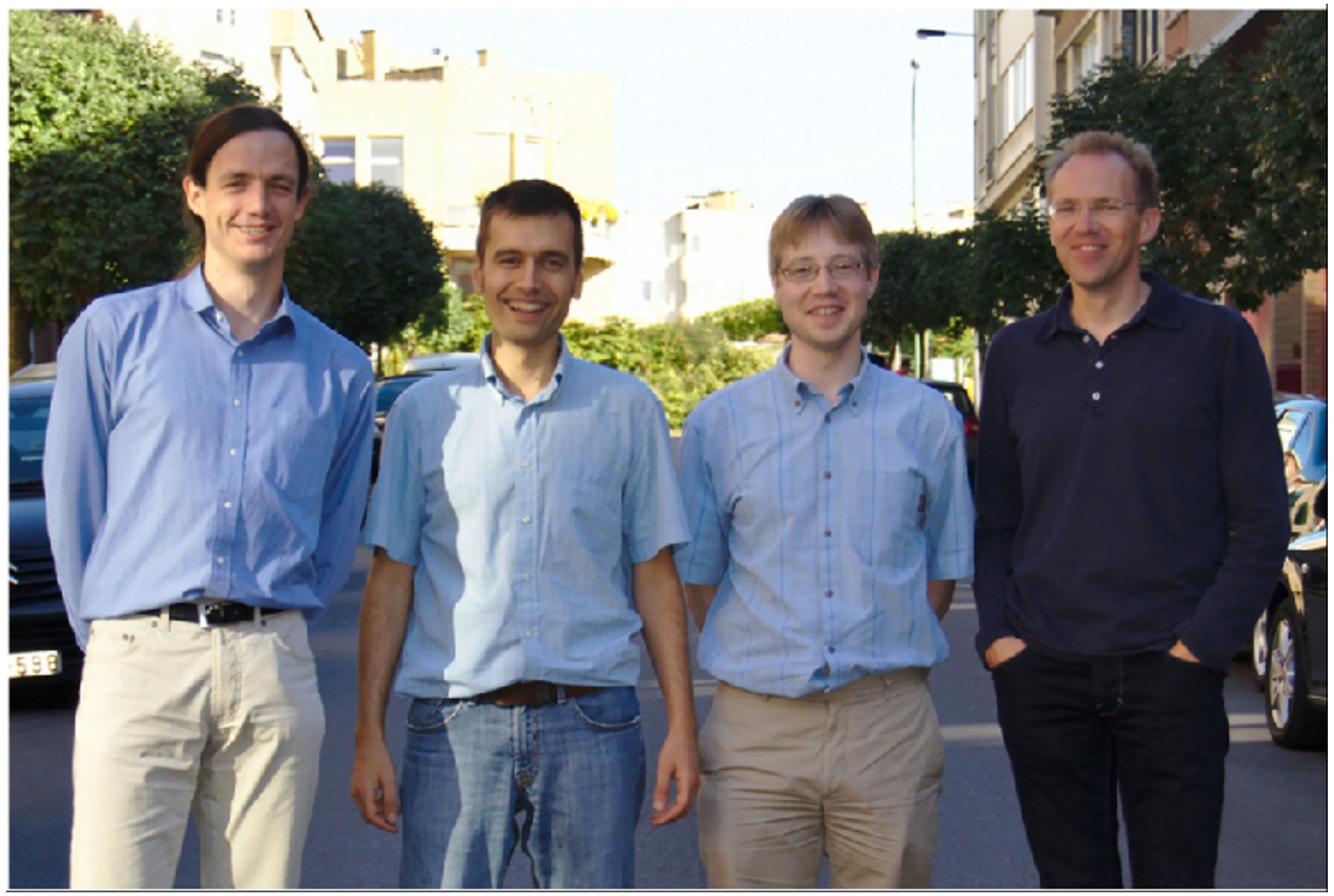
Lecture 19: Cryptanalysis

- Lab 10 is due Wednesday 4/24 at 11pm
- Lab 11 will be posted soon, due Wednesday 5/1
- Online course evaluation is live

SHA-3: quest for a Merkle-Damgard alternative

- 2004: Weakness found in Merkle-Damgard, eventually would break SHA-1 in 2017
- 2007: Call for submissions
- 2008: 64 submissions received
- 2009-12: Three workshops, one before each cutdown: 64 → 51 → 14 → 5 → 1
- Oct 2012: Keccak announced as winner, created by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche
- Aug 2015: NIST publishes Federal Information Processing Standard (FIPS) 202 standardizing Keccak



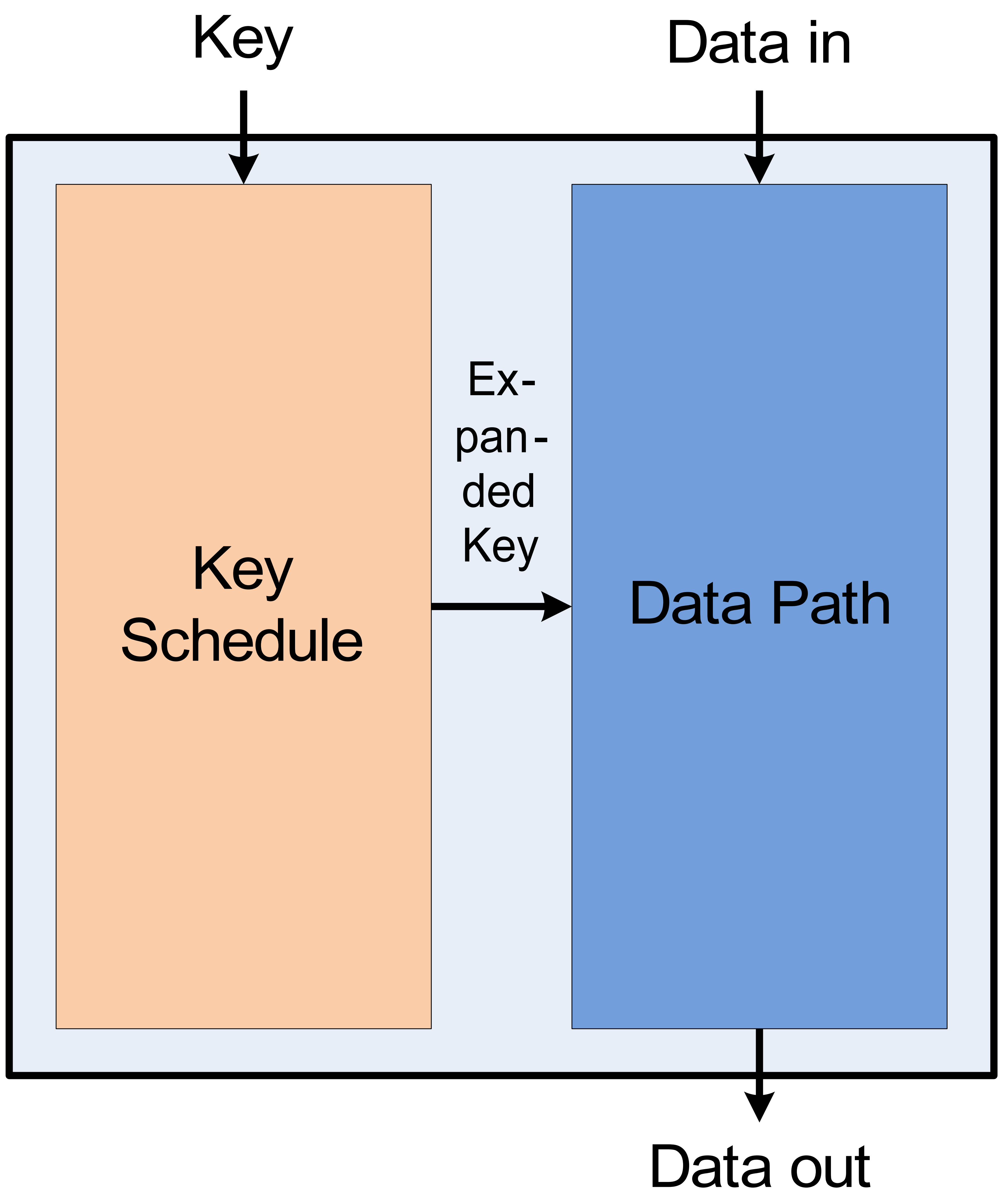
Why NIST chose Keccak, in their words

1. “Offers acceptable performance in software, and *excellent performance in hardware*.”

2. “Has a *large security margin*, suggesting a good chance of surviving without a practical attack during its working lifetime.”

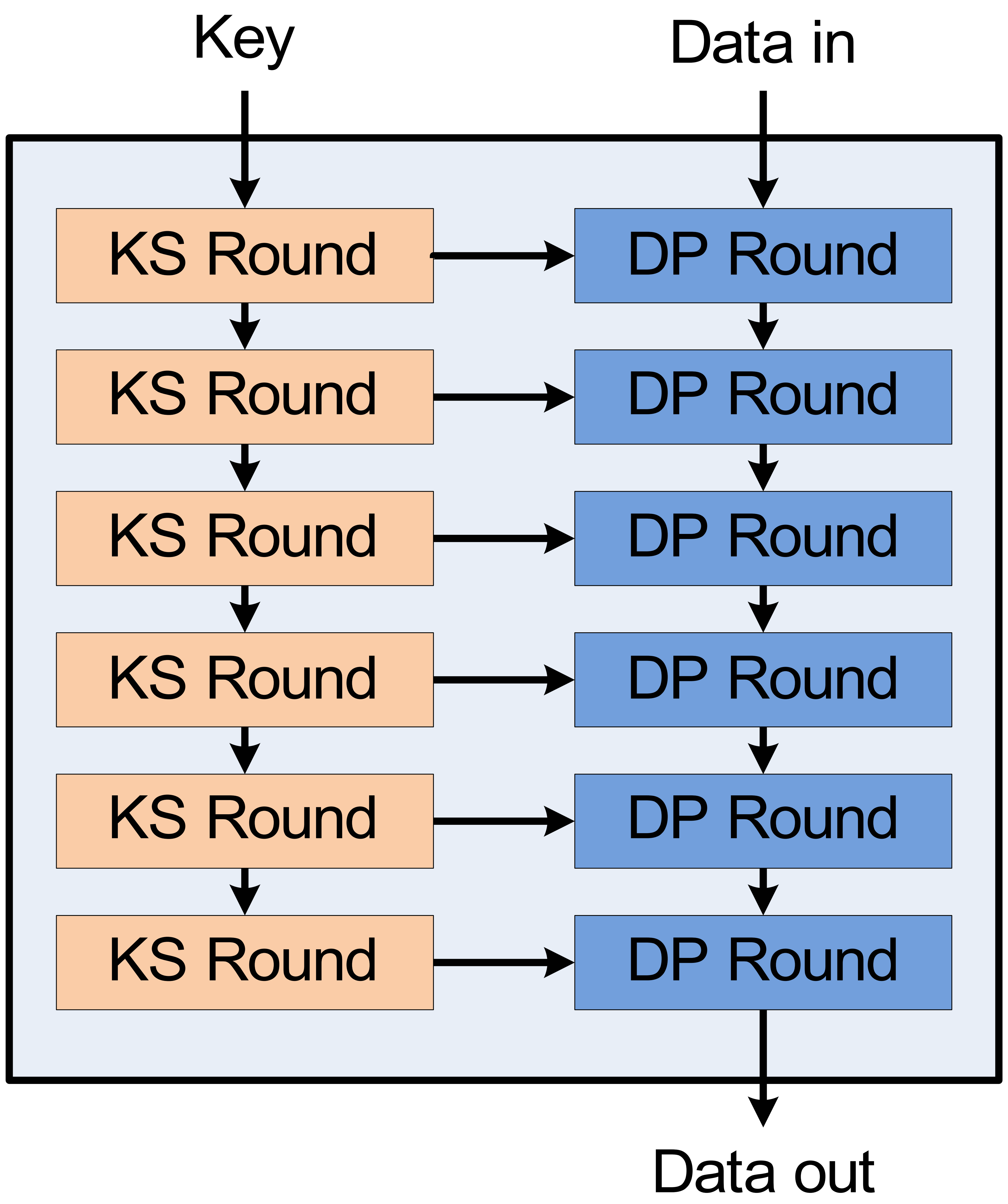
3. “A fundamentally new and different algorithm that is entirely *unrelated to the SHA-2 algorithms*.”

Block cipher operation



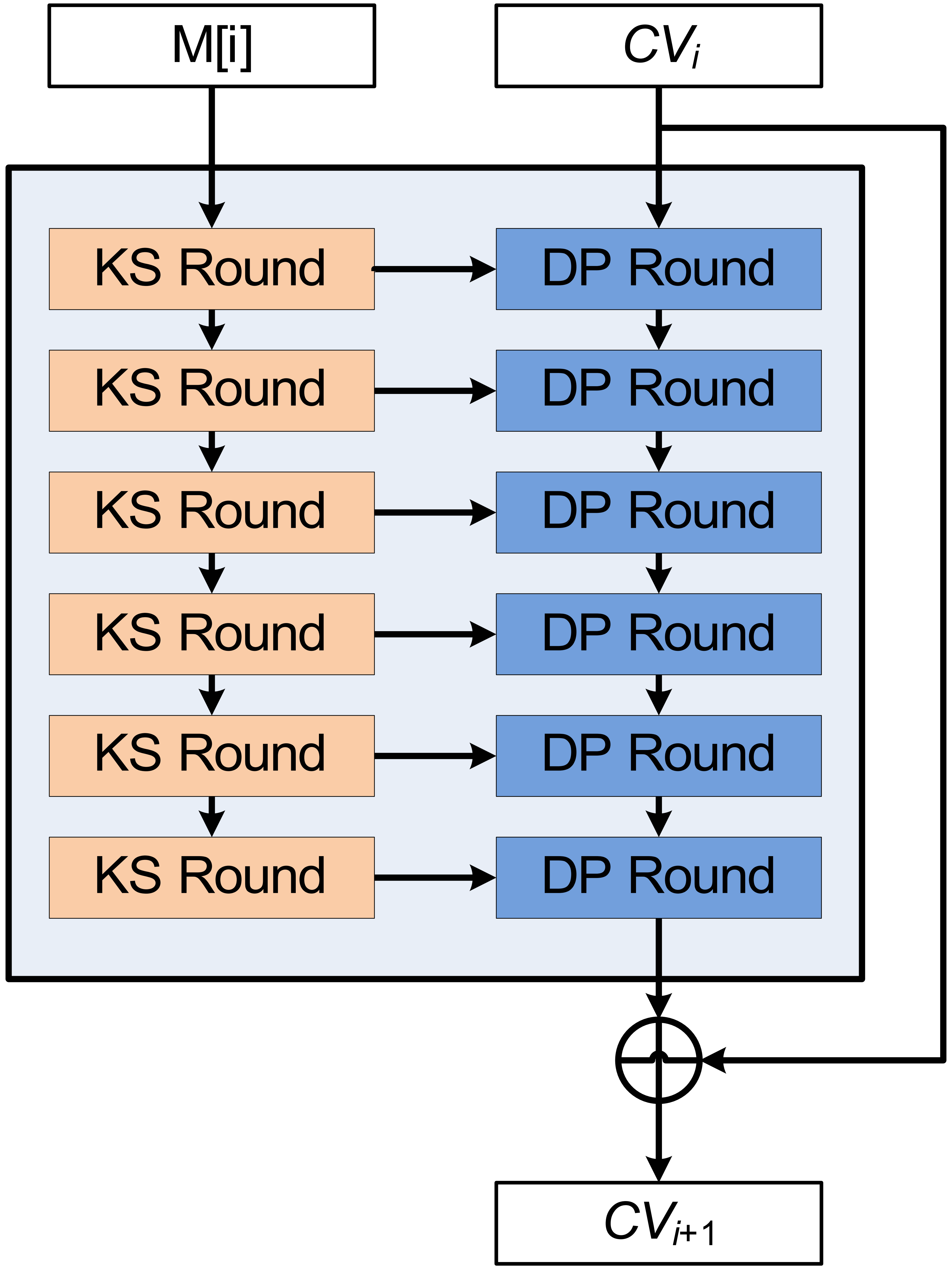
Slides from Keccak team

Block cipher internals



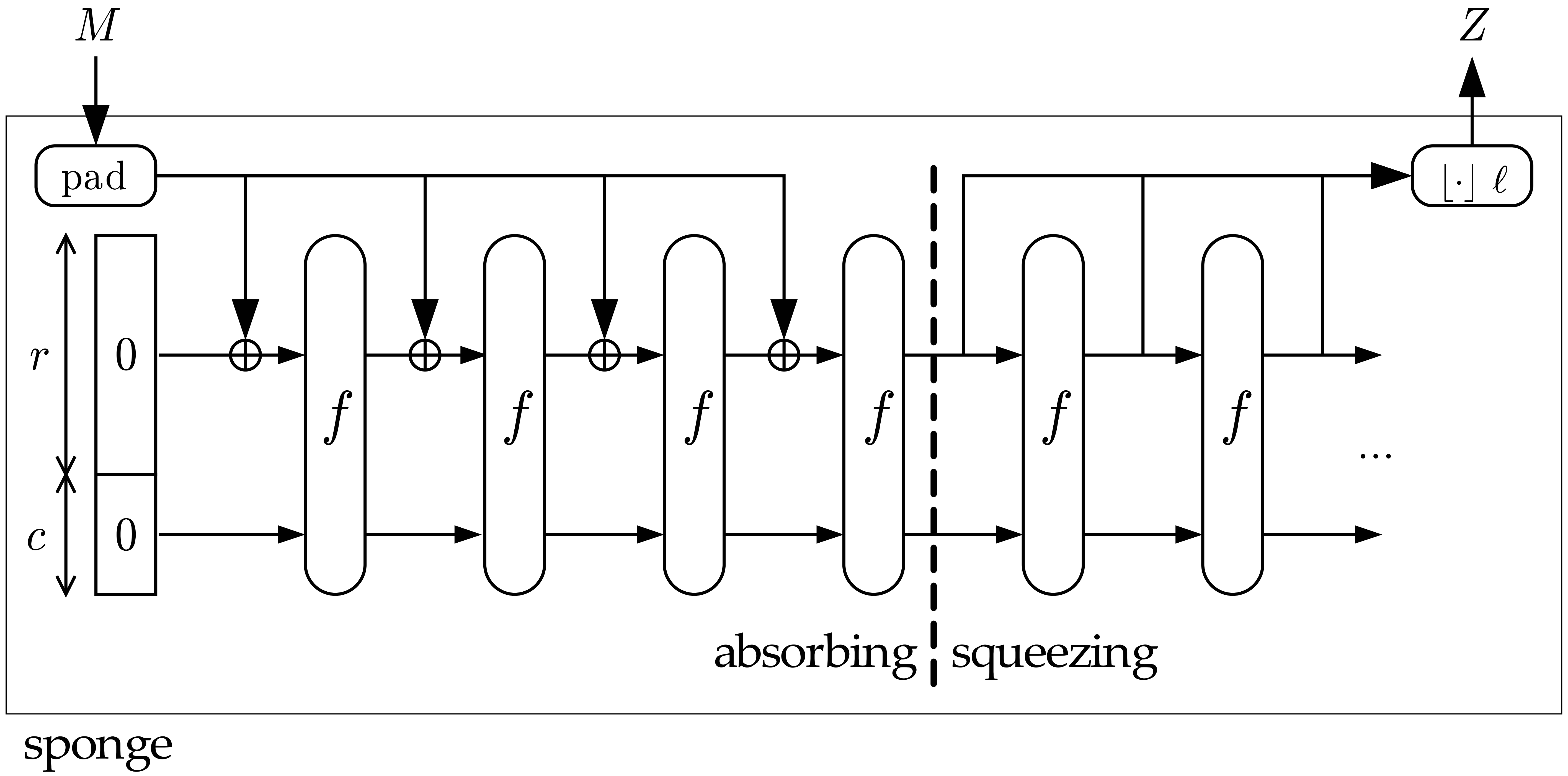
Slides from
Keccak team

Hashing use case: Davies-Meyer compression function



Slides from
Keccak team

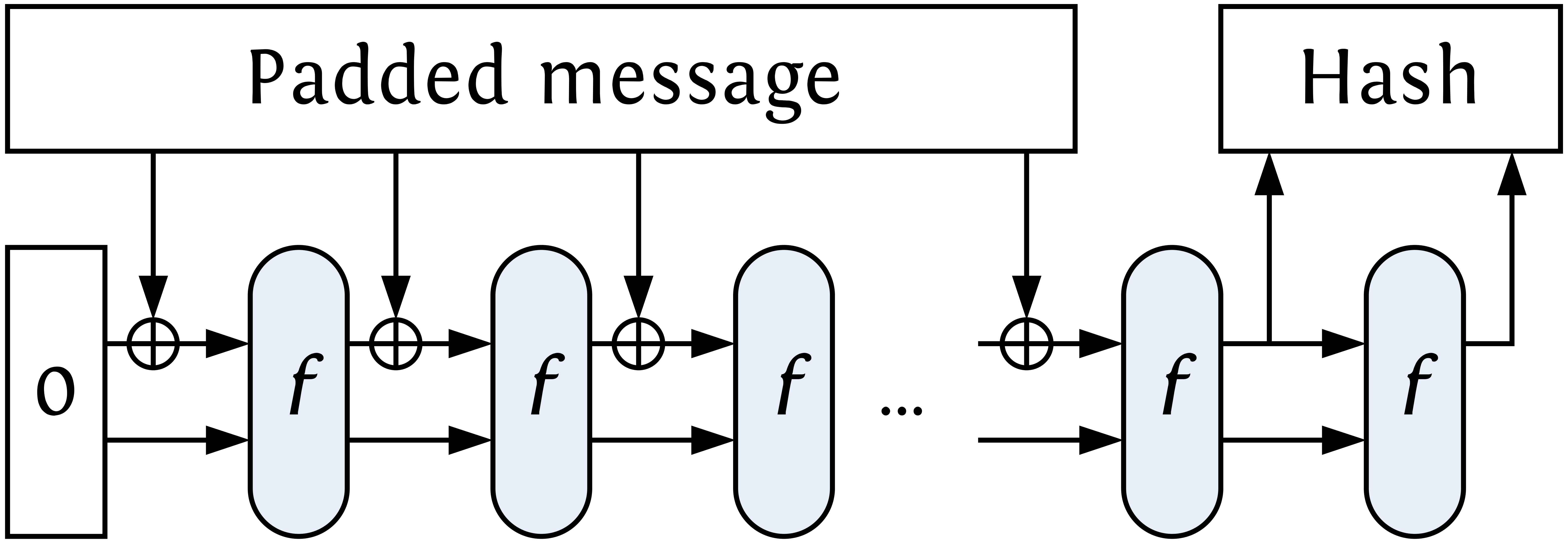
The result: the sponge construction



- f: a b-bit permutation with $b = r + c$
  - efficiency: processes r bits per call to f
  - security: provably resists generic attacks up to $2^{c/2}$
- Flexibility in trading rate r for capacity c or vice versa

Slides from Keccak team

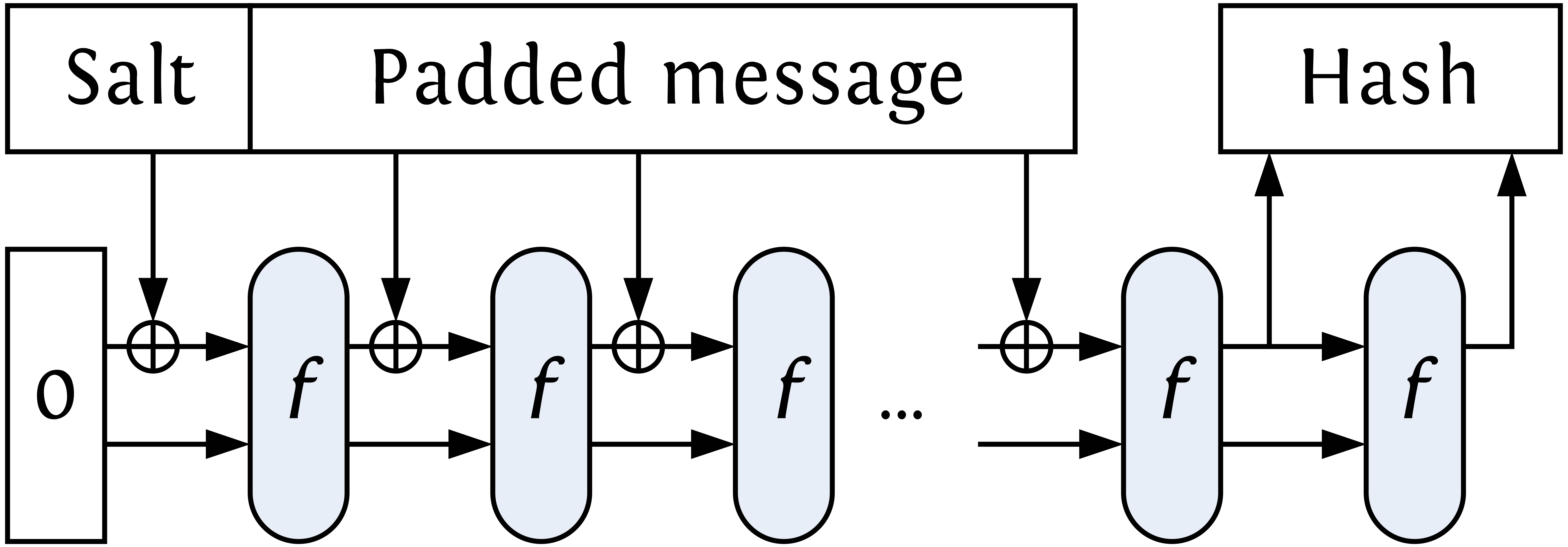
Regular hashing



- Electronic signatures
- Data integrity (*shaXsum* ...)
- Data identifier (*Git, online anti-virus, peer-2-peer* ...)

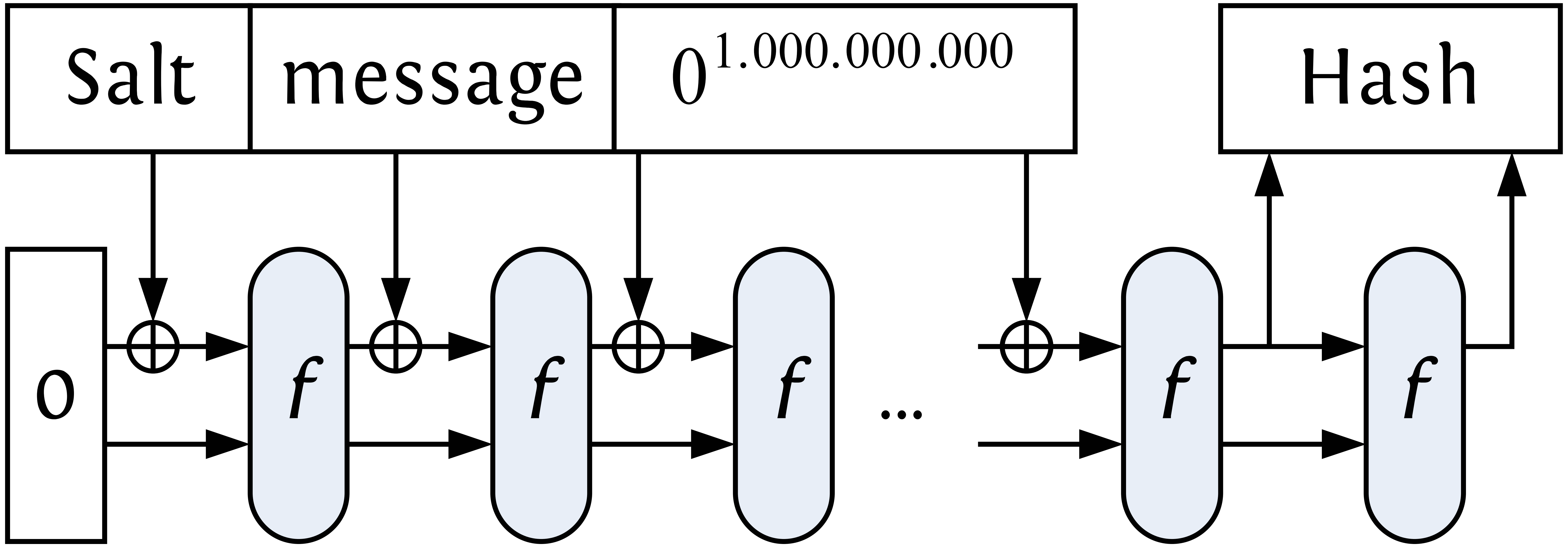
Slides from
Keccak team

Salted hashing



- Randomized hashing (RSASSA-PSS)
- Password storage and verification (*Kerberos*, /etc/shadow)

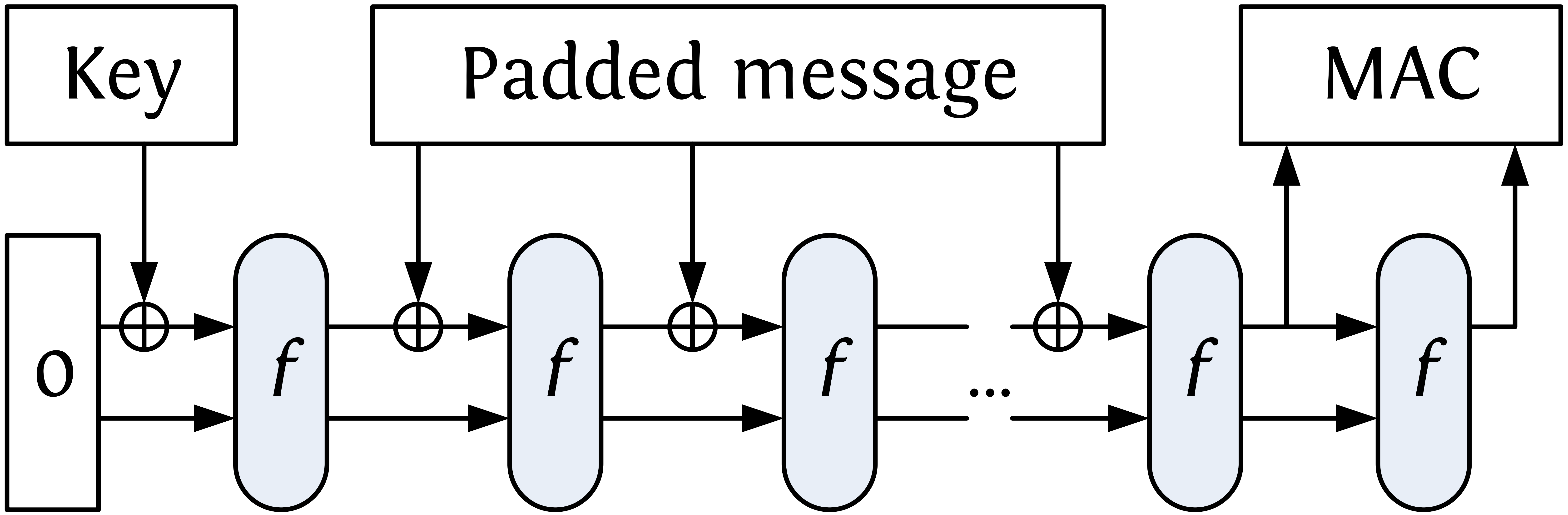
Salted hashing



- Randomized hashing (RSASSA-PSS)
- Password storage and verification (*Kerberos*, /etc/shadow)
  - ...Can be as slow as you like it!

Slides from
Keccak team

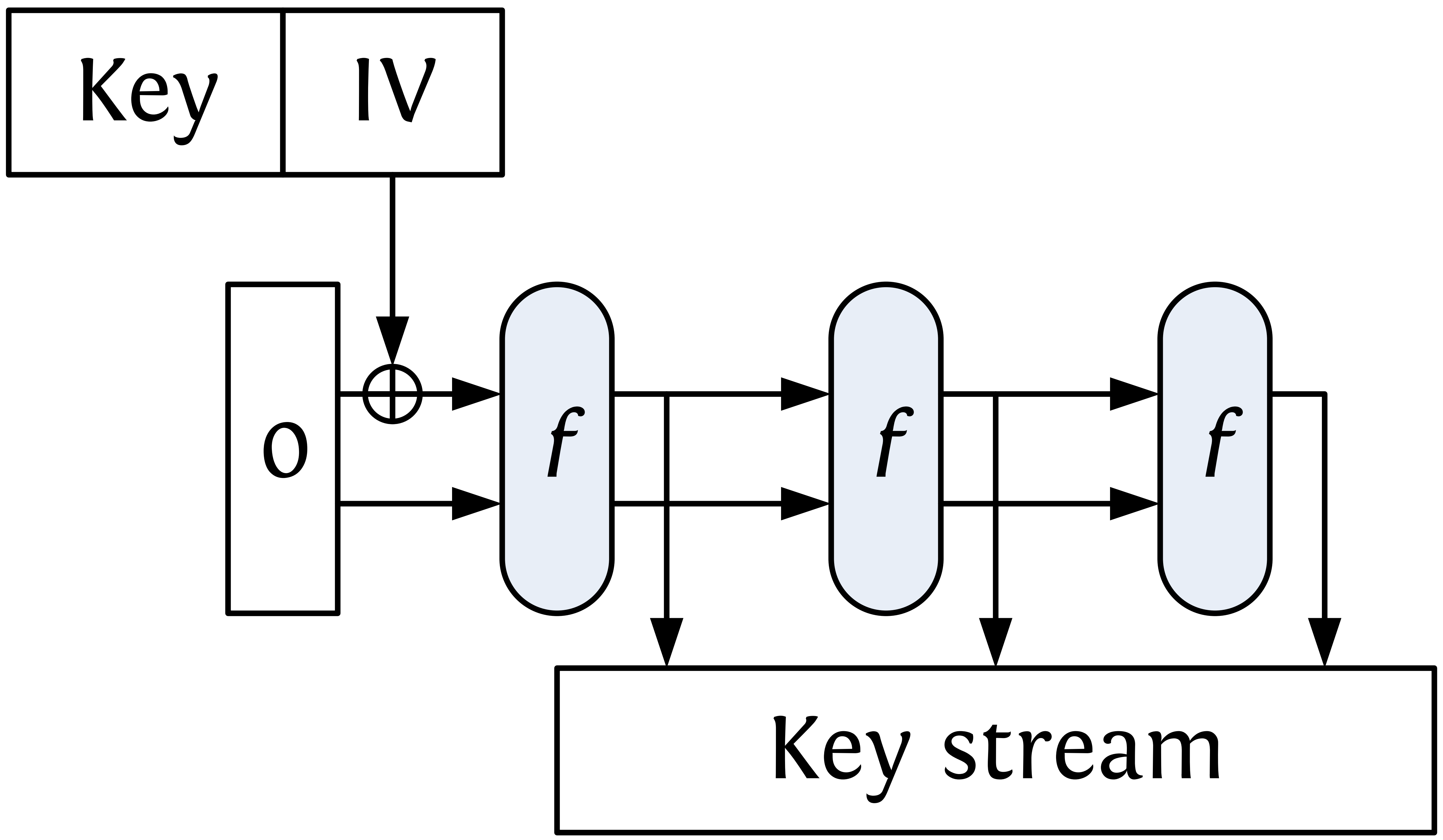
Message authentication codes



- As a message authentication code
- Simpler than HMAC [FIPS 198]
    - Required for SHA-1, SHA-2 due to length extension property
    - No longer needed for sponge

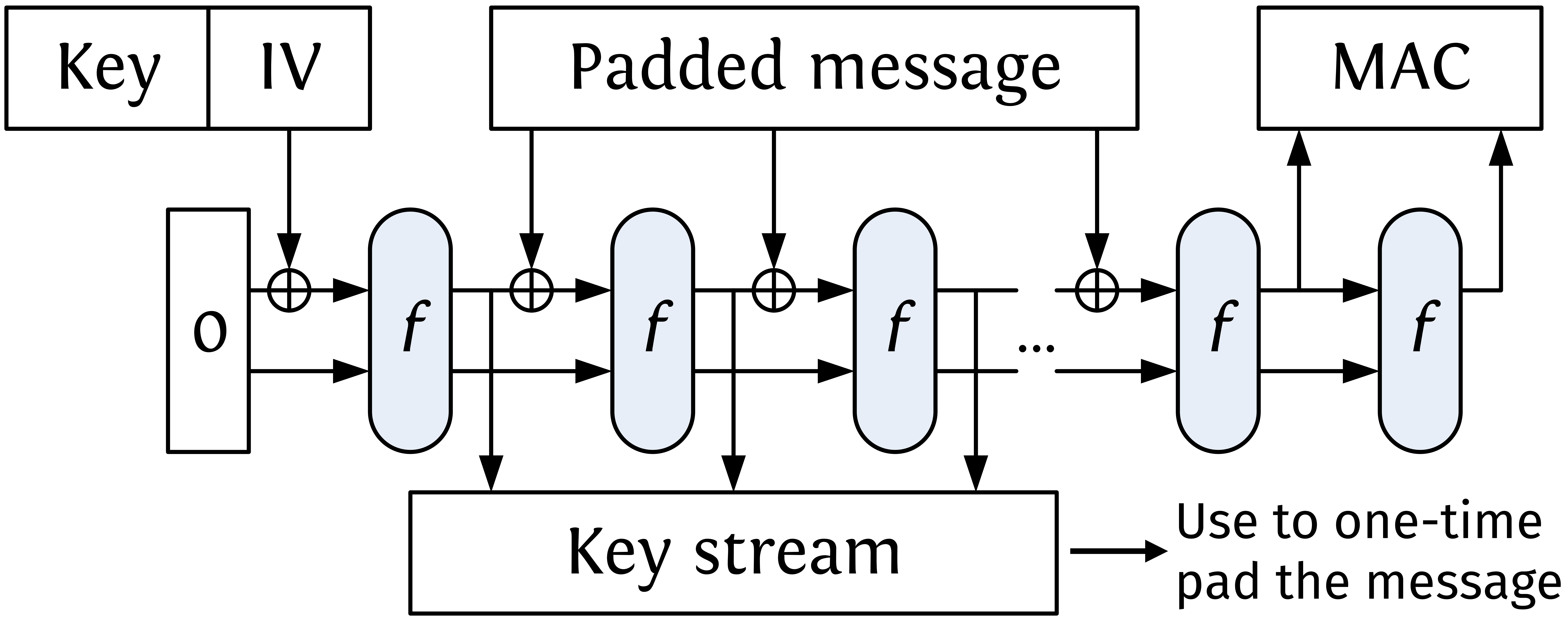
Slides from
Keccak team

Stream encryption



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

Single pass authenticated encryption

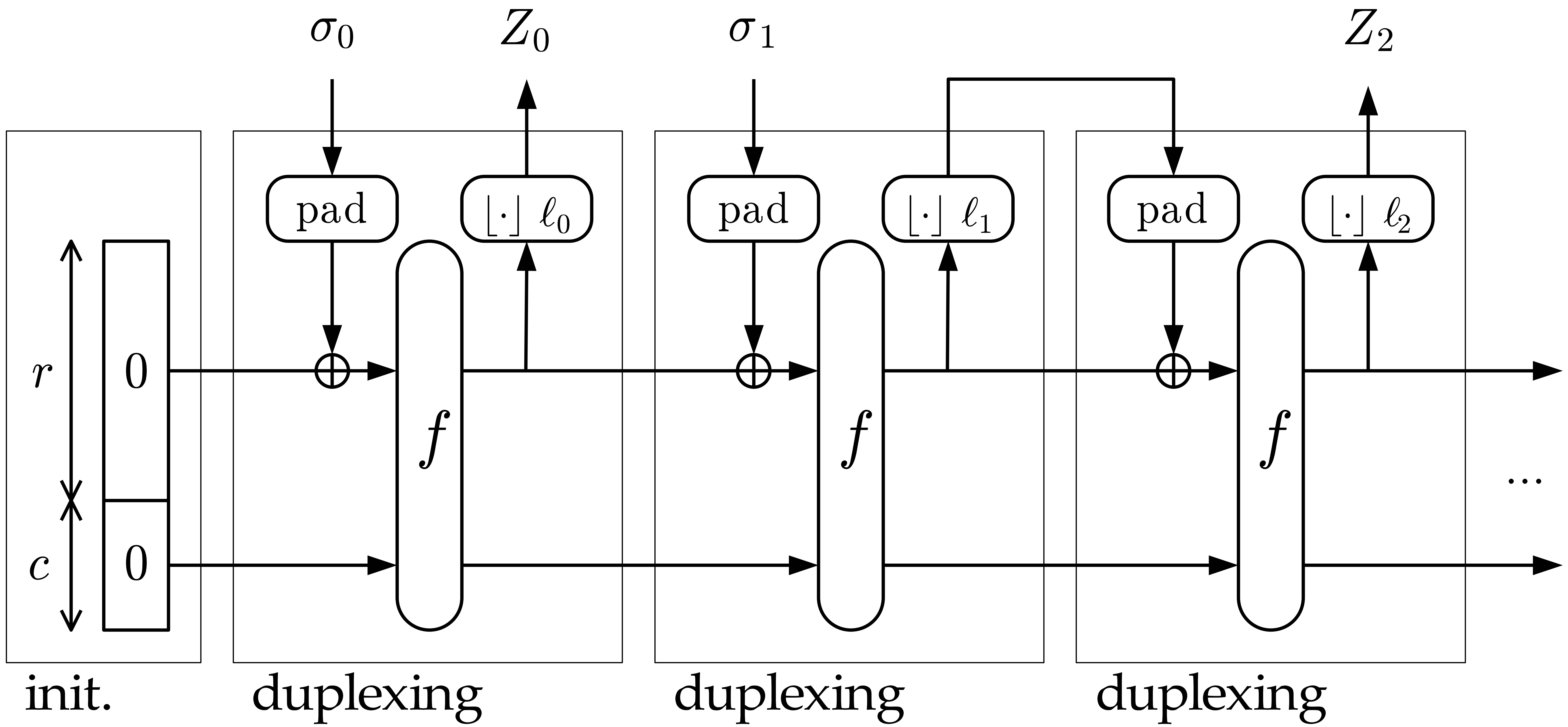


- Authentication and encryption in a single pass!
- Secure messaging (*SSL/TLS, SSH, IPSEC* ...)

Slides from
Keccak team

Reseedable pseudorandom sequence generator

- Defined in [Keccak Team, CHES 2010] and [Keccak Team, SAC 2011]
- Support for forward secrecy by *forgetting* in duplex:



Slides from Keccak team

Designing the permutation KECCAK-*f*

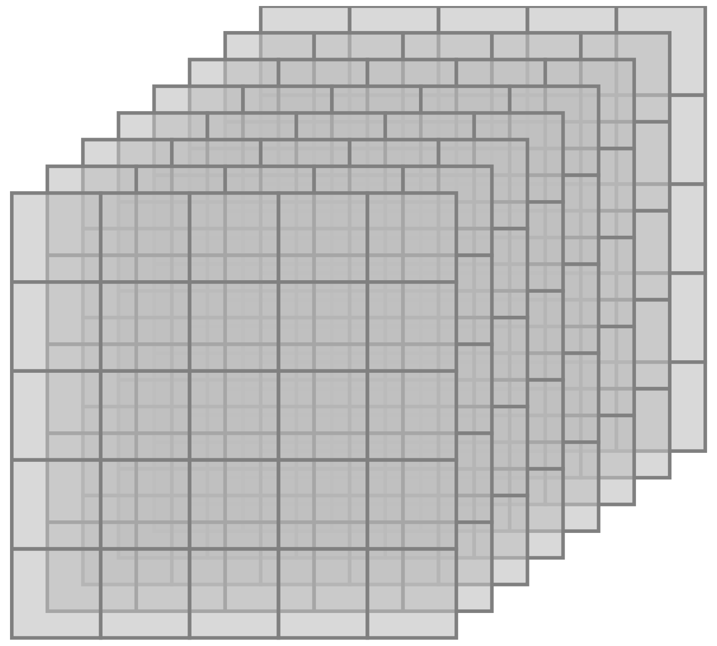
Our mission

To design a permutation called KECCAK-*f* that cannot be distinguished from a random permutation.

- Like a block cipher
  - sequence of identical rounds
  - round function that is nonlinear and has good diffusion
- …but not quite
  - no need for key schedule
  - round constants instead of round keys
  - inverse permutation need not be efficient

Slides from Keccak team

KECCAK

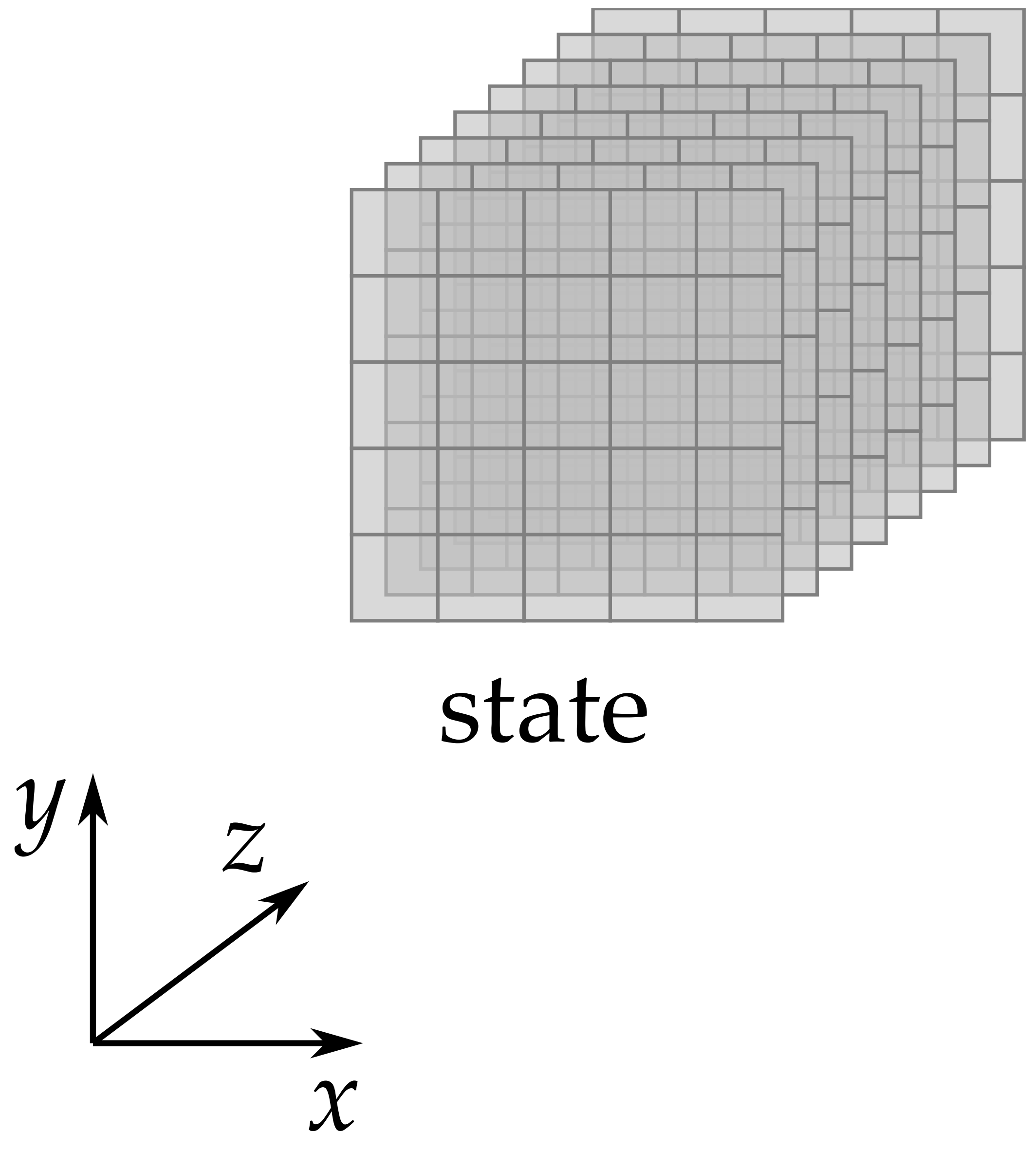


- Instantiation of a *sponge function*
- the permutation KECCAK-*f*
  - 7 permutations: $b \in \{25, 50, 100, 200, 400, 800, 1600\}$
- Security-speed trade-offs using the same permutation, e.g.,
  - SHA-3 instance: $r = 1088$ and $c = 512$
    - permutation width: 1600
    - security strength 256: post-quantum sufficient
  - Lightweight instance: $r = 40$ and $c = 160$
    - permutation width: 200
    - security strength 80: same as SHA-1

Slides from Keccak team

KECCAK-f: the permutations in KECCAK

Operates on 3D state:

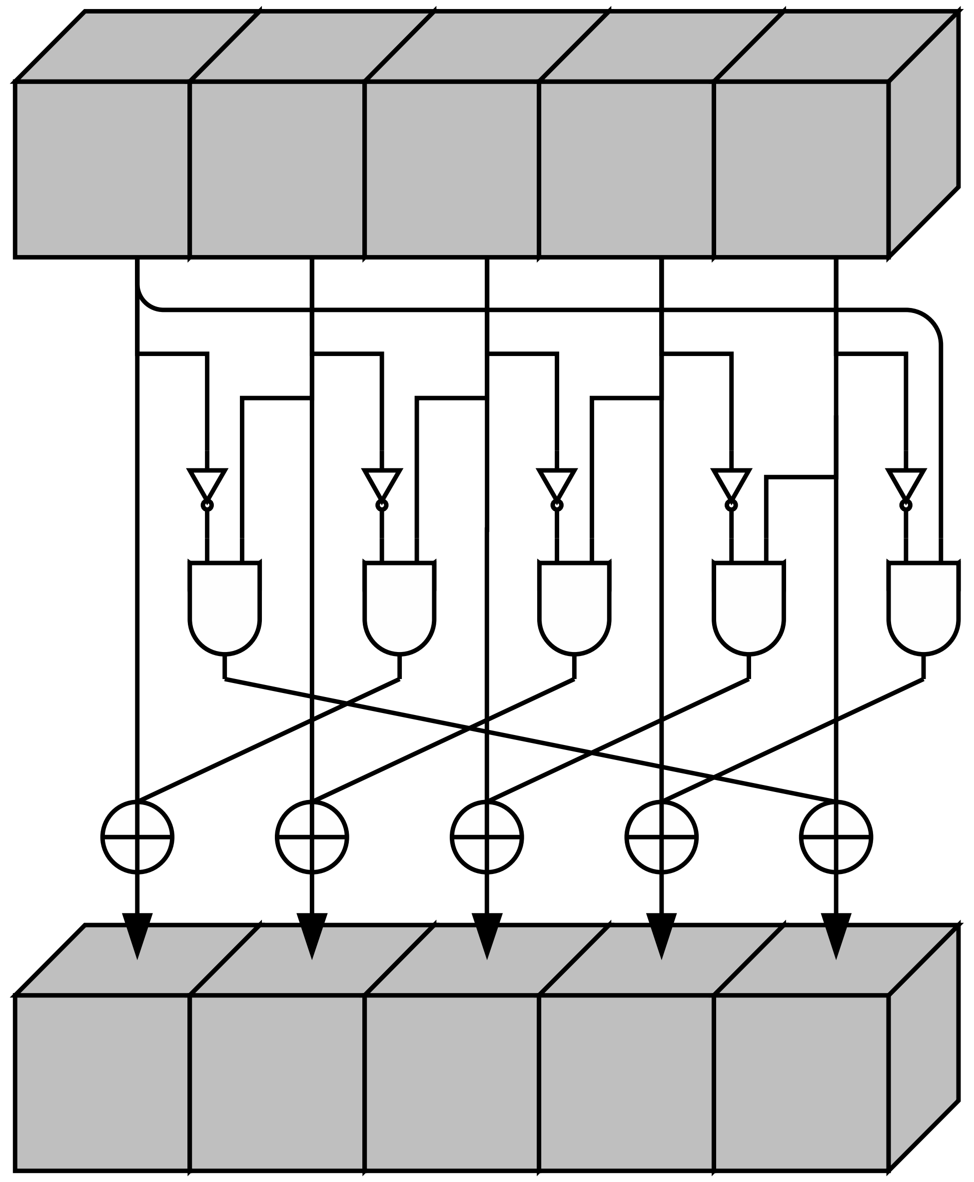


- (5×5)-bit slices
- 2^ℓ-bit lanes

- Round function R with 5 steps:
  - θ: mixing layer
  - ρ: bit transposition
  - π: bit transposition
  - χ: non-linear layer
  - ι: round constants
- # rounds: $12 + 2\ell$ for $b = 2^\ell 25$
  - 12 rounds in KECCAK-$f[25]$
  - 24 rounds in KECCAK-$f[1600]$

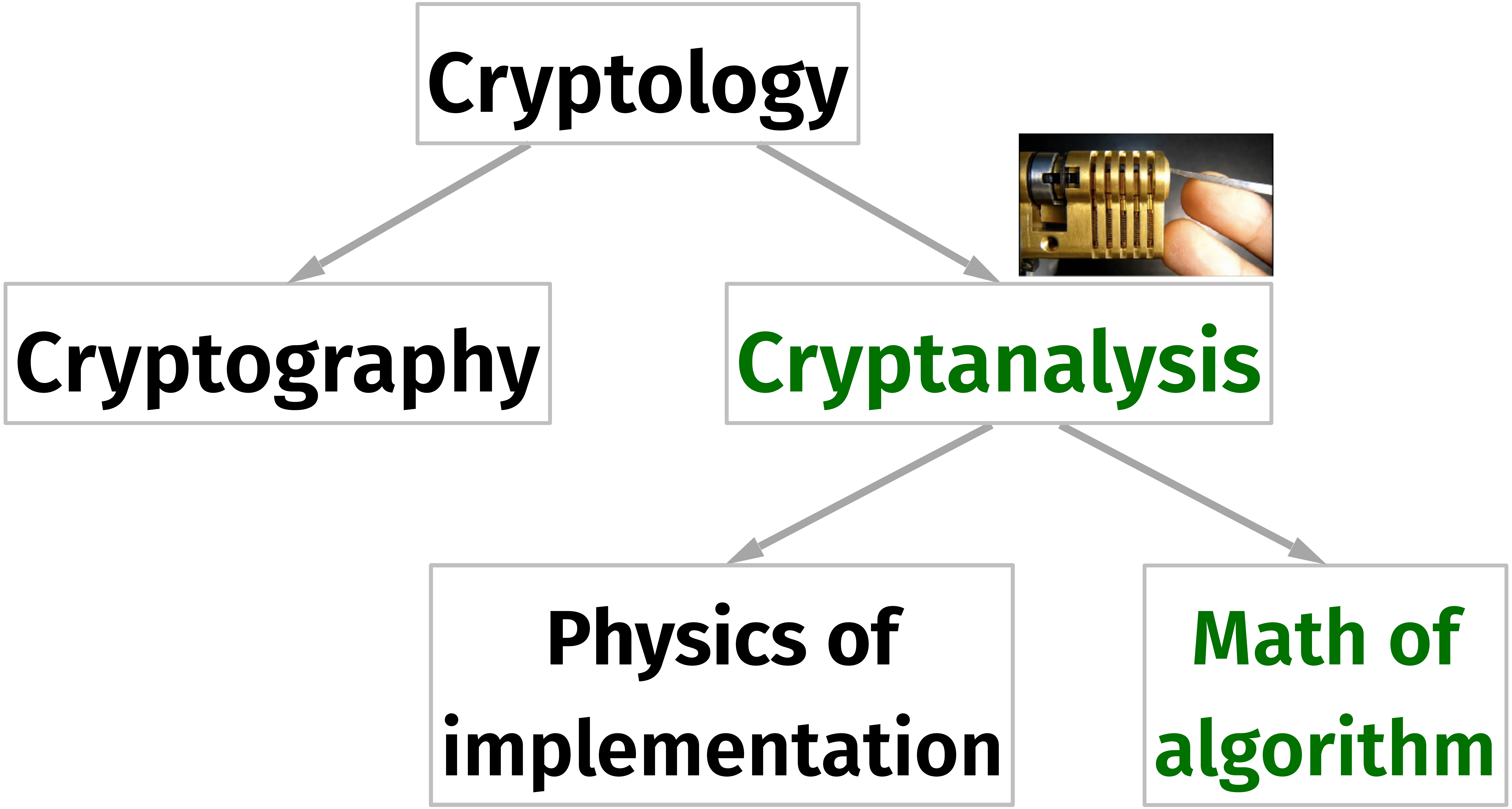
Slides from Keccak team

χ, the nonlinear mapping in KECCAK-*f*

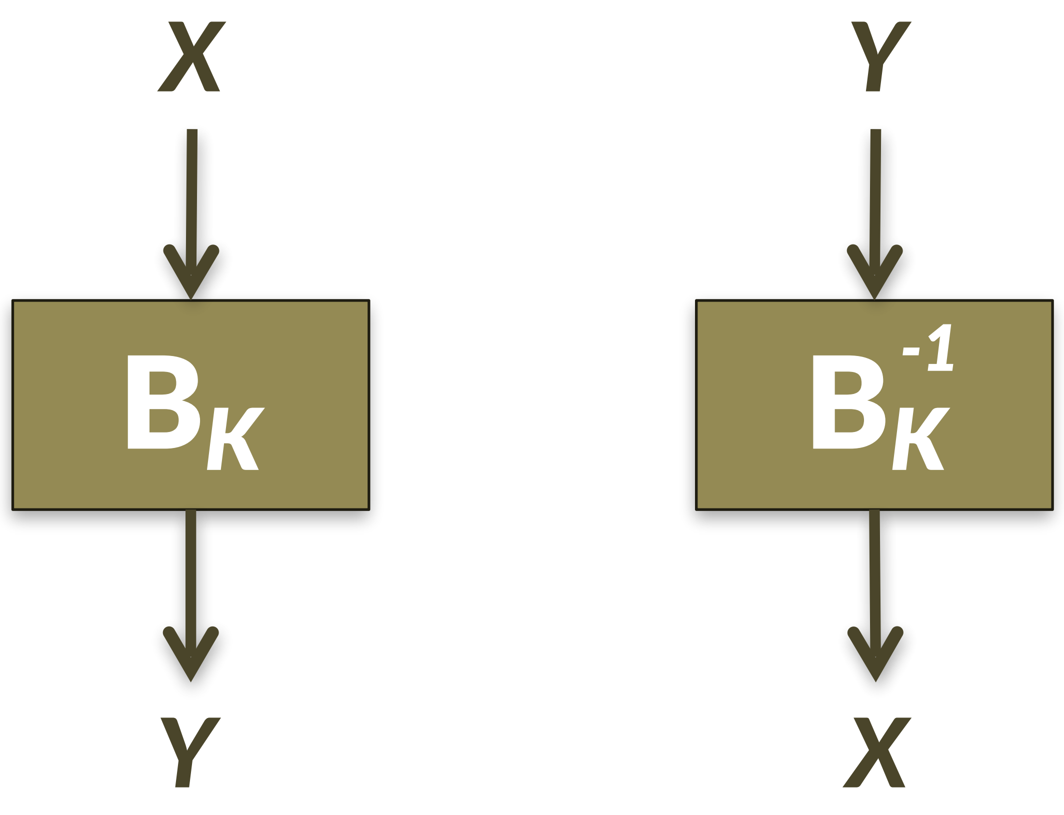


- "Flip bit if neighbors exhibit 01 pattern"
- Operates independently and in parallel on 5-bit rows
- Algebraic degree 2, inverse has degree 3
- LC/DC propagation properties easy to describe and analyze

Slides from
Keccak team


Cryptology
Cryptography
Cryptanalysis
Physics of implementation
Math of algorithm

Refresher: block ciphers



Design goals

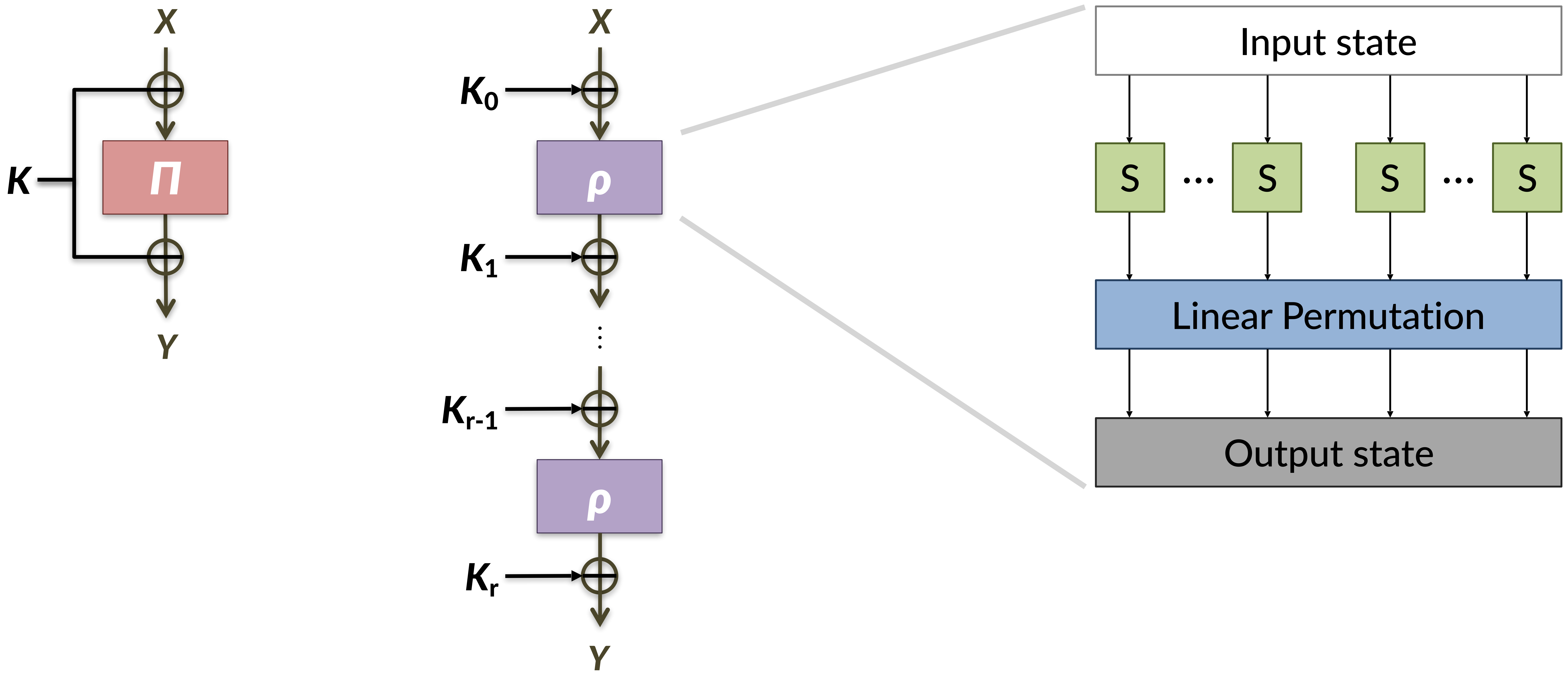
- Simple
- Makes no sense
- Simple to see why it makes no sense

Security goal: pseudorandomness

- B_K looks like a truly random function, aka Mallory cannot tell them apart
- Sanity check: what class of functions definitely isn't pseudorandom?

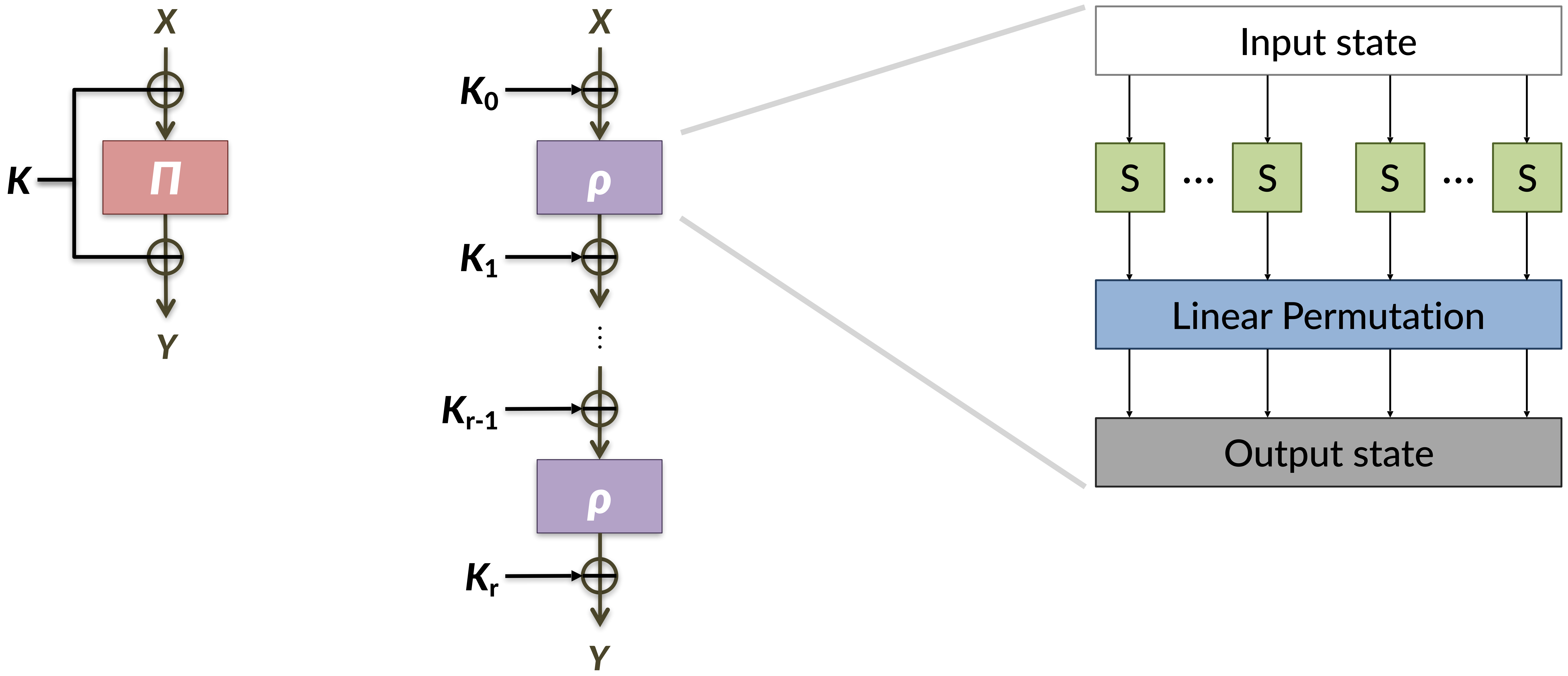
Refresher: block cipher design

Key alternation, over several rounds, each w/ substitution & permutation



Question: what if S is 'too linear'?

Key alternation, over several rounds, each w/ substitution & permutation



Question: what if S is 'too linear'?

<u>Form of the S-box</u>

1. A linear function on all N bits
2. Linear 'most of the time'
3. The 1st bit of output is a linear function of the 1st bit of input
4. Some subset of the output bits is linearly correlated with some subset of input bits
5. The difference in two S-box values is connected by a linear function

<u>How to break the cipher</u>

1. Solve a system of linear equations
2. Solve linear programming problem
3. Same as #1 (partial breaks count too)
4. Consider more correlations...
5. This is the derivative of the previous questions (in the calculus sense)

Refresher: Claude Shannon's 2 goals for block ciphers

Confusion

- Uncertain K → can't correlate X, Y
- Ideal: Prob[correlation] so small that attacker prefers a brute force attack

Diffusion

- 1 bit ΔX → huge ΔY
- Ideal: each output bit depends on all input bits (2 rounds in AES)

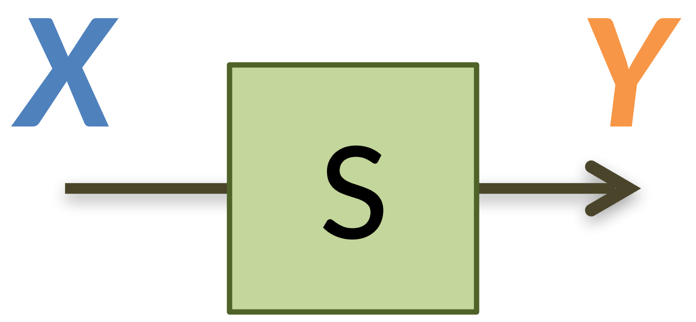
Question: what if S is ‘too linear’?

Confusion

- Uncertain K → can’t correlate X, Y
- Ideal: Prob[correlation] so small that attacker prefers a brute force attack

Linear cryptanalysis

Exploits the fact that S may behave ‘similarly’ to a linear function

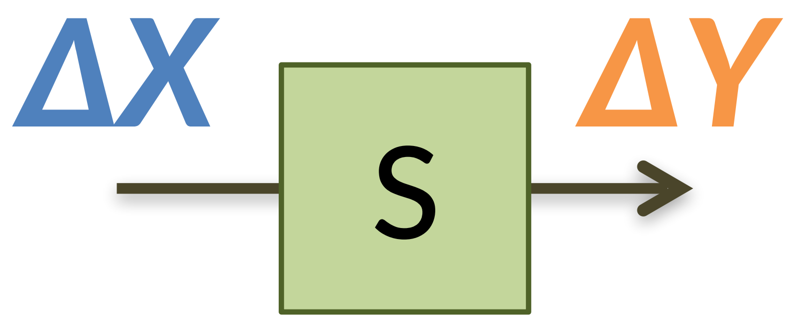


Diffusion

- 1 bit ΔX → huge ΔY
- Ideal: each output bit depends on all input bits (2 rounds in AES)

Differential cryptanalysis

Exploits the fact that *differences* in inputs + outputs may be correlated



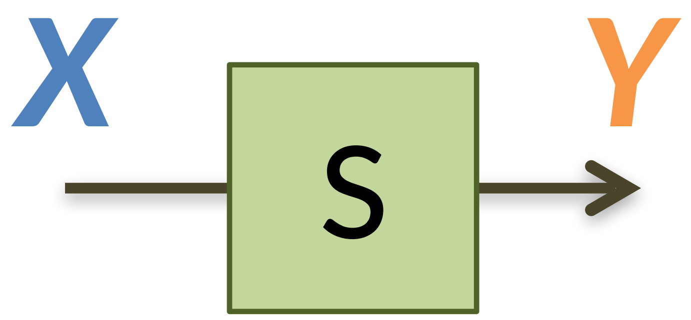
Question: what if S is 'too linear'?

Confusion

- Uncertain K → can't correlate X, Y
- Ideal: Prob[correlation] so small that attacker prefers a brute force attack

Linear cryptanalysis

Exploits the fact that S may behave 'similarly' to a linear function

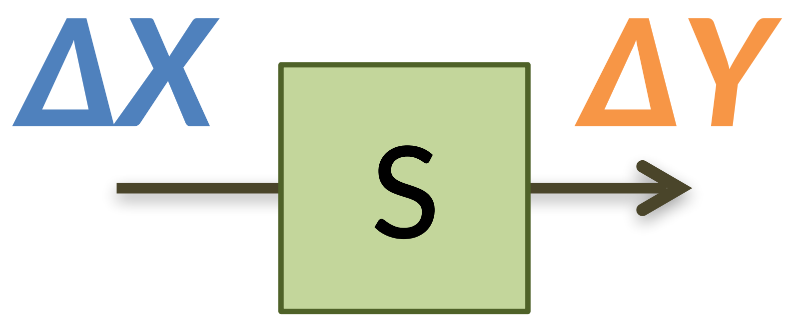


Diffusion

- 1 bit ΔX → huge ΔY
- Ideal: each output bit depends on all input bits (2 rounds in AES)

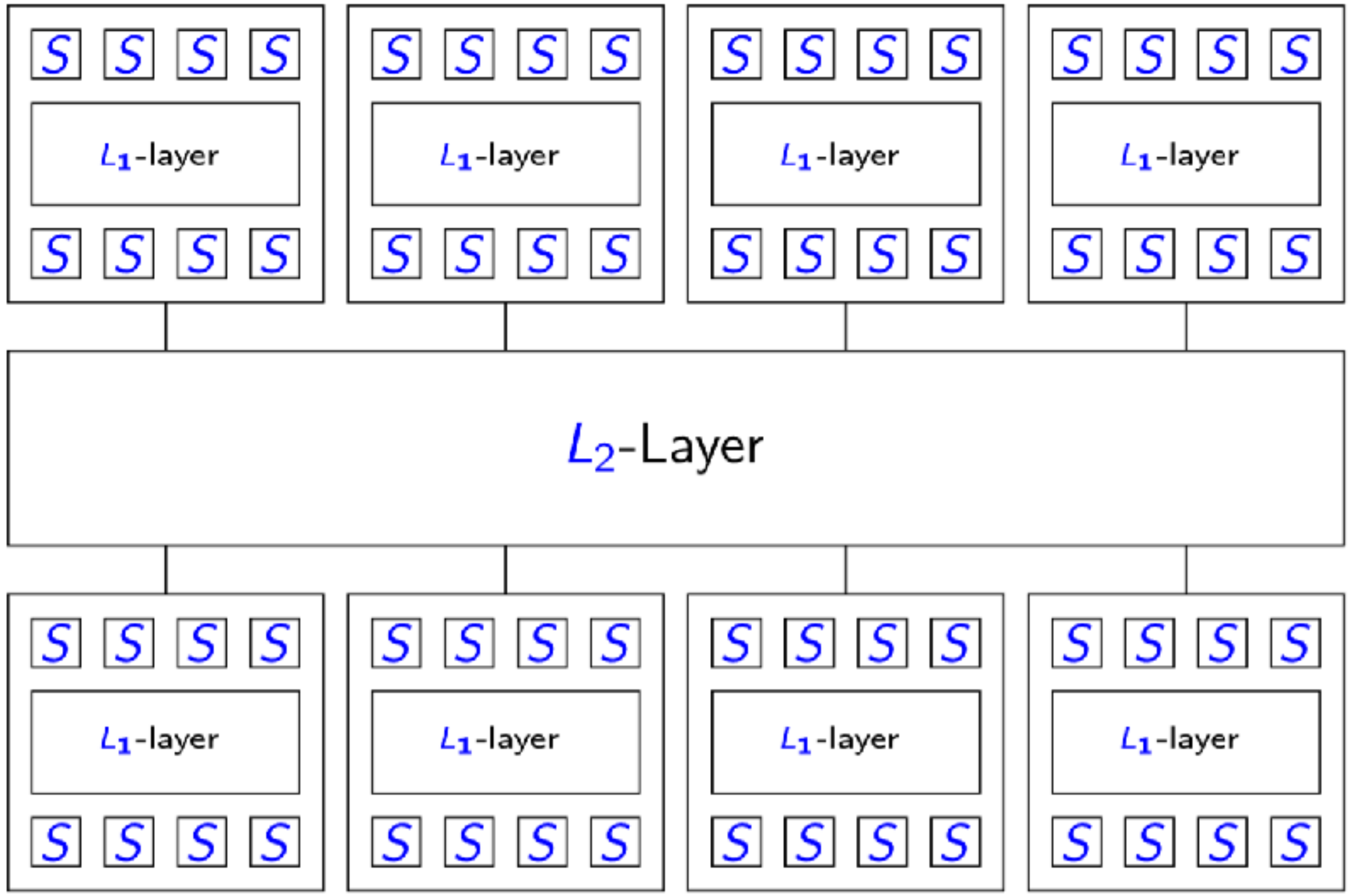
Differential cryptanalysis *(our focus)*

Exploits the fact that *differences* in inputs + outputs may be correlated



Cryptanalysis of AES: Wide trail strategy through 4 rounds

- Picture depicts 4 rounds of AES
  - ≥ 25 active S-boxes in 4 rounds
  - Each has max diff propagation of 2^{-6}
- So Pr [four-round trail] ≈ 2^{-150}
  - An 8-round trail has C < 2^{-300}
  - A 12-round trail has C < 2^{-450}
- Brute force search is better



> "Instead of spending most of its resources on large S-boxes, the wide trail strategy aims at designing the round transformations such that there are no [linear or differential] trails/characteristics of low weight"

Bounds for differential trails in KECCAK-$f[1600]$

Rounds	Lower bound		Best known	
1	2		2	
2	8		8	
3	32	[KECCAK team]	32	[Duc et al.]
4			134	[KECCAK team]
5			510	[Naya-Plasencia et al.]
6	74	[KECCAK team]	1360	[KECCAK team]
24	296		???	

Slides from
Keccak team