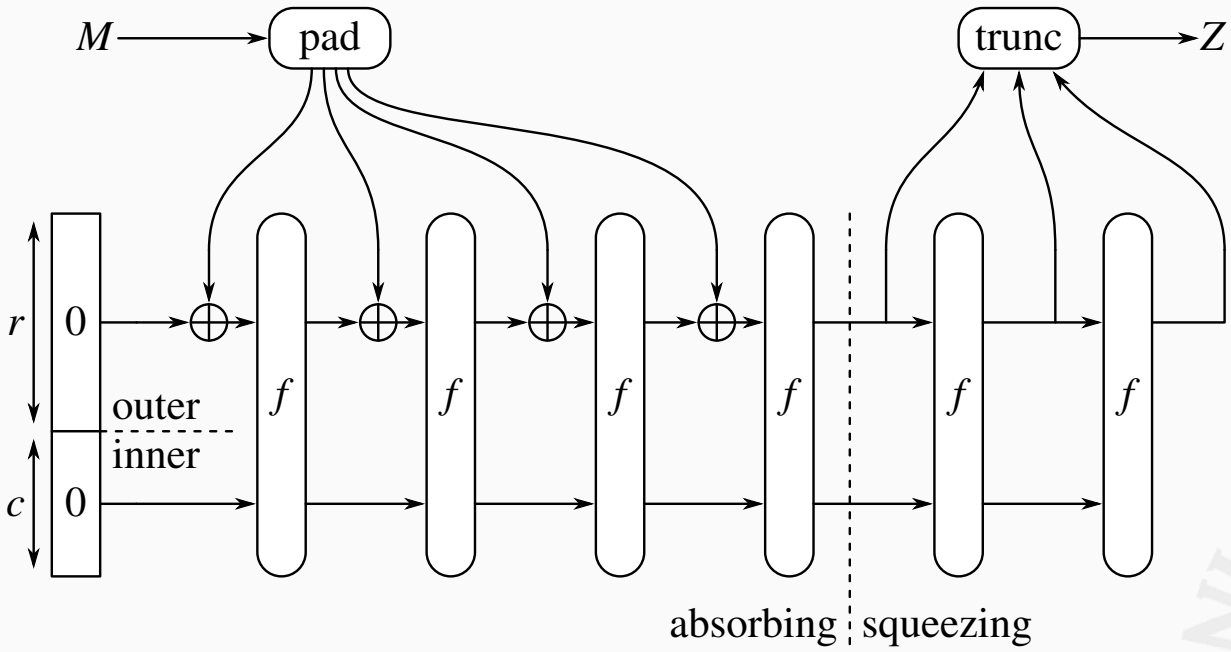
return ciphertext C of length $|P|$



Instance of HHFHFH of [Bernstein, Nandi & Sarkar, Dagstuhl 2016]

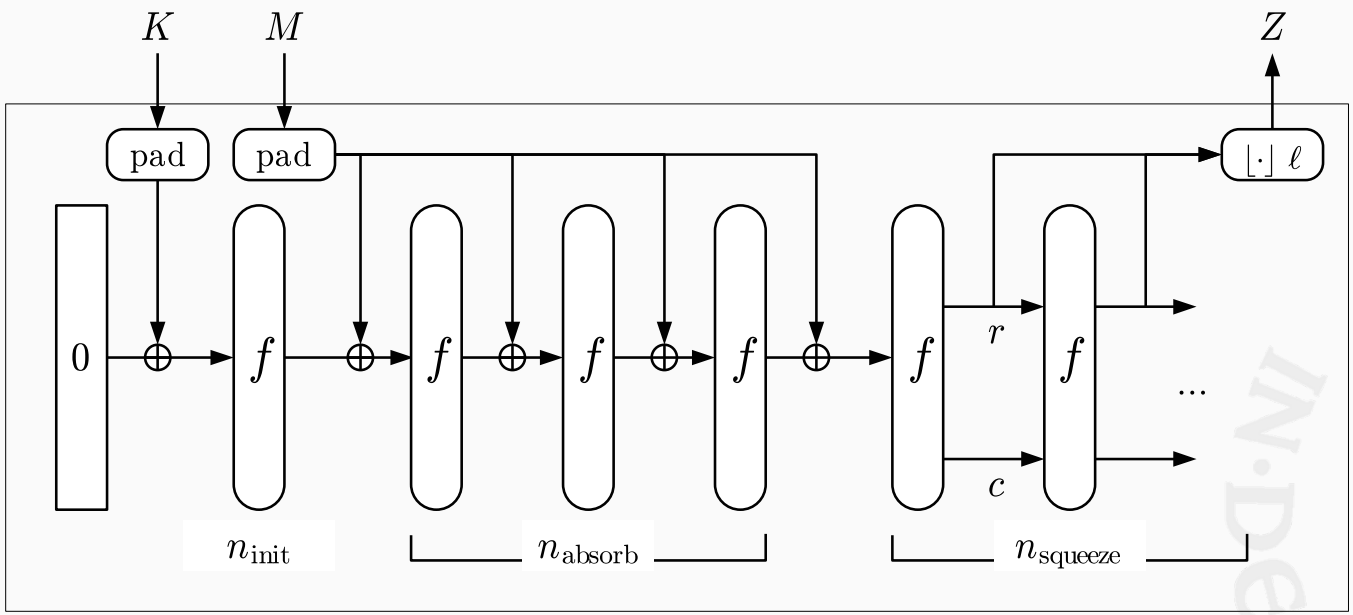
Outline

Sponge [Keccak Team, Ecrypt 2008]



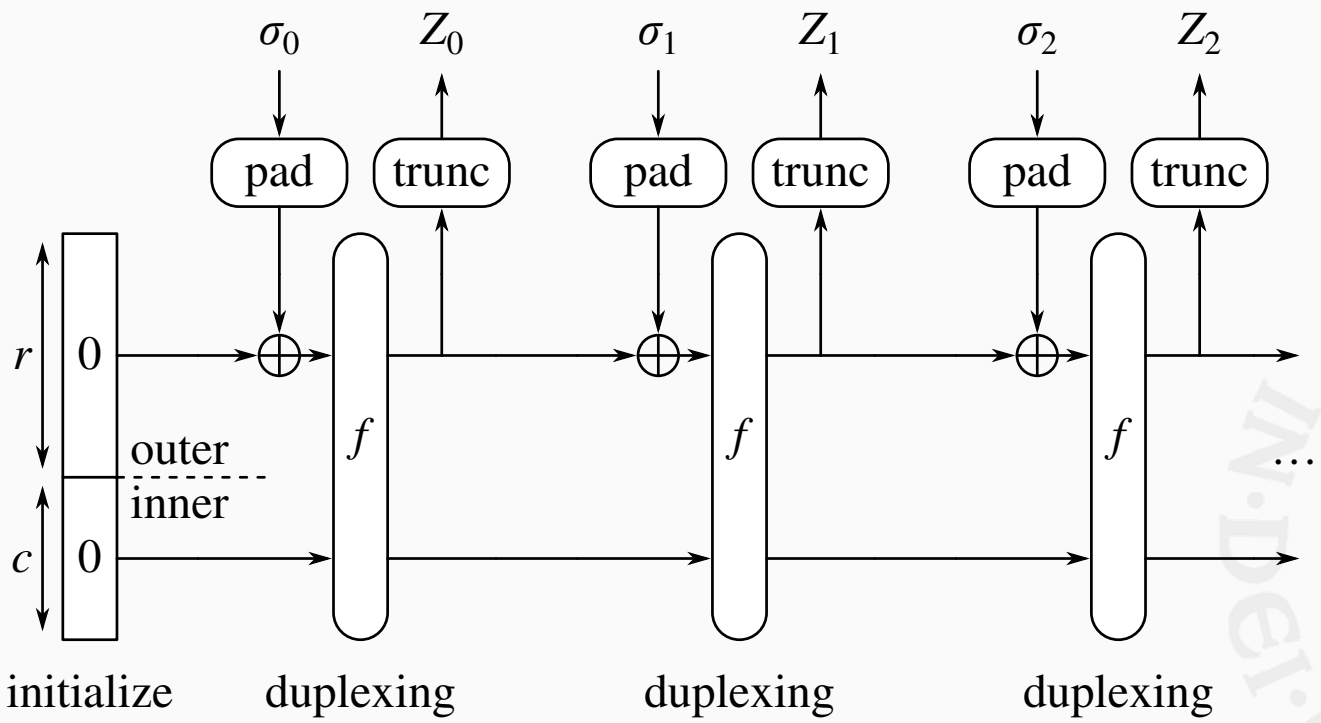
- f is a cryptographic permutation
  - *if f is random, this offers $c/2$ bits of security*
  - design f *such that this is secure*: public scrutiny
- Pre-pending M with K gives PRF

More efficient: donkeySponge [Keccak Team, DIAC 2012]

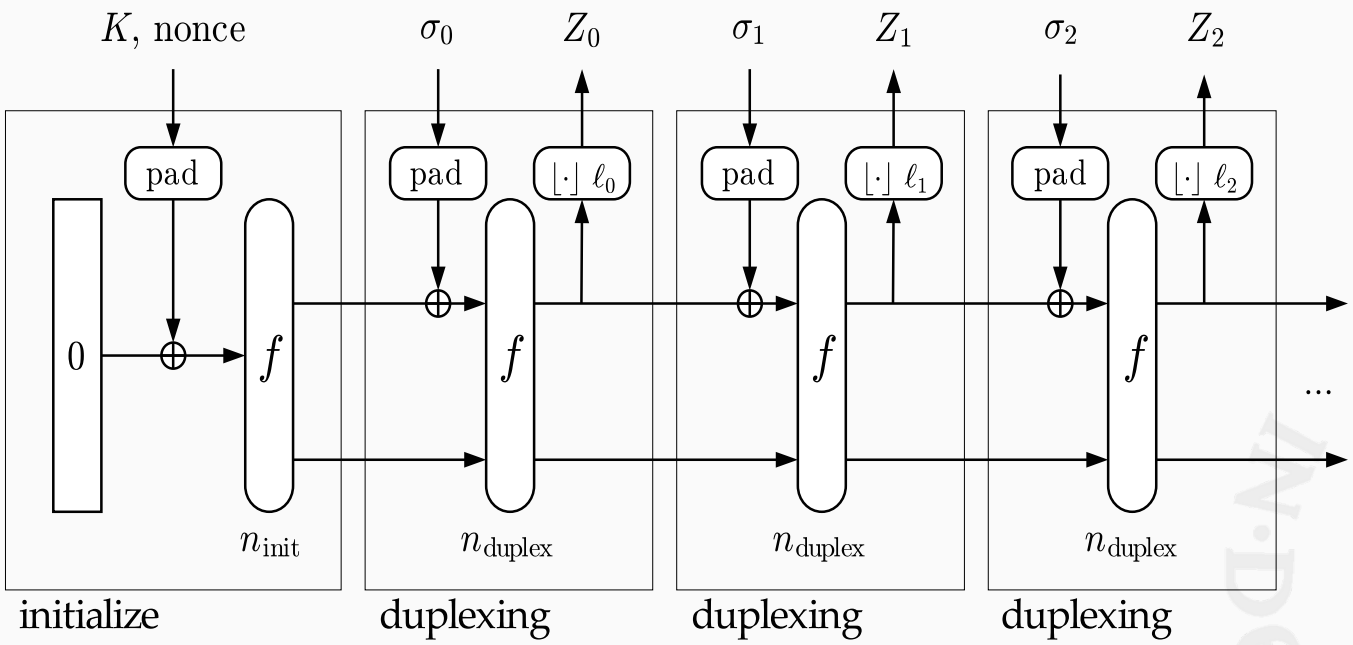


donkey sponge

Incrementality: duplex [Keccak Team, SAC 2011]

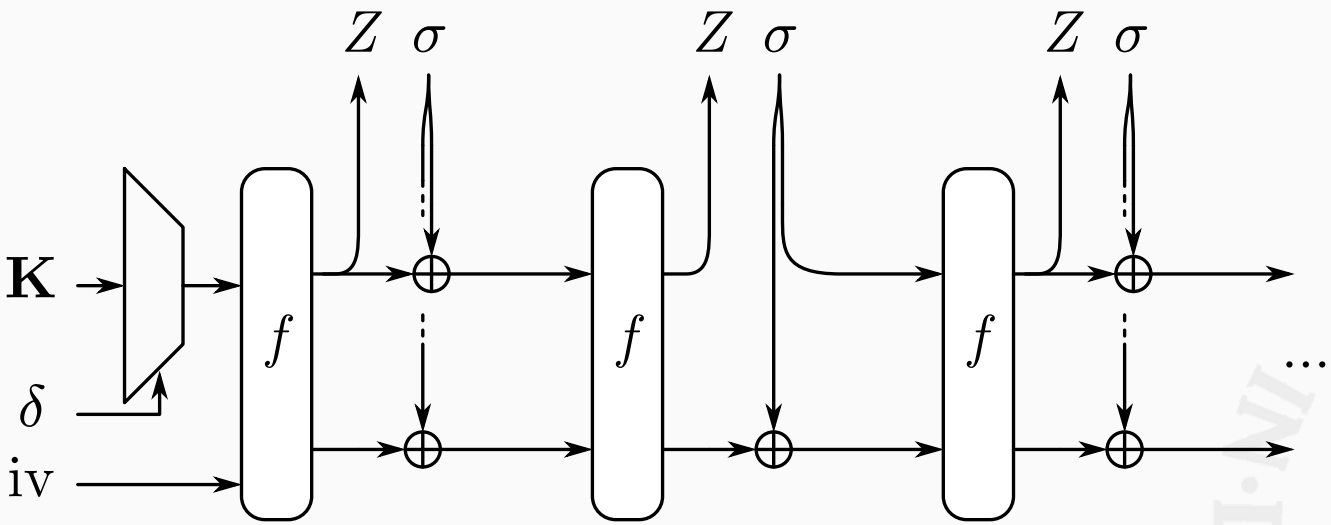


More efficient: MonkeyDuplex [Keccak Team, DIAC 2012]



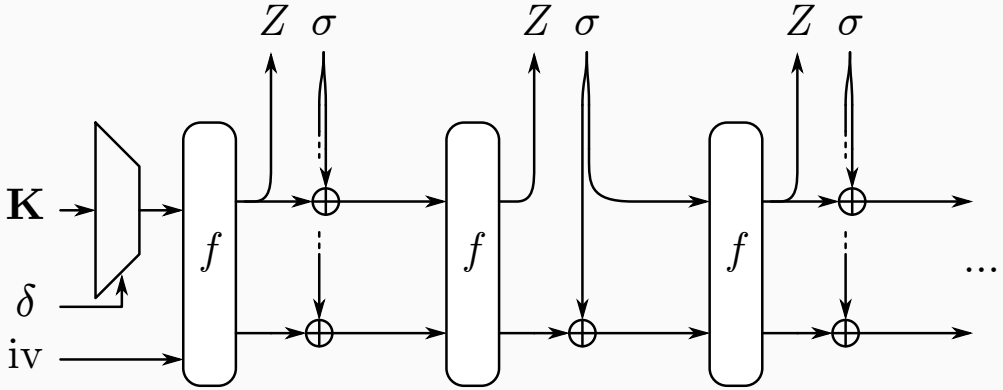
Instances: KETJE + half a dozen other CAESAR submissions

Consolidation: Full-state keyed duplex

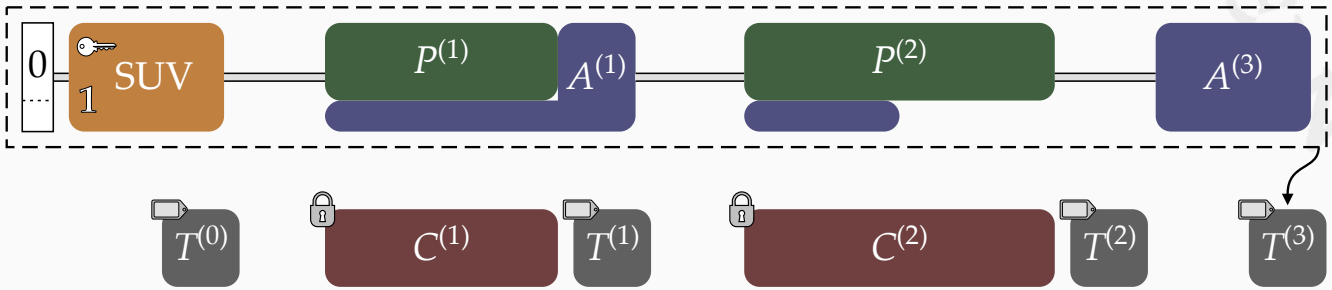


[Mennink, Reyhanitabar, & Vizar, AC 2015], [Bart, Gilles and me, eprint 2017/498]

Modes on top of keyed duplex

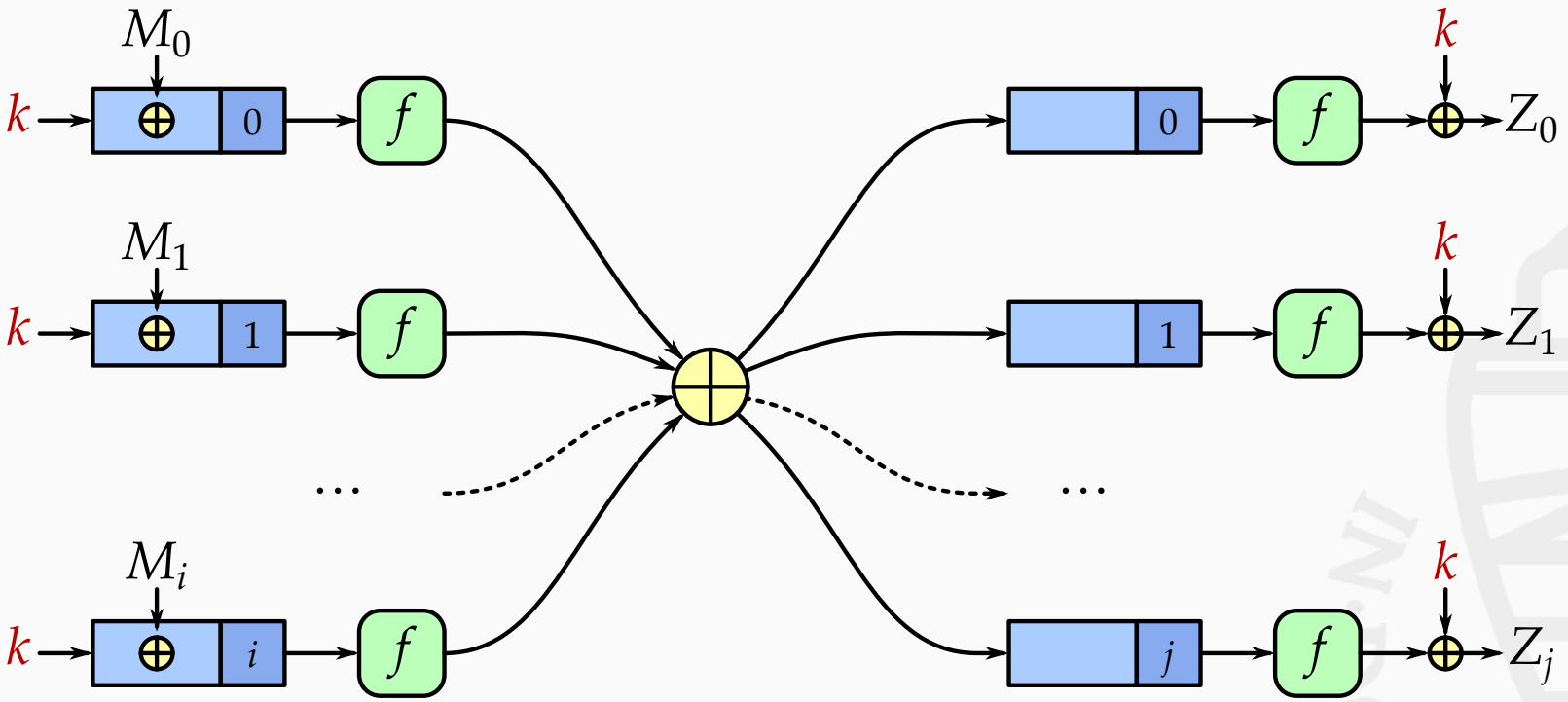


- iv=counter encryption: security $c = b - r$ bits
- MAC: security $c/2$ bits
- SAE: Motorist [Keyak 2015]: security $c - \epsilon$



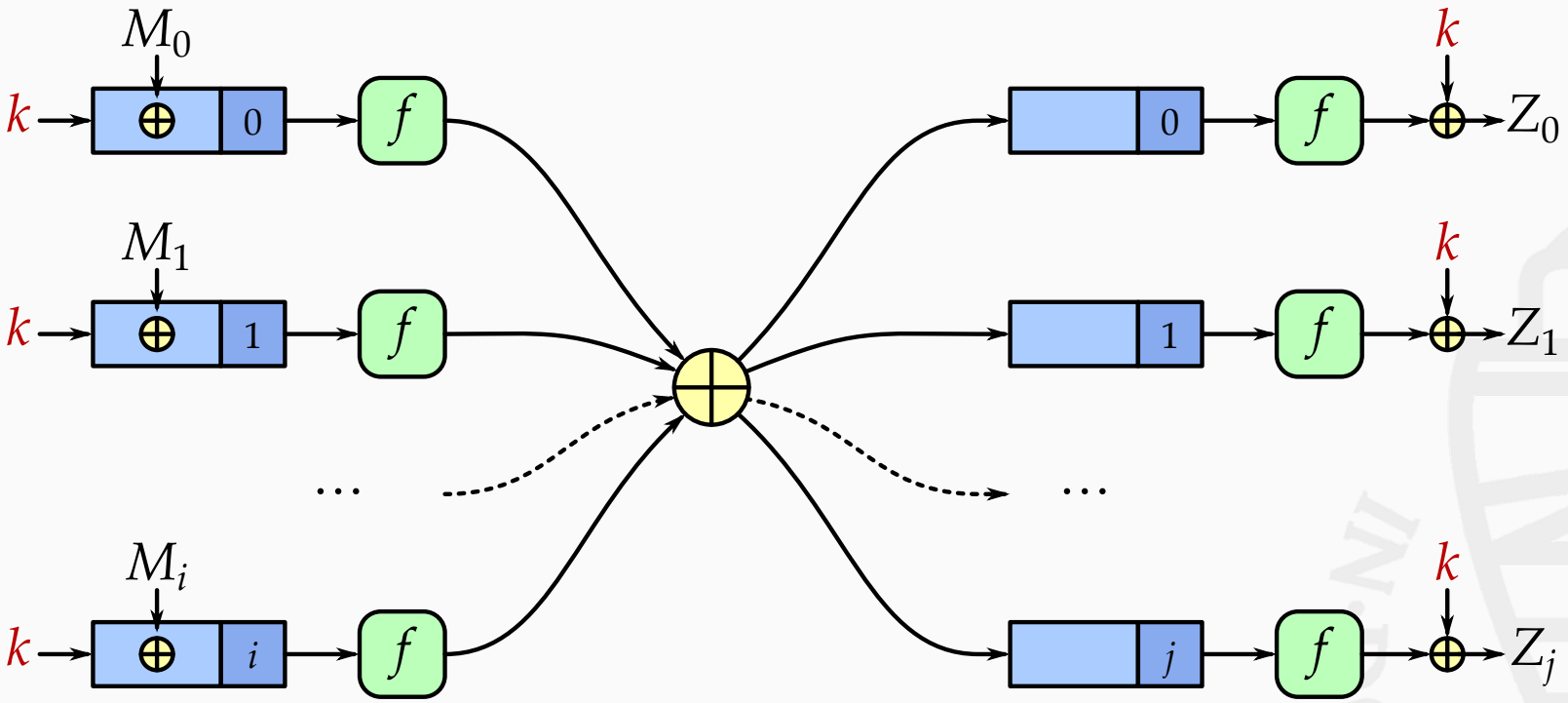
Outline

A parallel sponge variant: early attempt [Keccak Team 2014-2016]



Similar to Protected Counter Sums [Bernstein, "stretch", JOC 1999]

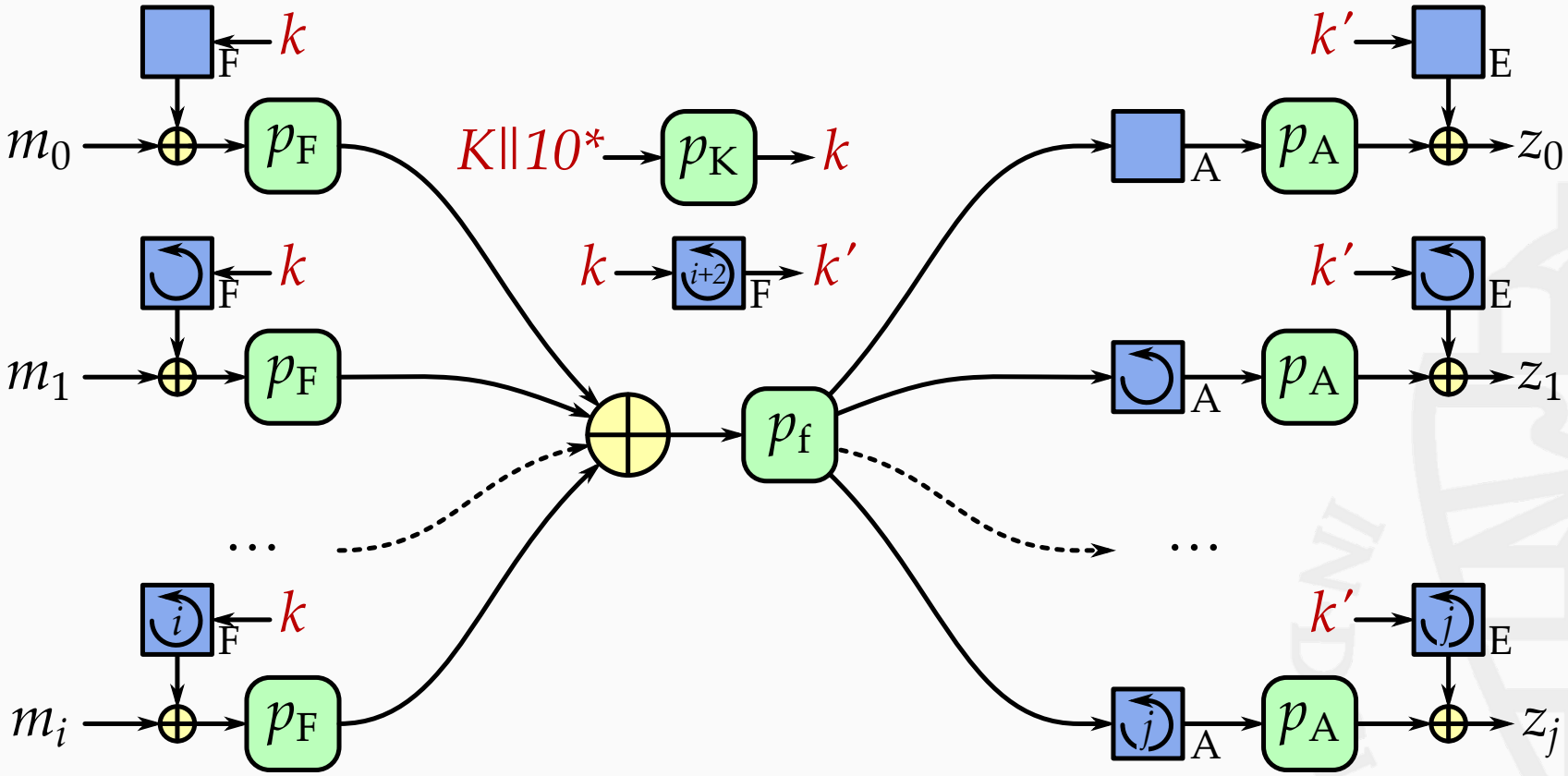
A parallel sponge variant: early attempt [Keccak Team 2014-2016]



Similar to Protected Counter Sums [Bernstein, "stretch", JOC 1999]

Problem: collisions with higher-order differentials if f has low degree

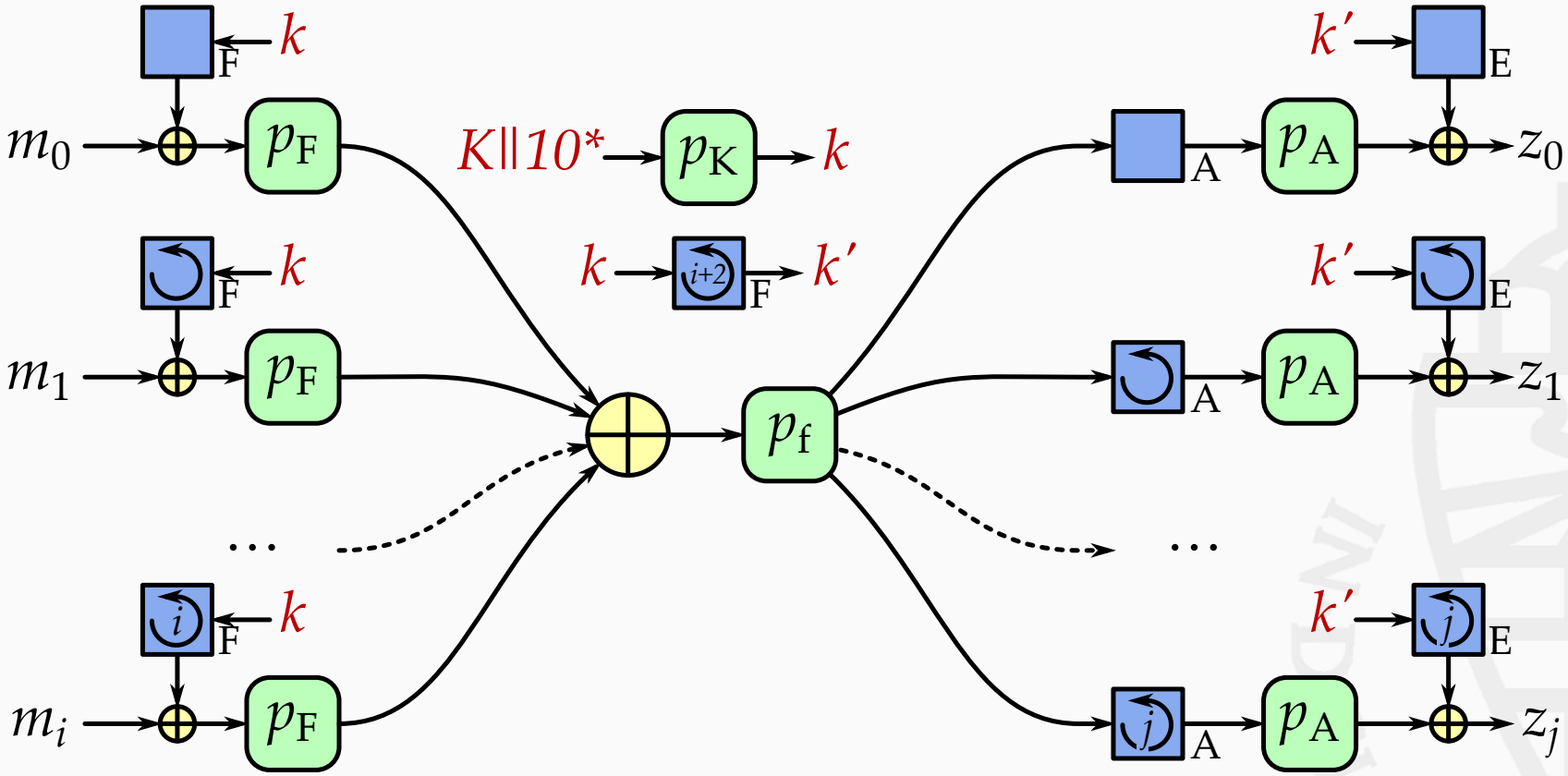
Farfalle now [Keccak Team, eprint 2016/1188, update TODO]



- Input mask rolling and f against accumulator collisions
- State rolling, f and output mask against state retrieval from output
- Middle f against higher-order DC
- Input-output attacks would span 3 f layers

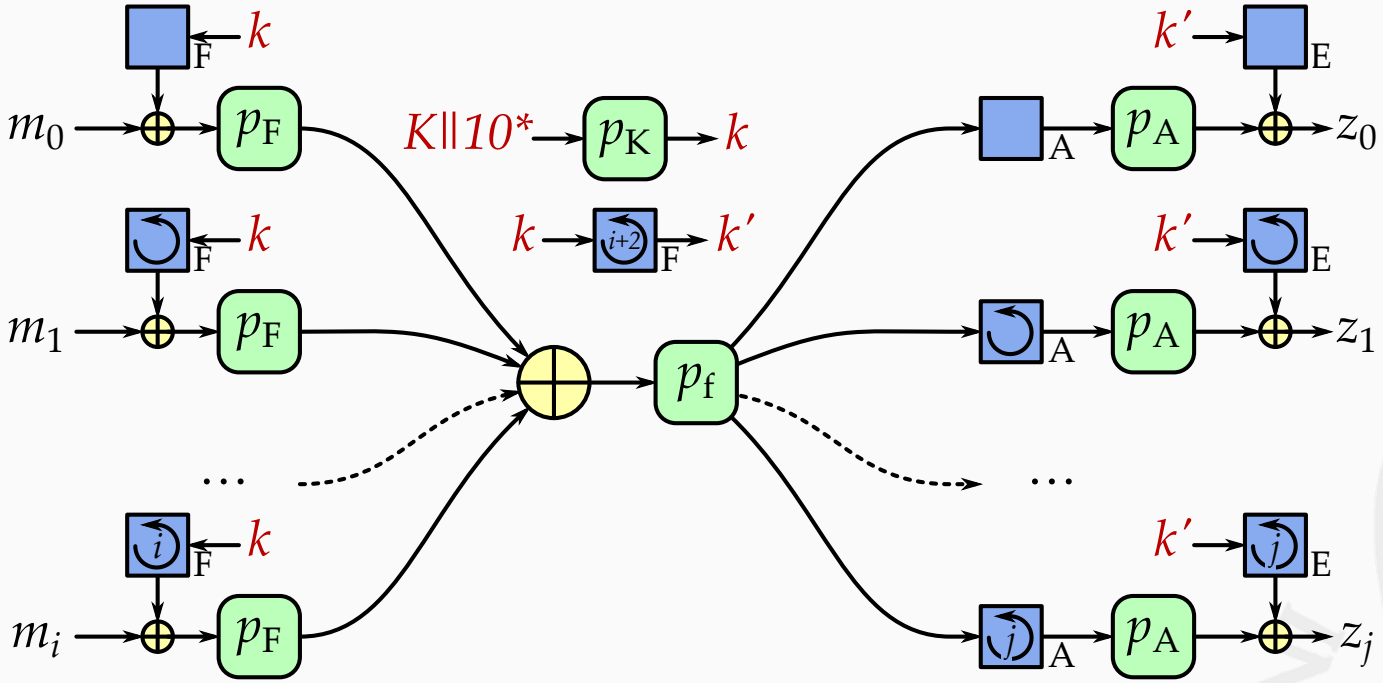
Outline

KRAVATTE = Farfalle with KECCAK-p



- Target security: 128 bits, incl. multi-target
- $f = \text{KECCAK-}p[1600, n_r]$ with $n_r = 6, 4, 4$
- Rolling function: operates on 5 lanes only, LFSR-like with order $2^{320} - 1$
  - lightweight, inspired by [Granger, Jovanonvic, Mennink & Neves, EC 2016]

KRAVATTE performance



- Work per byte for long messages, compared to SHA3-256
  - MAC: 1/5
  - encryption: 1/8
  - SAE: 1/3
  - SIV: 1/3
- …and fully parallelizable

Conclusions

- Symmetric cryptography can be built up with
  - PRF as basic function
  - permutations as the building block to build PRFs
- Simple and potentially efficient

Thanks for your attention!

Links to software:

- `https://github.com/gvanas/KeccakCodePackage`
- `https://github.com/gvanas/KeccakTools`