A Narrow Trail in Hash Function Design

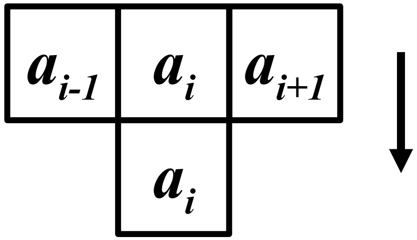
Joan Daemen

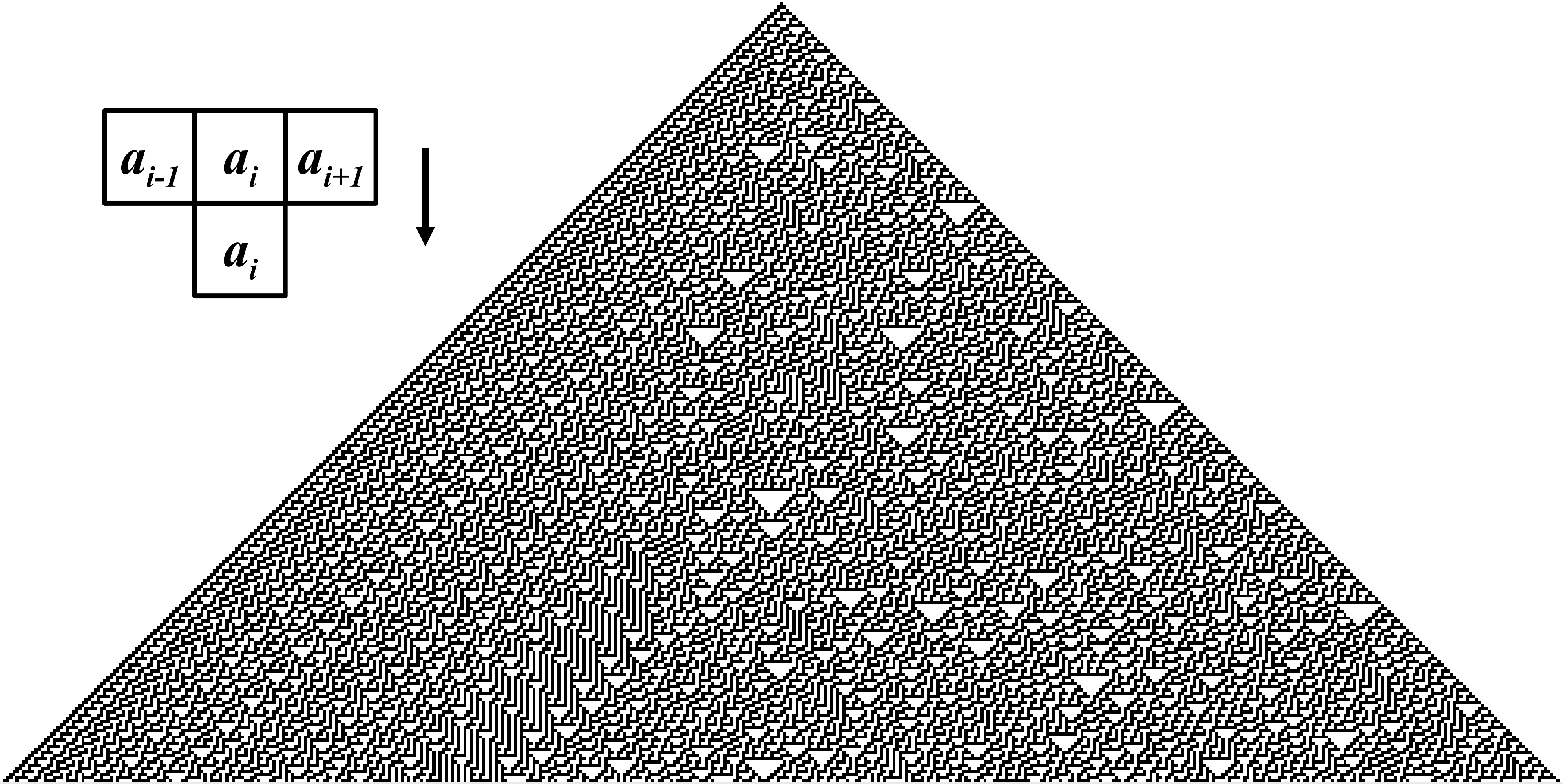
STMicroelectronics

Cellular Automata in cryptography

- CA guru Stephen Wolfram at Crypto ’85:
  - Stream cipher: Pseudorandom generation with CA
- Crypto guru Ivan Damgård at Crypto ’89
  - Hash function: Compression function with CA

CA Rule 30: $a_i = \overline{a}_{i-1} \oplus (a_i a_{i+1} == 00)$





CA Rule 45 (γ): $a_i = \overline{a}_{i-1} \oplus (a_i a_{i+1} == 01)$

- ’89-’90: cycle length hot issue in stream ciphers
- First experiment: determine cycle distributions
  - all rules with 3-neighborhood
  - lengths between 11 and 32
- Observation:
  - rule 45, odd length: cycles as if random permutation
  - invertible!
- Stream cipher idea:
  - take odd length and ...
  - use γ instead of 30

Other CA rules

- ’91: breakdown of CA crypto:
  - Eurocrypt ’91: Meier-Staffelbach break Wolfram’s PRG
  - Asiacrypt ’91: I break Damgård’s hash function
- Problem: lack of diffusion ... and even worse for γ
- Search for invertible CA rules of neighborhood size 4 and 5:
  - *complementing landscape:* similar to γ but even worse
  - linear rules: e.g. $a_i = a_{i-1} \oplus a_i \oplus a_{i+1}$ not suited
  - Some *enigmatic rules*: excellent diffusion

Patching the stream cipher

- Enigmatic rules
  - invertible when length is co-prime to 6
  - γ followed by $a_i = a_{i-1} \oplus a_i \oplus a_{i+1}$
  - ... and variants
- Stream cipher proposal (WIC ‘91):
  - Introduce linear CA rule for diffusion: θ
  - use $\theta \circ \gamma$ instead of γ
  - Output *distant* bits

Patching the hash function

- Replace CA by *engine* with better diffusion
  - Diffusion of $\theta \circ \gamma$ grows only linearly
- Add bit transposition π: $a_i = a_{z*i \bmod length}$
  - Disperses neighboring bits
  - *Not a CA rule*
- $\pi \circ \theta \circ \gamma$ leads to exponential diffusion
  - No longer CA
  - “Wide Trail”

DC and LC properties of γ

- Differential probability: $\mathrm{DP}(a',b')$
  - $\mathrm{DP}(a',b') = 2^{-\mathrm{w}(a')}$ with $\mathrm{w}(a')$ *restriction weight* of a'
  - Conditions imposed by differential are linear
- Input-output correlation: $\mathrm{LP}(u,v)$
  - $\mathrm{LP}(u,v) = 2^{-\mathrm{w}'(\mathrm{v})}$ with $\mathrm{w}'(v)$ *correlation weight* of v

RIPE (’89)

- EC project in the RACE framework
  - open call for cryptographic primitives
  - symmetric crypto: hash functions only
  - evaluation by consortia members
    - from academia and industry
- Several submissions:

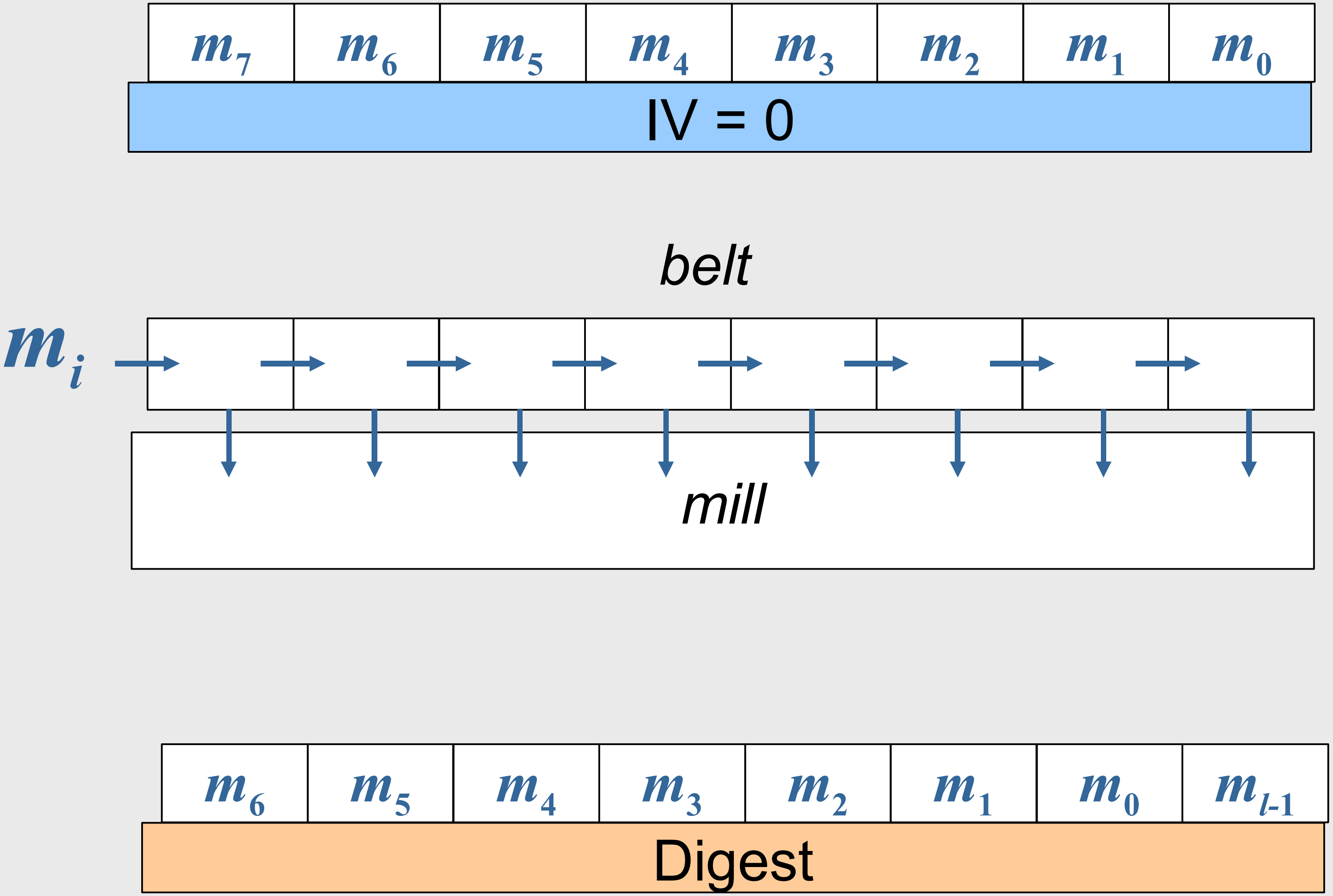
Merkle:	Snefru,
Rivest:	MD4 and later MD5
Schnorr:	FFT-Hash I

Cellhash: structure

- *Iterated block cipher*:
  - inspired by compression function of MD4:
  - digest is encryption of fixed $IV = 00...0$ with message as key
  - block length: 257 bits
- Round function (Mill):

  $\forall\ \pi \circ \theta \circ \gamma$ alternated with message injection σ
  - Due to π: unsuited for software
- # rounds:
  - # 32-bit blocks in padded message m (≥ 8)
- Message expansion (Belt): simple shift register

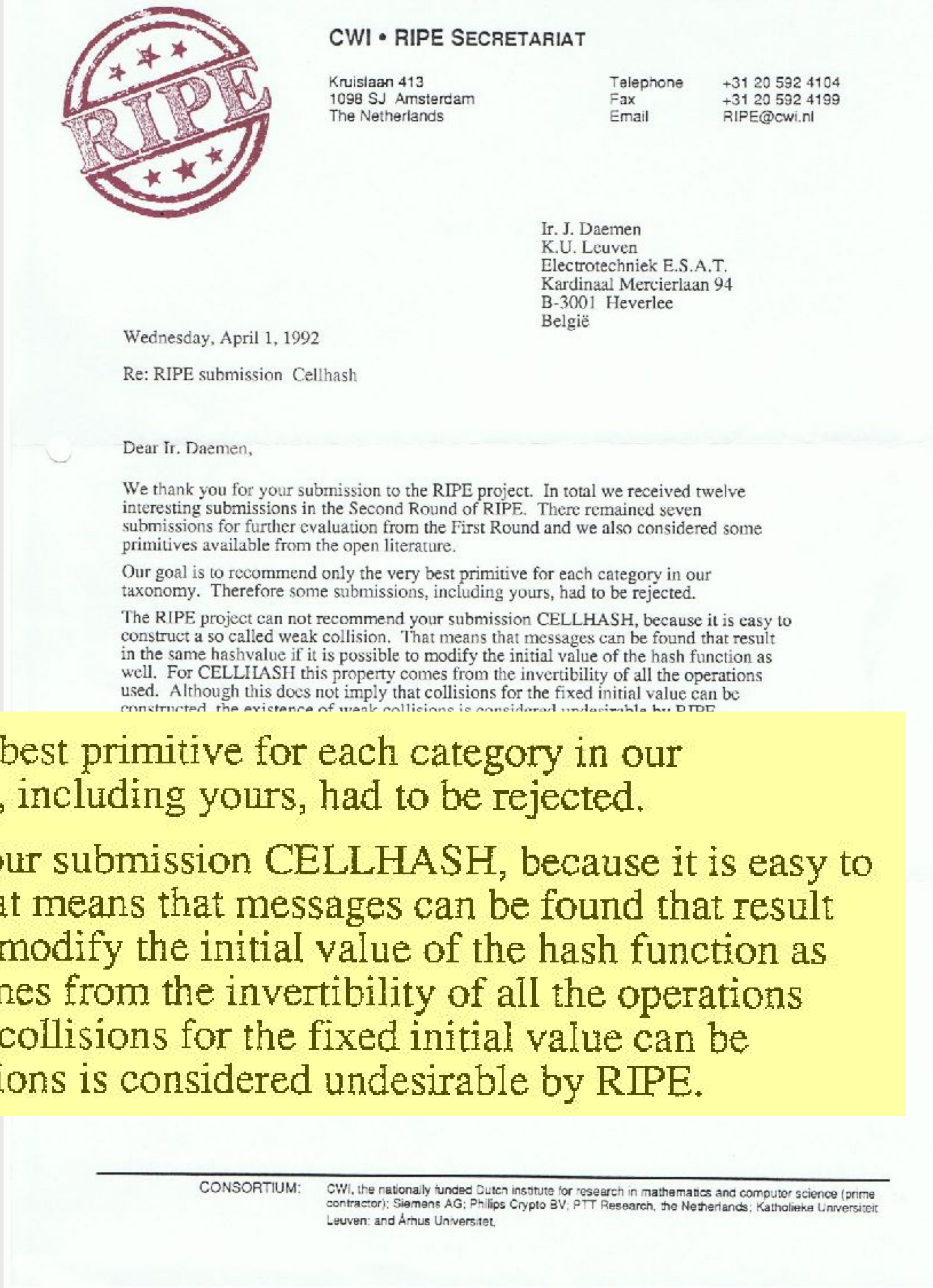
Cellhash operation


m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0
IV = 0
belt
m_i
mill
m_6 m_5 m_4 m_3 m_2 m_1 m_0 m_{l-1}
Digest

Cellhash distinguishing features

- Does not follow Merkle-Damgård:
  - *“fixing input length makes design of collision-resistant compression function easier”*
  - no collision-resistant FIL compression function (like FFT-Hash)
  - no MD-strengthening
- Streaming way of operation:
  - Round function cryptographically weak as such, but ...
    - strength expected from iterated application
    - similar to rounds in block ciphers, steps in MD4, Snefru, ...
  - Invertible and no feed-forward loop (Davies-Meyer):
    - (2nd) pre-image: meet-in-the-middle: $2^{n/2}$ instead of 2^{n}
- Large chaining value: 257 bits instead of usual 128

April 1, ’92



RIPE

CWI • RIPE SECRETARIAT

Kruislaan 413
1098 SJ Amsterdam
The Netherlands

Telephone	+31 20 592 4104
Fax	+31 20 592 4199
Email	RIPE@cwi.nl

Ir. J. Daemen
K.U. Leuven
Electrotechniek E.S.A.T.
Kardinaal Mercierlaan 94
B-3001 Heverlee
België

Wednesday, April 1, 1992

Re: RIPE submission Cellhash

Dear Ir. Daemen,

We thank you for your submission to the RIPE project. In total we received twelve interesting submissions in the Second Round of RIPE. There remained seven submissions for further evaluation from the First Round and we also considered some primitives available from the open literature.

Our goal is to recommend only the very best primitive for each category in our taxonomy. Therefore some submissions, including yours, had to be rejected.

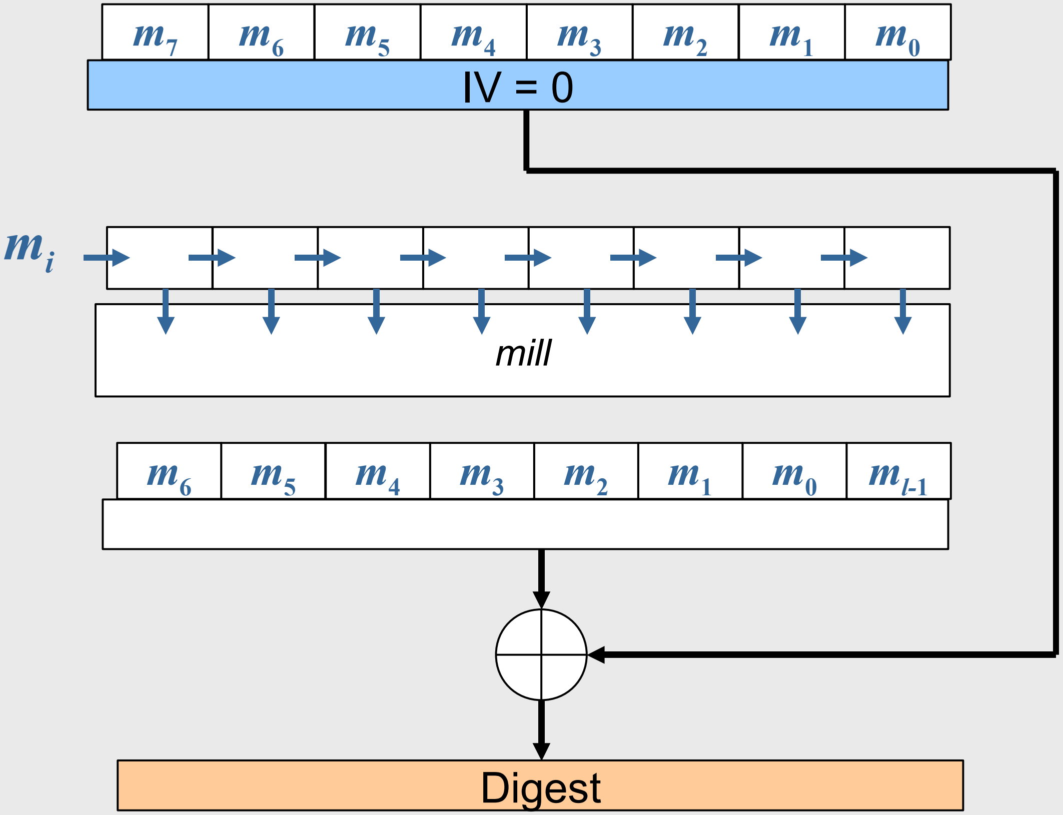
The RIPE project can not recommend your submission CELLHASH, because it is easy to construct a so called weak collision. That means that messages can be found that result in the same hashvalue if it is possible to modify the initial value of the hash function as well. For CELLHASH this property comes from the invertibility of all the operations used. Although this does not imply that collisions for the fixed initial value can be constructed, the existence of weak collisions is considered undesirable by RIPE.

Our goal is to recommend only the very best primitive for each category in our taxonomy. Therefore some submissions, including yours, had to be rejected.

The RIPE project can not recommend your submission CELLHASH, because it is easy to construct a so called weak collision. That means that messages can be found that result in the same hashvalue if it is possible to modify the initial value of the hash function as well. For CELLHASH this property comes from the invertibility of all the operations used. Although this does not imply that collisions for the fixed initial value can be constructed, the existence of weak collisions is considered undesirable by RIPE.

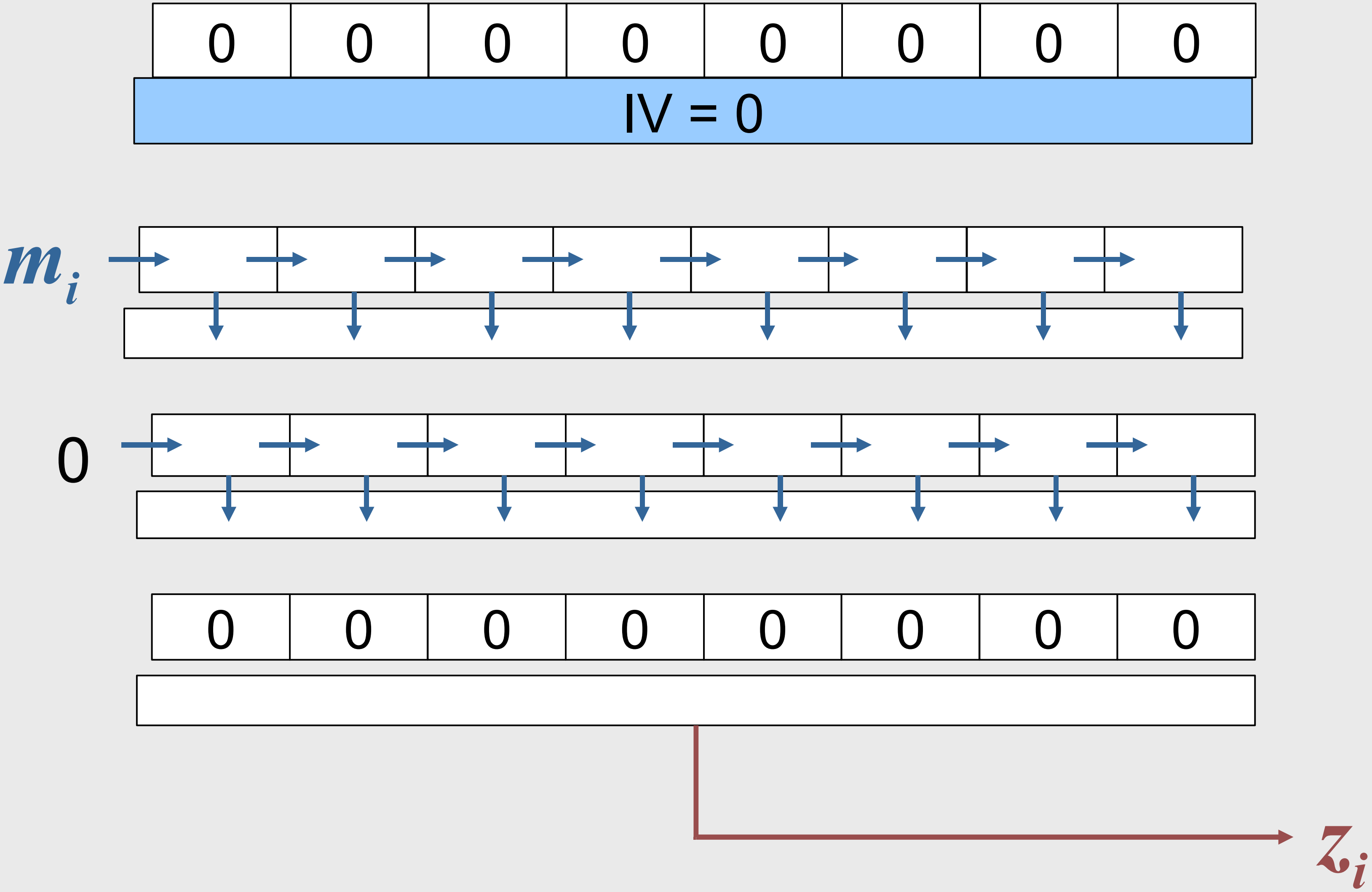
CONSORTIUM: CWI, the nationally funded Dutch institute for research in mathematics and computer science (prime contractor); Siemens AG; Philips Crypto BV; PTT Research, the Netherlands; Katholieke Universiteit Leuven; and Århus Universitet.

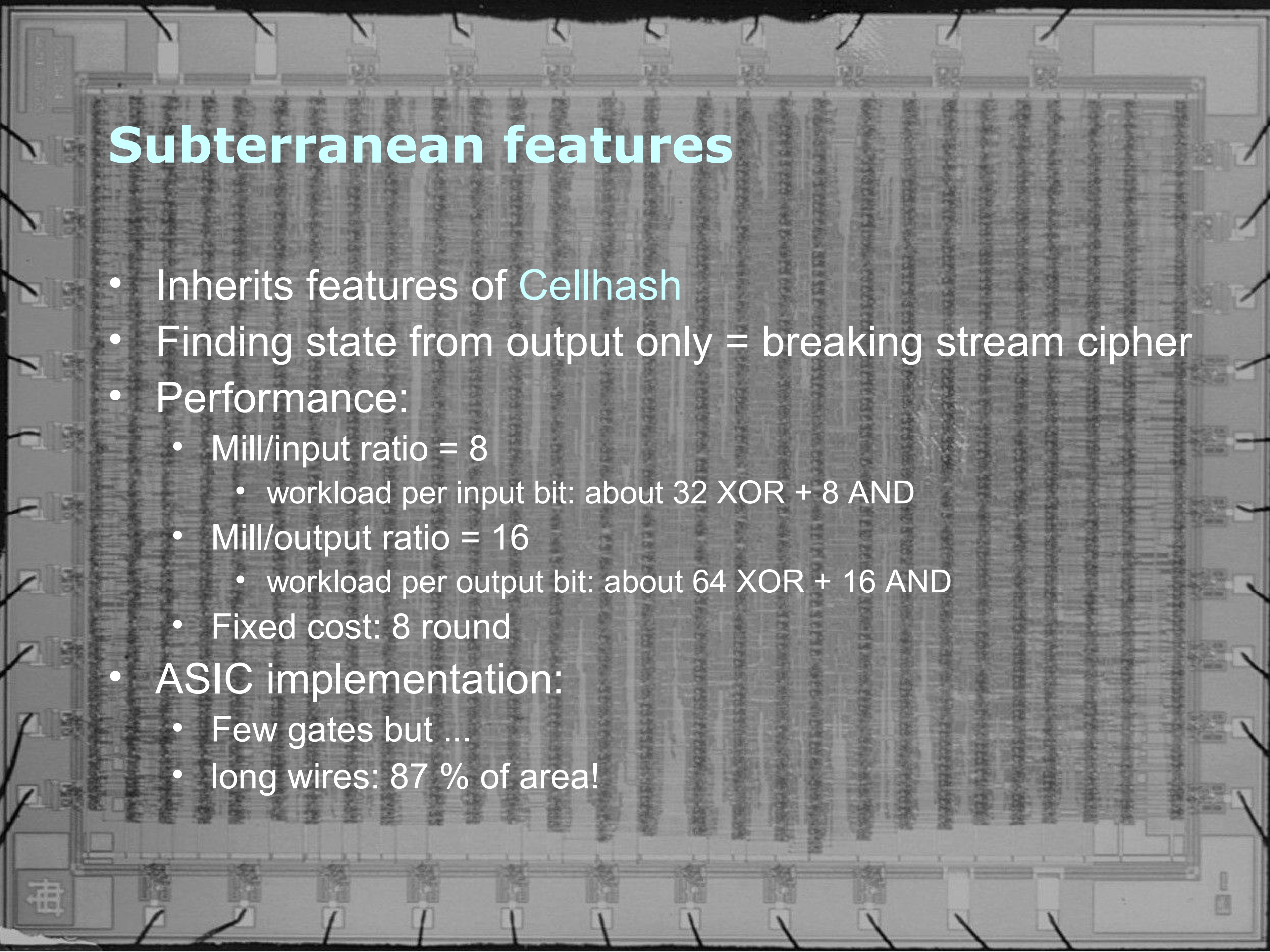
Beunhash



= Cellhash... without weak collisions ;-)

Subterranean ('91)





Subterranean features

- Inherits features of Cellhash
- Finding state from output only = breaking stream cipher
- Performance:
  - Mill/input ratio = 8
    - workload per input bit: about 32 XOR + 8 AND
  - Mill/output ratio = 16
    - workload per output bit: about 64 XOR + 16 AND
  - Fixed cost: 8 round
- ASIC implementation:
  - Few gates but ...
  - long wires: 87 % of area!

Status of Cellhash/Subterranean today

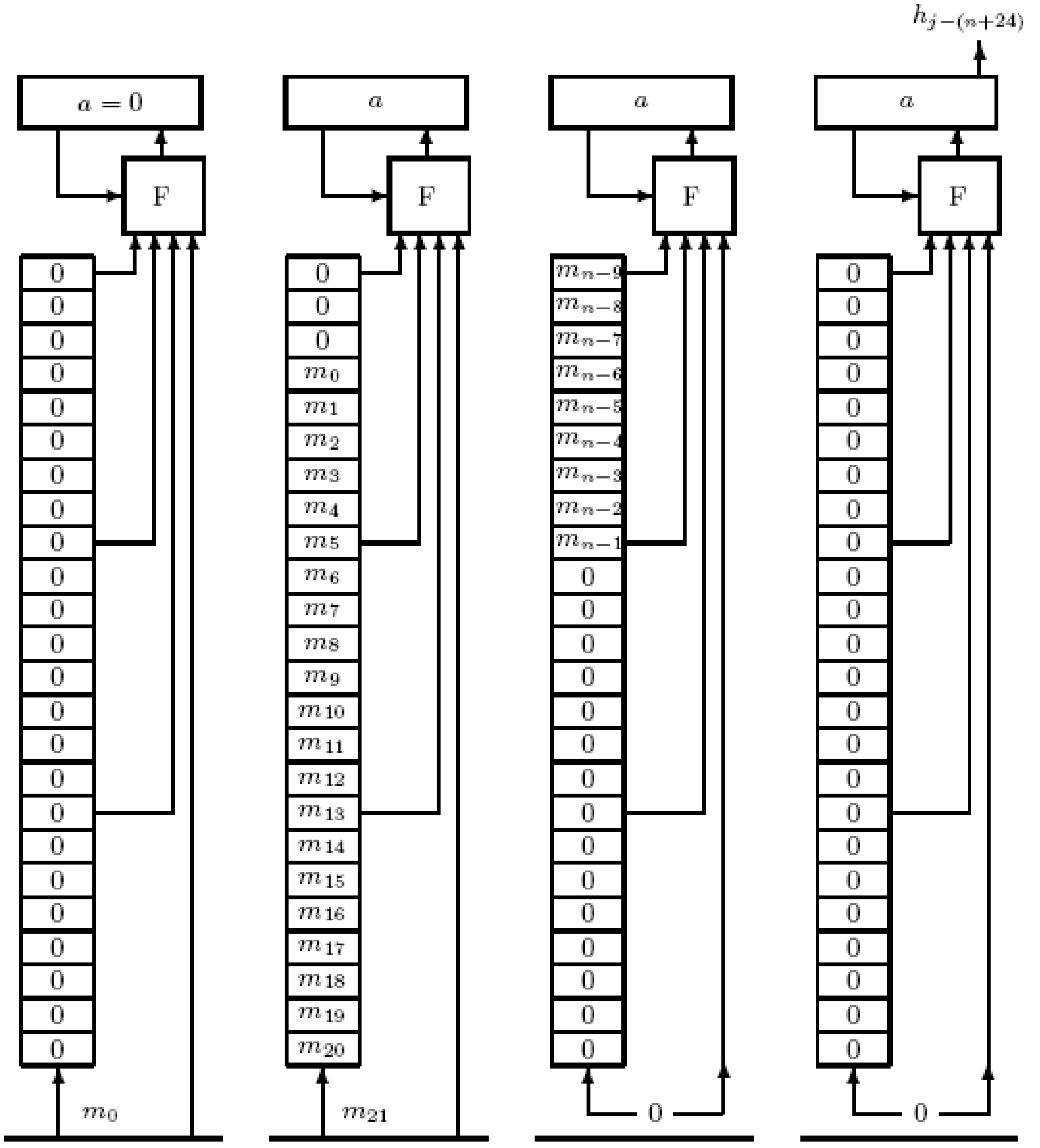
- Cellhash theoretical weaknesses:
  - Length-extension: trivial to do
  - (2nd) pre-image attack: about $2^{128,5}$ calls (< 2^{257} for RO)
- Subterranean:
  - Length-extension problem solved
  - No attacks found with complexity below 2^{128} calls to round function
  - Structure appears to be sound
- Only cryptanalysis paper known to me:
  - pre-image attack by Donghoon Chang at http://eprint.iacr.org/2006/412
  - Meet-in-the-middle applied to pre-image finding

Boognish ('92-'93)

- First attempt of translating Subterranean to software
- Word-oriented: 32-bit words
- Mill: 5 words
  - ∀ γ, θ and belt injection σ: bit-slice
  - ∀ π: rotations within the words
- Belt: shift register
  - 1 word wide
  - 24 words long
- Mill/input ratio: 5

Boognish operation

- Failure: Mill too small



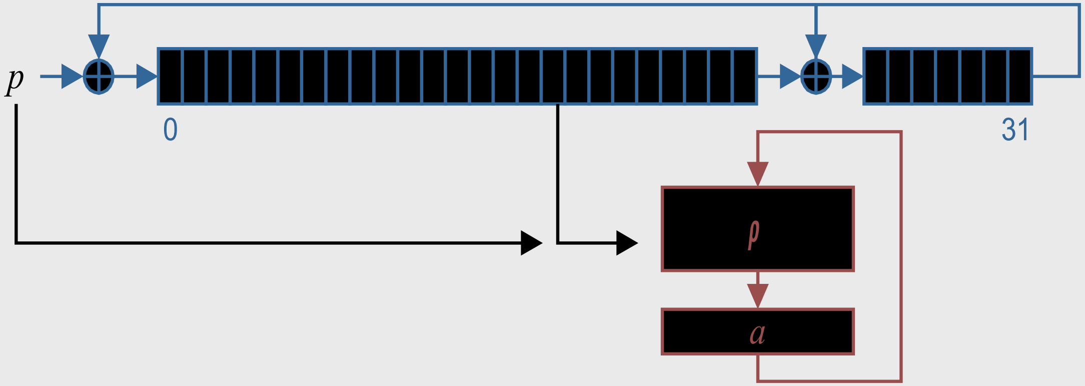
$j = 0$ $j = 21$ $j = n + 15$ $n + 24 \leq j < n + 24 + p$

StepRightUp (’94) and Panama (’98)

- “Subterranean for Software”: Stream and hash
- StepRightUp never published but appeared in thesis
- Panama: StepRightUp with very small tweak
  - Joint work with Craig Clapp
  - Published at FSE ’98
- Belt: linear feedback shift register
  - 8 words wide
  - 32 words long
- Characteristics:
  - Mill/input ratio ≈ 2
  - Huge mill
  - Feedback from Mill to Belt in pull rounds
  - Security claim: *hermetic* ≈ behave as a random oracle

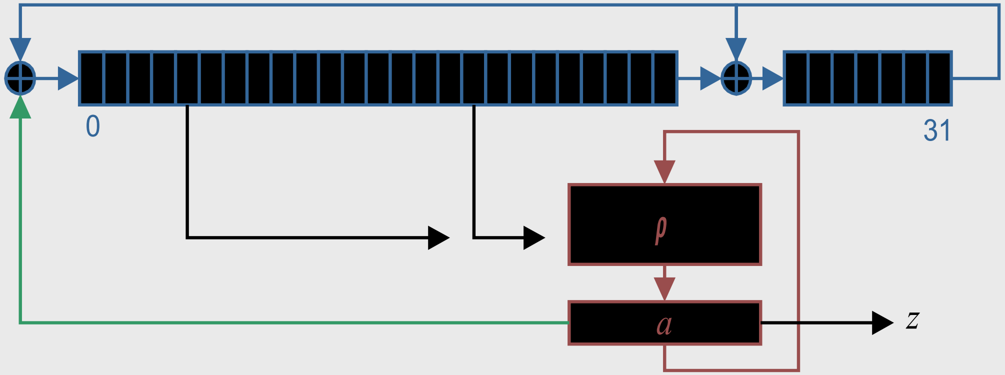
Panama operation

push rounds:

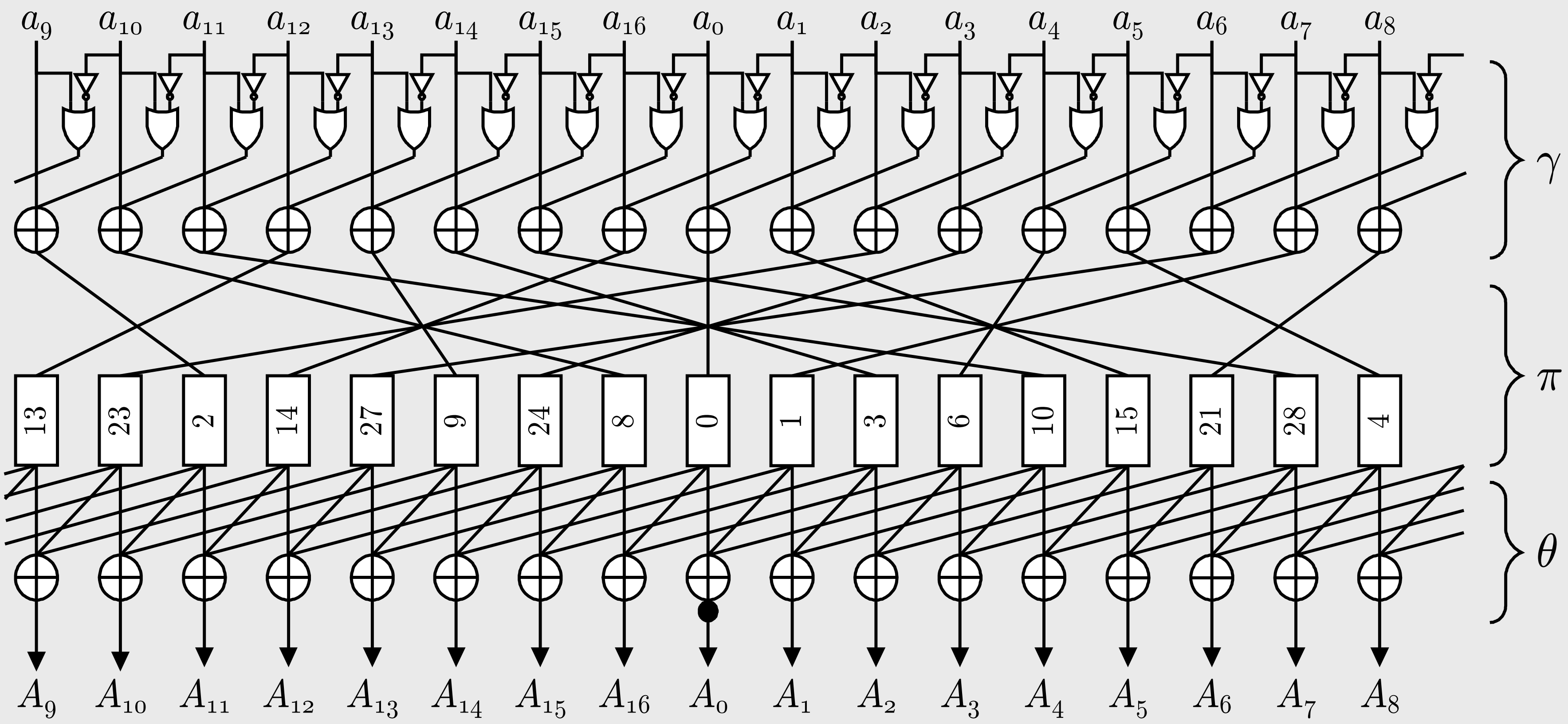


blank rounds

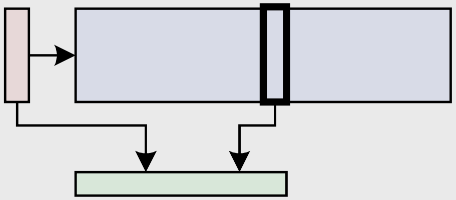
pull rounds:



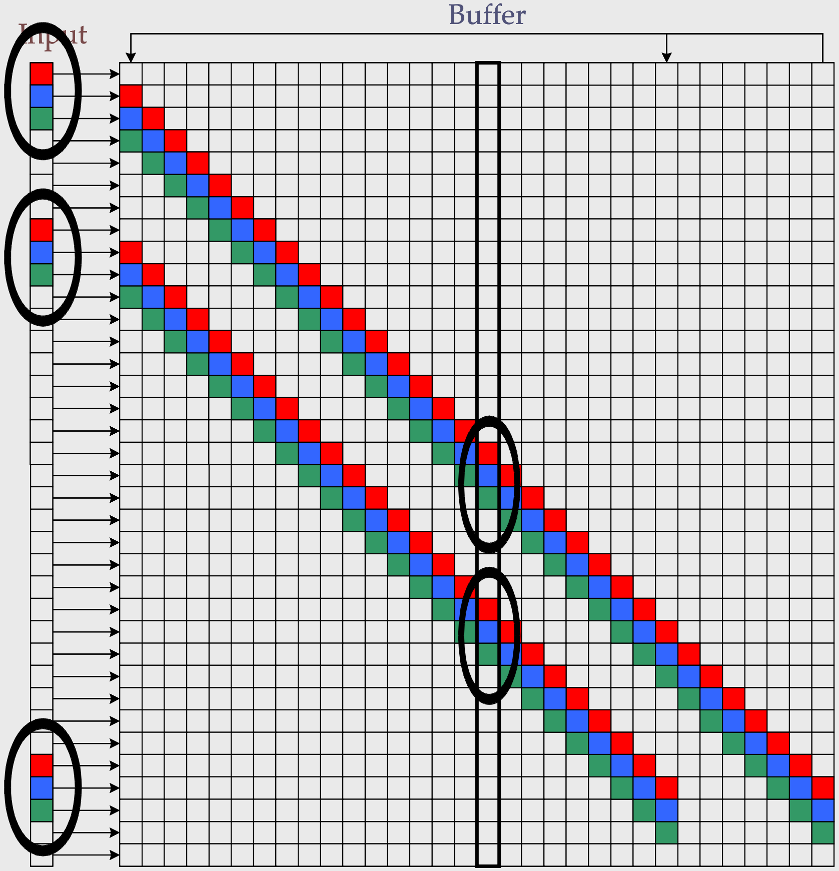
Panama Mill function


a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8
γ
π
13 23 2 14 27 9 24 8 0 1 3 6 10 15 21 28 4
θ
A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8

Collisions in Panama



- Belt collisions
  - Atom
  - Rijmen et al. at FSE ’01
  - Our attack at FSE ’07
- State injection
  - Five sub-collisions, treated independently
- Discussion
  - Both attacks exploit symmetry
    - but that is not THE problem
  - Mill/input ratio too low!
    - 8 words of freedom per round for the propagation in the 17-word mill
    - Trails with low backtracking cost



Fixing Panama

- Mill/Input ratio ↑: reduce input size... but also hashing speed :-(
- Make sub-collisions overlap
  - more taps in belt LFSR?
  - more feed-forward from Belt to Mill?
- But: Panama stream cipher is still unbroken today!
  - Small Mill/output ratio appears to be no problem
  - Maybe including feedback from Mill to Belt (also in Push mode) will help?

Fixing Panama (cont’d)

- Final correction attack:

  ∀ ∀ M, M^*, find X, X^* with $\mathrm{H}(M\ X)=\mathrm{H}(M^*\ X^*)$
  - $X' = X \oplus X^*$ fully determined by M and M^*
  - find a trail in Mill such that it results in a collision
  - small Mill/input ratio gives a small backtracking cost, ...
  - independent of what the belt looks like!
- So:
  - Reduce input size
  - Include feedback from Mill to Belt
  - Simplify Belt
- Result is RadioGatún (Bertoni, Daemen, Peeters, Van Assche)

Lessons learnt

1) Invertible round function
2) Streaming-oriented hashing (with *blank rounds* at end)
    - Trail backtracking as reference attack
    - Mill/Input ratio shall not be too small
3) Symmetry and parallelism of CA rules
4) Gradual output extraction and variable-length output
    - Reference: *Sponge Functions*
5) Belt-and-Mill construction
    - Belt without feedback from Mill is useless