Keccak

From:

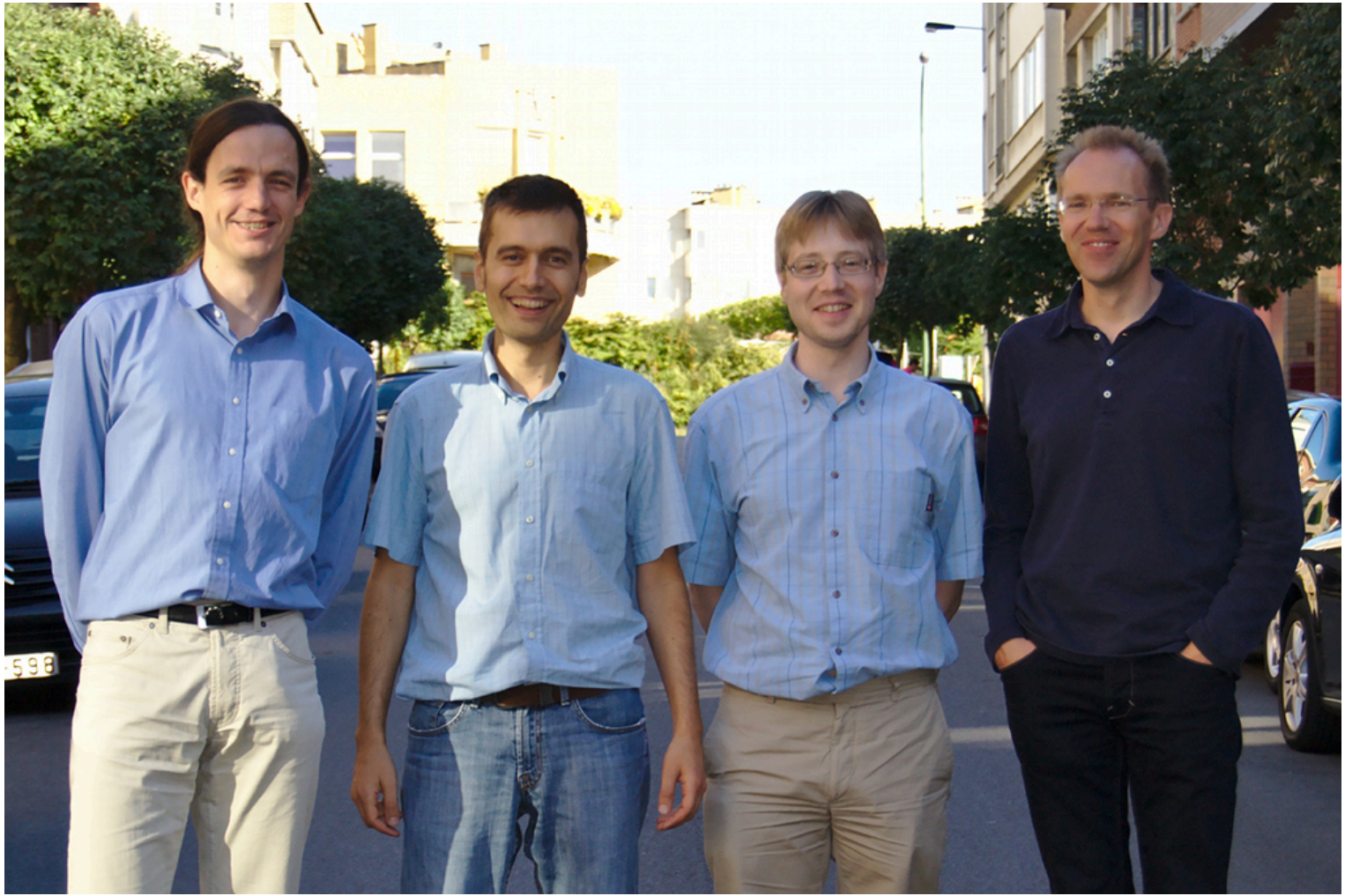
1.The Keccak reference

2. Keccak and the SHA-3 Standardization

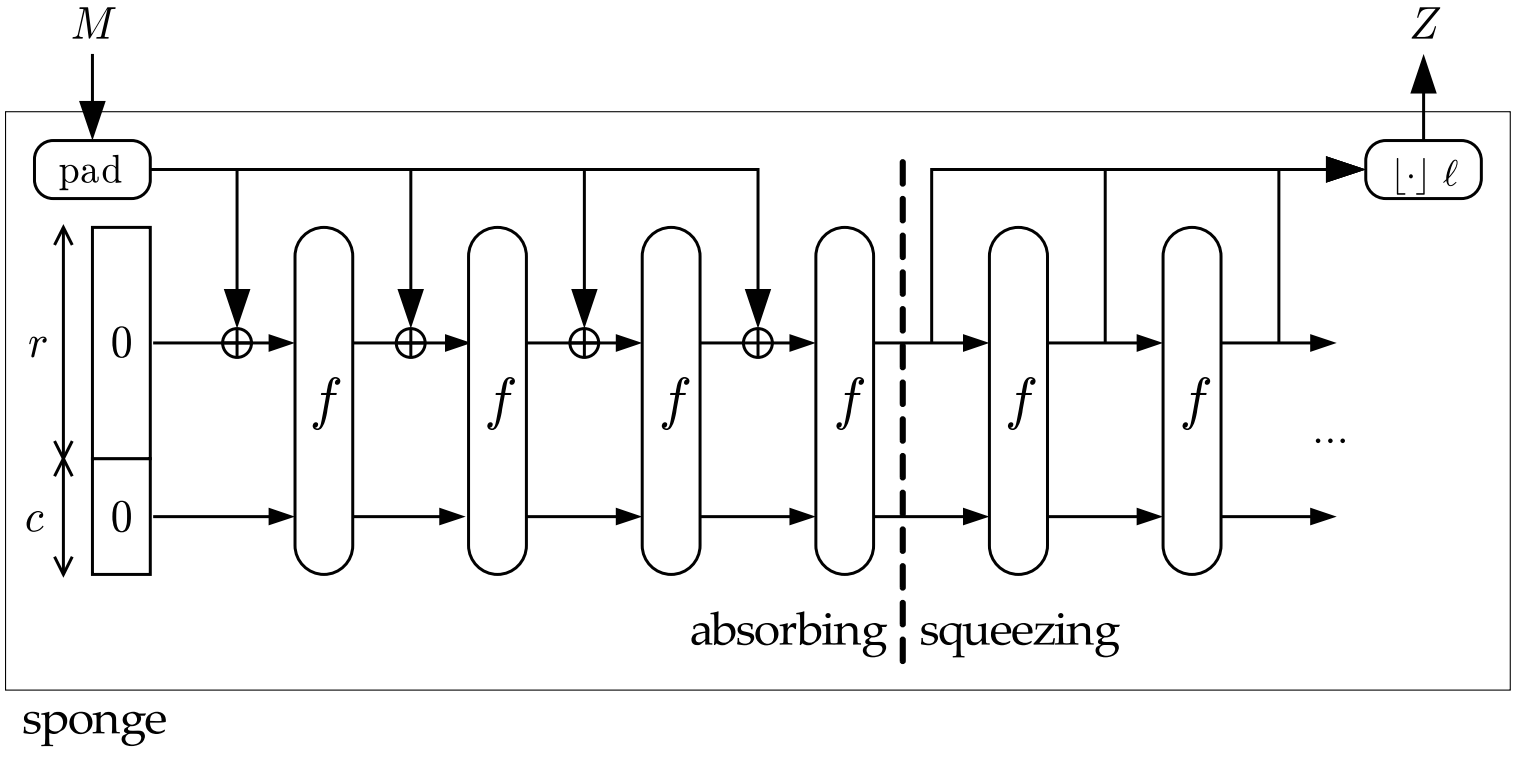
written by

Guido Bertoni (STMicroelectronics),
Joan Daemen (STMicroelectronics),
Michaël Peeters (NXP Semiconductors),
Gilles Van Assche (STMicroelectronics)

see http://keccak.noekeon.org/



The sponge construction

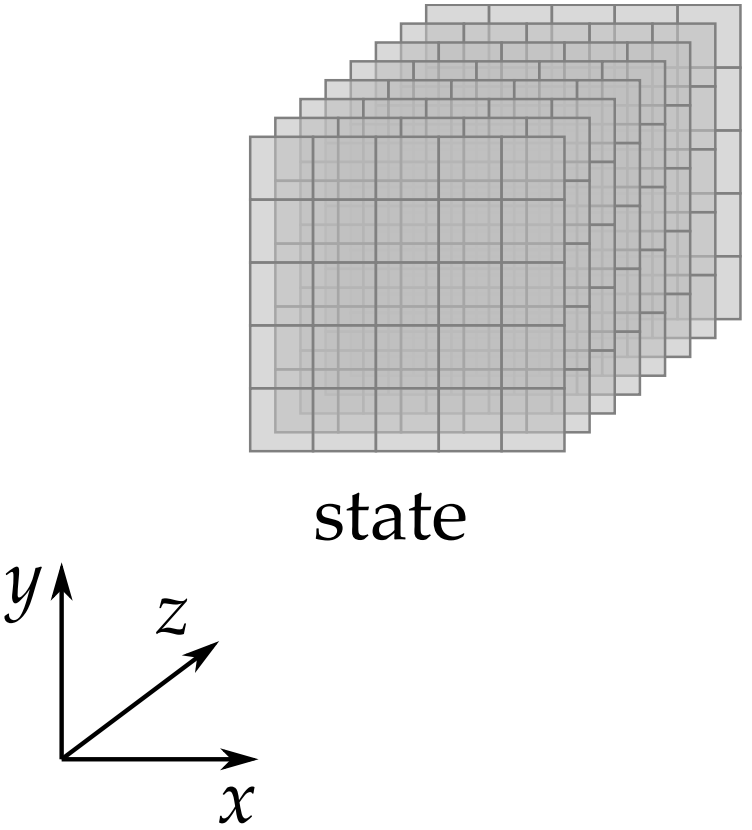


- More general than a hash function: arbitrary-length output
- Calls a b-bit permutation f, with $b = r + c$
  - r bits of *rate*
  - c bits of *capacity* (security parameter)

Keccak

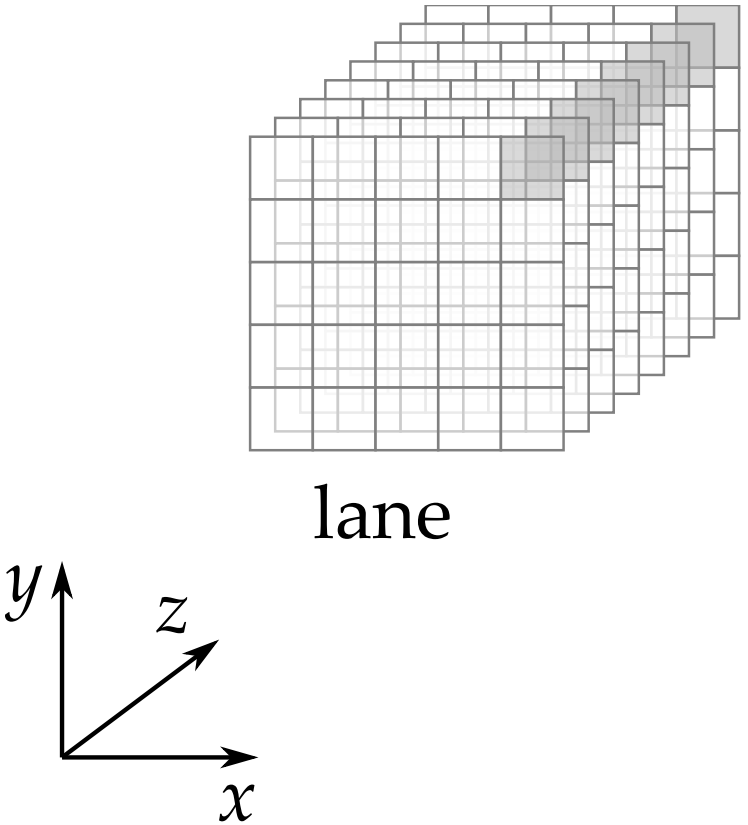
- Instantiation of a *sponge function*
- the permutation Keccak-f
  - 7 permutations: $b \in \{25, 50, 100, 200, 400, 800, 1600\}$
- Security-speed trade-offs using the same permutation, e.g.,
  - SHA-3 instance: $r = 1088$ and $c = 512$
    - permutation width: 1600
    - security strength 256: post-quantum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

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



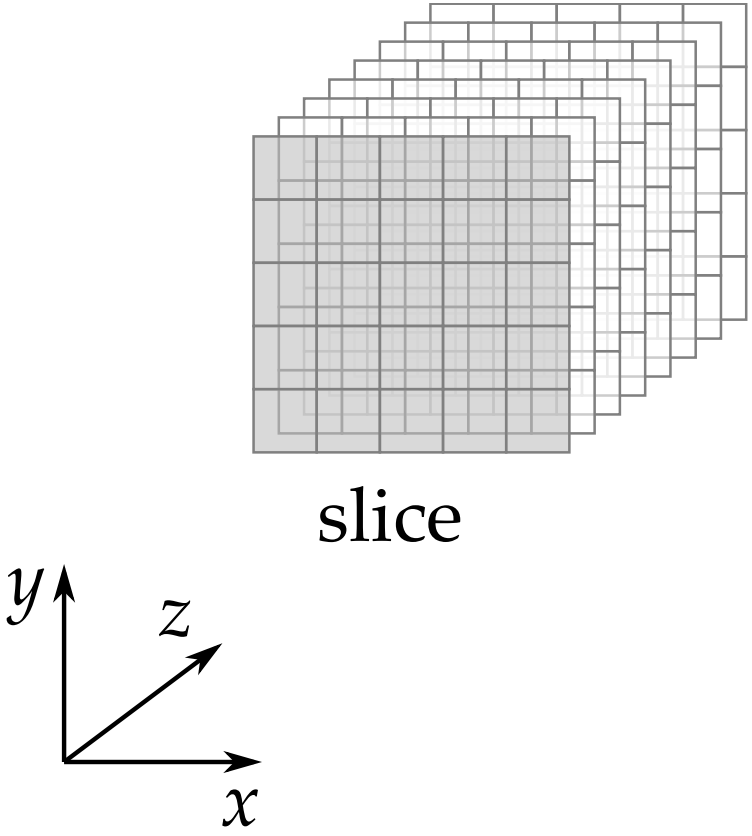
- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)

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



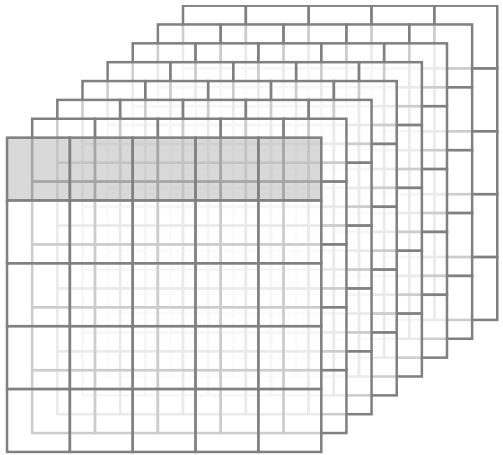
- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)

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

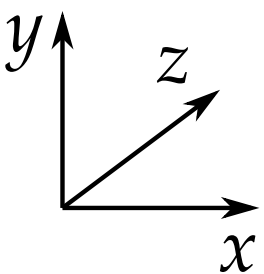


- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them

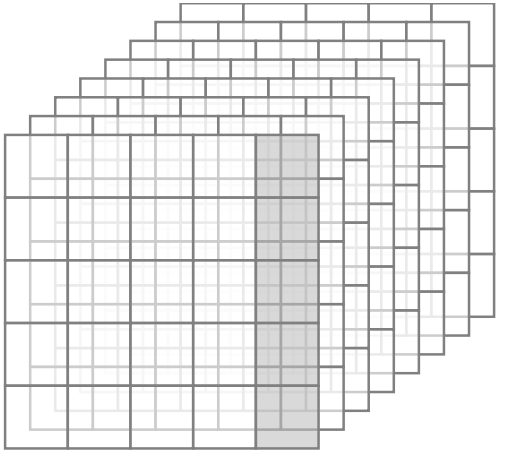
The state: an array of $5 \times 5 \times 2^{\ell}$ bits



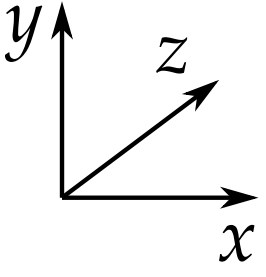
row

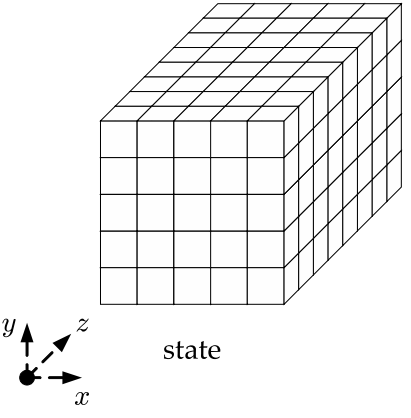


The state: an array of $5 \times 5 \times 2^{\ell}$ bits

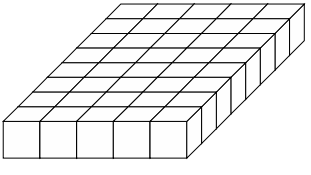


column

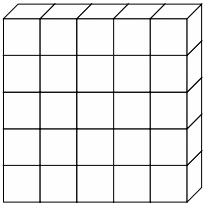




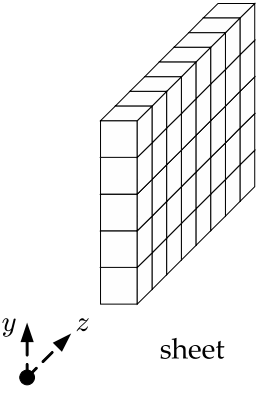
state



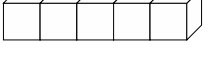
plane



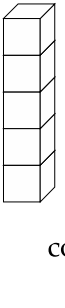
slice



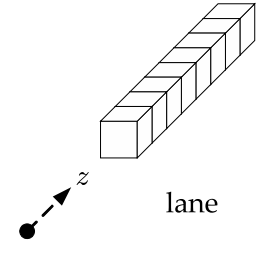
sheet



row



column



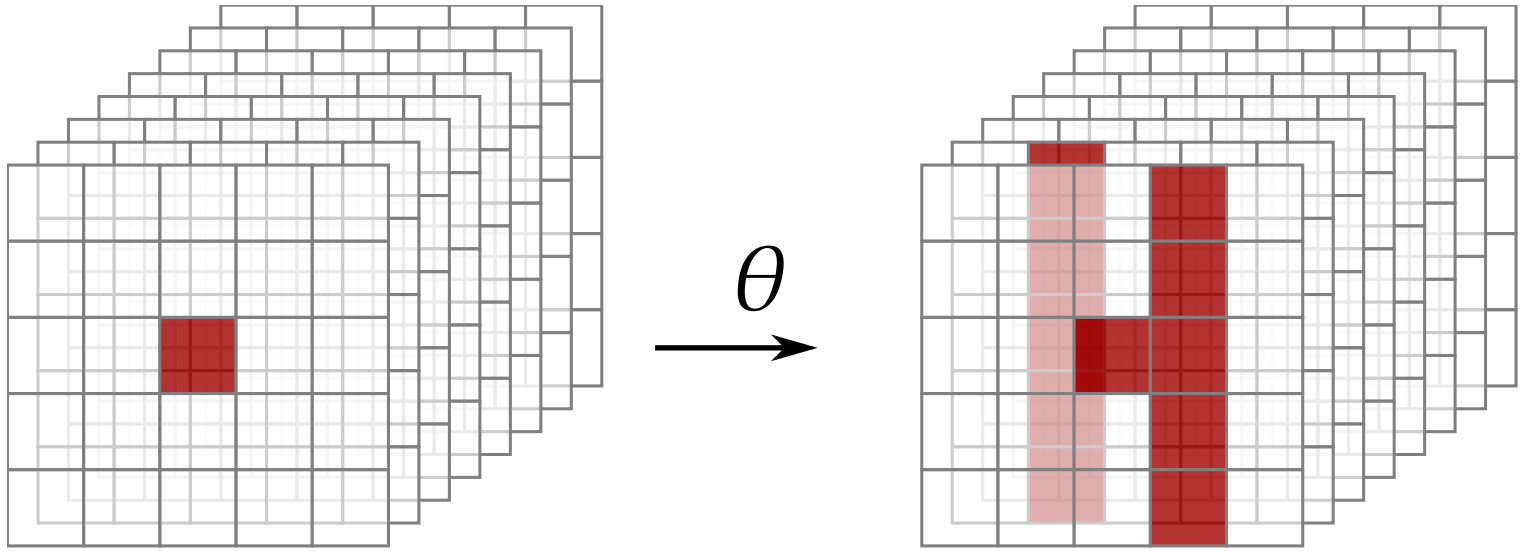
lane

bit

KECCAK-$f[b]$ is an iterated permutation, consisting of a sequence of n_r rounds R, indexed with i_r from 0 to $n_\mathrm{r} - 1$. A round consists of five steps:

$$\mathrm{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta, \text{ with}$$

$$\begin{aligned}
\theta: \quad & a[x][y][z] \leftarrow a[x][y][z] + \sum_{y'=0}^{4} a[x-1][y'][z] + \sum_{y'=0}^{4} a[x+1][y'][z-1], \\
\rho: \quad & a[x][y][z] \leftarrow a[x][y][z-(t+1)(t+2)/2], \\
& \quad \text{with } t \text{ satisfying } 0 \le t < 24 \text{ and } \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}^t \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \text{ in } \mathrm{GF}(5)^{2\times 2}, \\
& \quad \text{or } t = -1 \text{ if } x = y = 0, \\
\pi: \quad & a[x][y] \leftarrow a[x'][y'], \text{ with } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}, \\
\chi: \quad & a[x] \leftarrow a[x] + (a[x+1] + 1)a[x+2], \\
\iota: \quad & a \leftarrow a + \mathrm{RC}[i_\mathrm{r}].
\end{aligned}$$

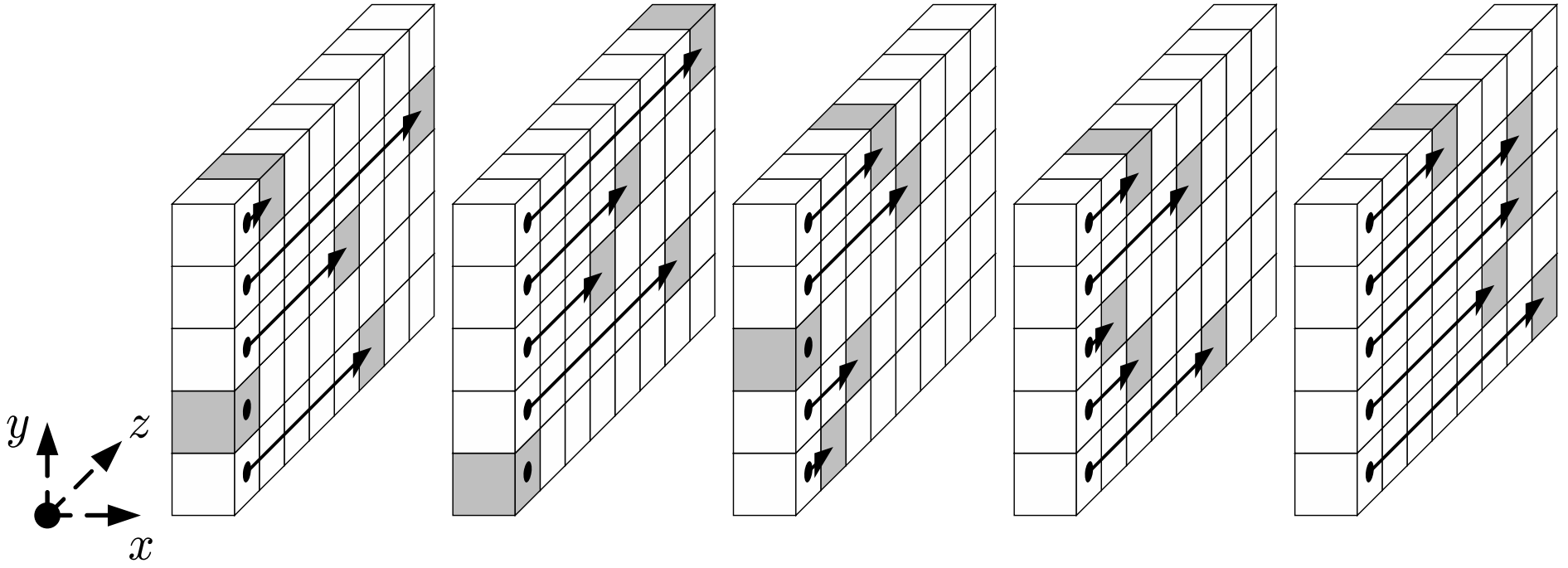

θ

ρ for inter-slice dispersion

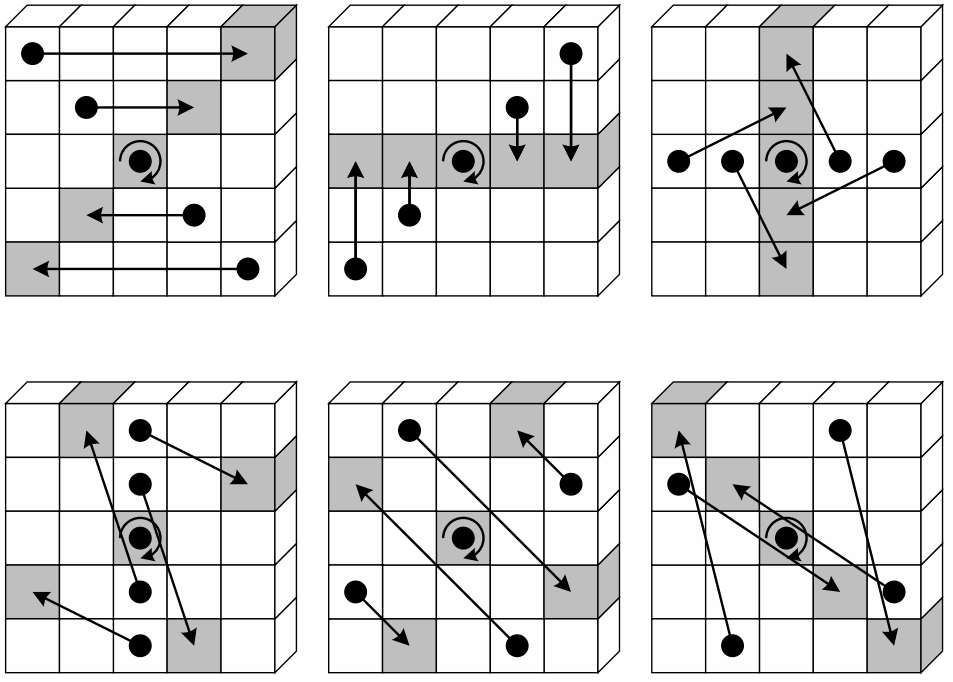
- We need diffusion between the slices ...
- ρ: cyclic shifts of lanes with offsets

$$i(i+1)/2 \bmod 2^\ell$$

- Offsets cycle through all values below 2^ℓ

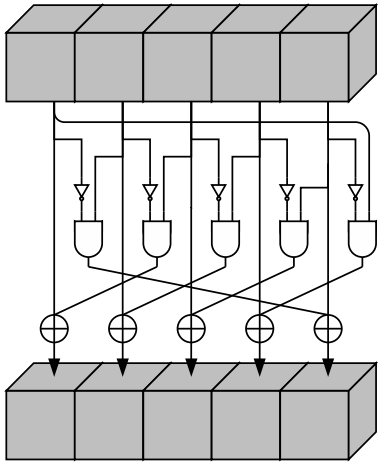


π for disturbing horizontal/vertical alignment



$$a_{x,y} \leftarrow a_{x',y'} \text{ with } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

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

ι to break symmetry

- XOR of round-dependent constant to lane in origin
- Without ι, the round mapping would be symmetric
  - invariant to translation in the z-direction
- Without ι, all rounds would be the same
  - susceptibility to *slide* attacks
  - defective cycle structure
- Without ι, we get simple fixed points (000 and 111)

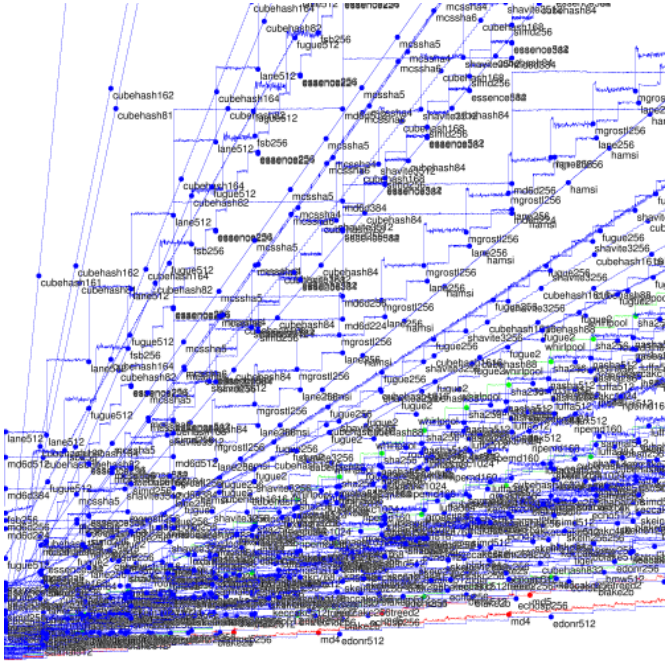
Keccak-f summary

- Round function:

$$\mathsf{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

- Number of rounds: $12 + 2\ell$
  - Keccak-$f[25]$ has 12 rounds
  - Keccak-$f[1600]$ has 24 rounds
- Efficiency
  - high level of parallellism
  - flexibility: bit-interleaving
  - software: competitive on wide range of CPU
  - dedicated hardware: very competitive
  - suited for protection against side-channel attack

Performance in software



- Faster than SHA-2 on all modern PC
- KeccakTree faster than MD5 on some platforms

C/b	Algo	Strength
4.79	keccakc256treed2	128
4.98	md5	< 64
5.89	keccakc512treed2	256
6.09	sha1	< 80
8.25	keccakc256	128
10.02	keccakc512	256
13.73	sha512	256
21.66	sha256	128

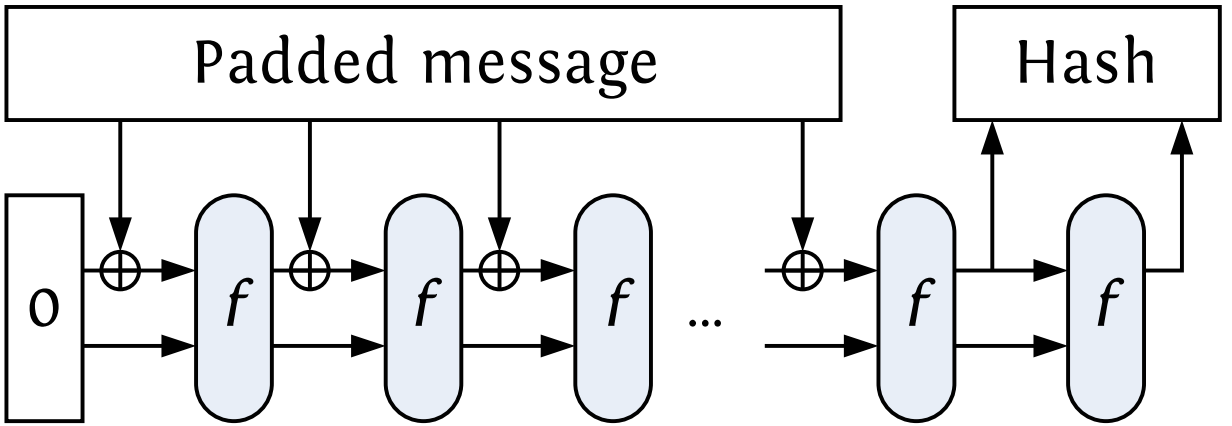
[eBASH, hydra6, http://bench.cr.yp.to/]

KECCAK-*f* in pseudo-code

```
KECCAK-F[b](A) {
  forall i in 0…n_r-1
    A = Round[b](A, RC[i])
  return A
}

Round[b](A,RC) {
  θ step
  C[x] = A[x,0] xor A[x,1] xor A[x,2] xor A[x,3] xor A[x,4], forall x in 0…4
  D[x] = C[x-1] xor rot(C[x+1],1),                          forall x in 0…4
  A[x,y] = A[x,y] xor D[x],                                 forall (x,y) in (0…4,0…4)

  ρ and π steps
  B[y,2*x+3*y] = rot(A[x,y], r[x,y]),                       forall (x,y) in (0…4,0…4)

  χ step
  A[x,y] = B[x,y] xor ((not B[x+1,y]) and B[x+2,y]),        forall (x,y) in (0…4,0…4)

  ι step
  A[0,0] = A[0,0] xor RC

  return A
}
```

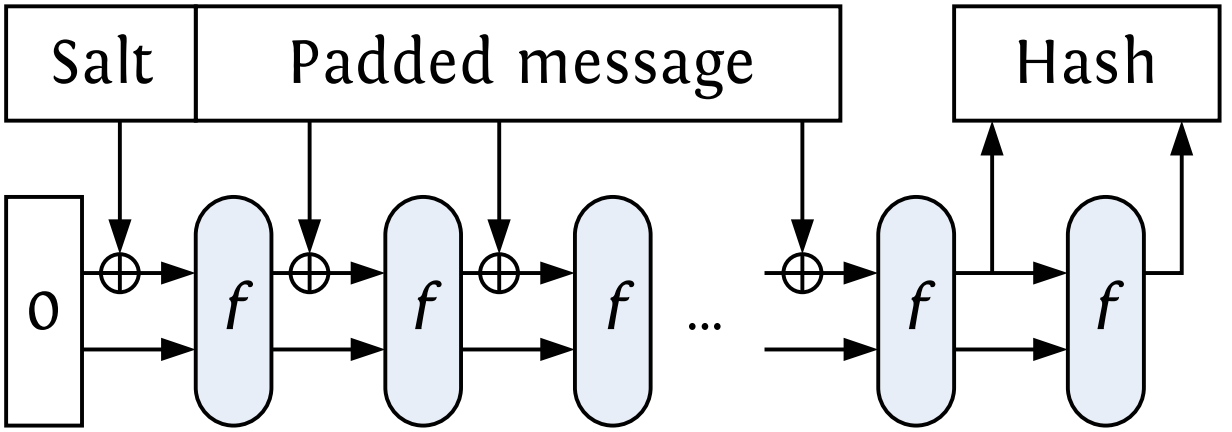
http://keccak.noekeon.org/specs_summary.html

Regular hashing



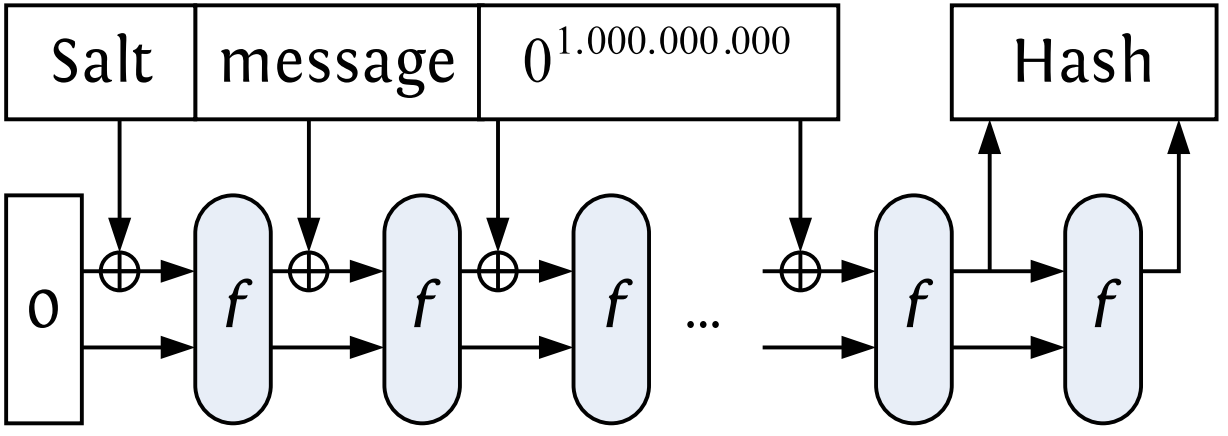
- Electronic signatures
- Data integrity (*shaXsum* ...)
- Data identifier (*Git, online anti-virus, peer-2-peer* ...)

Salted hashing



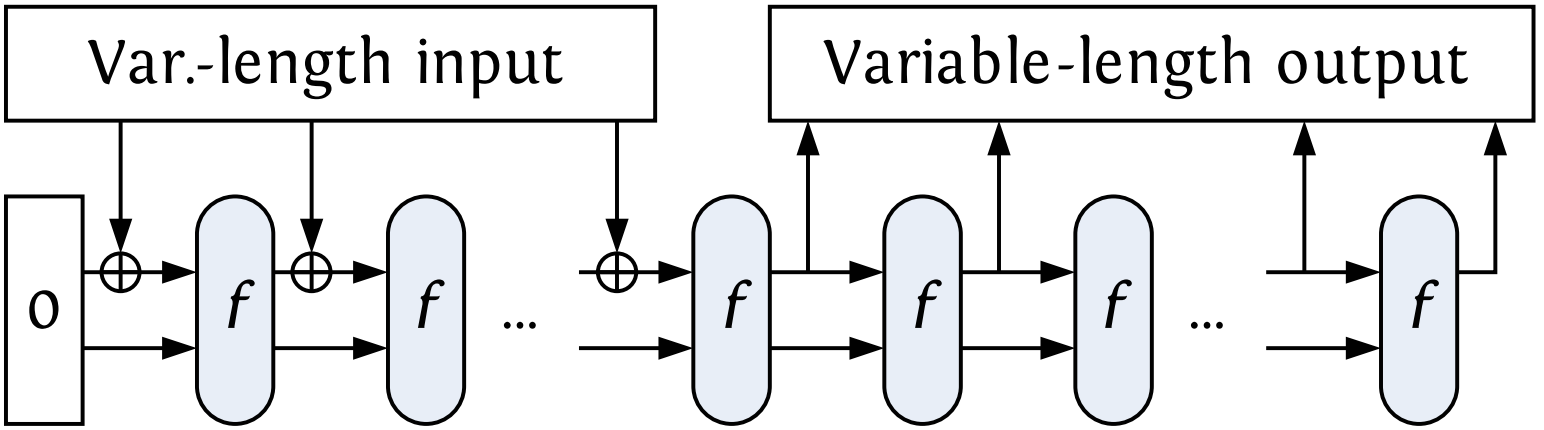
- Randomized hashing (RSASSA-PSS)
- Password storage and verification (*Kerberos*, /etc/shadow)

Salted hashing



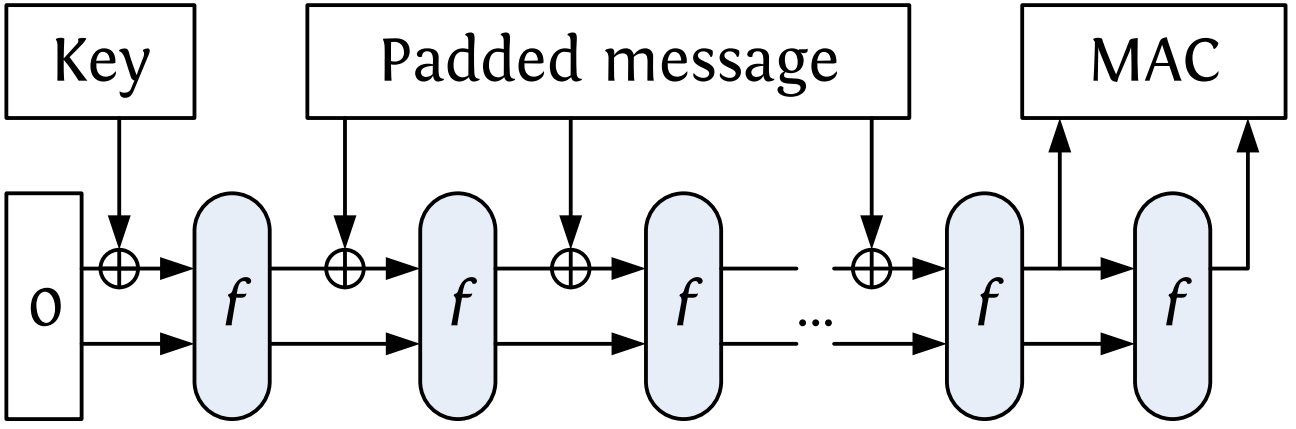
- Randomized hashing (RSASSA-PSS)
- Password storage and verification (*Kerberos*, /etc/shadow)
  - ...Can be as slow as you like it!

Mask generation function



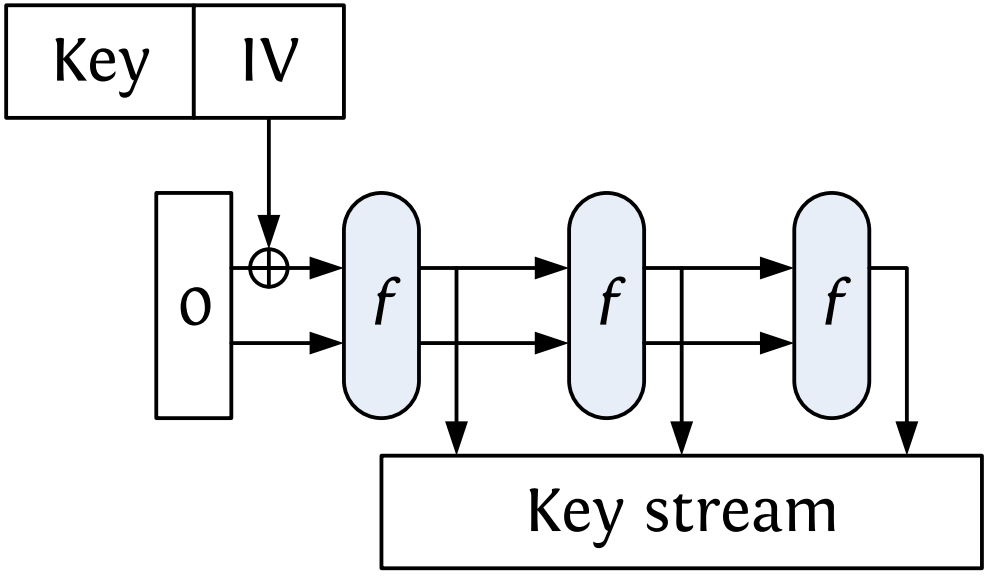
- Key derivation function in SSL, TLS
- Full-domain hashing in public key cryptography
  - electronic signatures RSASSA-PSS [PKCS#1]
  - encryption RSAES-OAEP [PKCS#1]
  - key encapsulation methods (KEM)

Message authentication codes



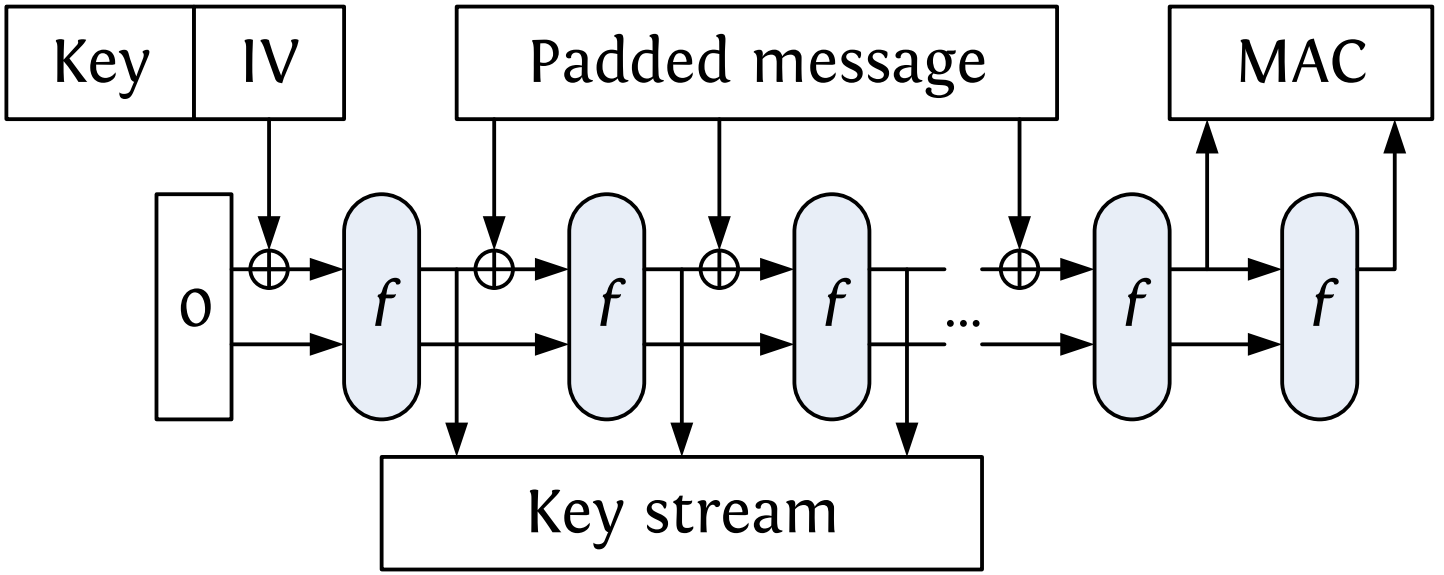
- As a message authentication code
- Simpler than HMAC [FIPS 198]
  - Required for SHA-1, SHA-2 due to length extension property
  - No longer needed for sponge

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* ...)