Subspace Distinguisher for 5/8 Rounds of the ECHO-256 Hash Function

Martin Schläffer

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Austria

martin.schlaeffer@iaik.tugraz.at

Outline

Outline

Analysis of SHA-3 Candidates

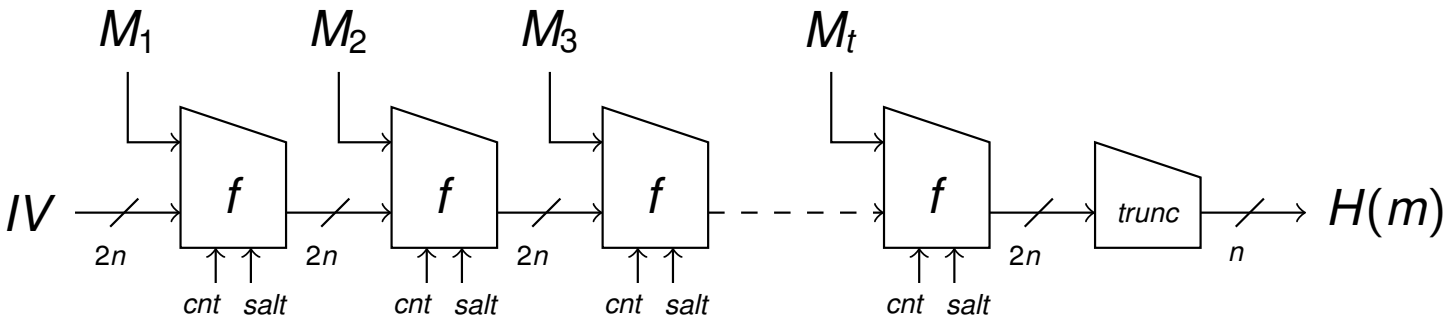
- Mostly attacks on building blocks
  - most of the time distinguishers
  - often high complexities, weak properties
  - sometimes better generic attacks exist
  - unknown how properties extend to hash function
  - can serve as lower bound for attacks on hash function
- Few attacks on hash function
  - can help in detecting more interesting properties
  - easier to compare attacks on hash functions
  - better evaluation of security margins
  - more work for attacker

How to Evaluate Attacks on SHA-3 Candidates?

- Number of attacked rounds (security parameter)
- Complexity of the attacks
- Which part has been attacked
- Which property has been found
- How much freedom is left in the attacks?
- Number of (non-incremental) independent results
- Effort spent on each hash function?
- Is the hash function easy to analyze?

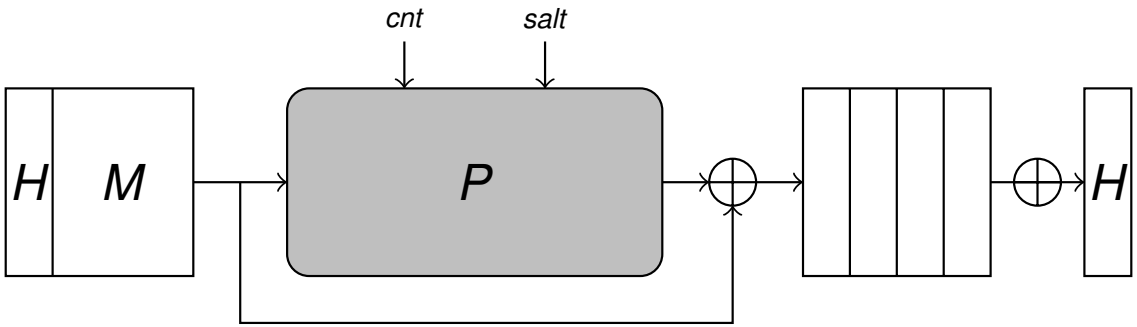
Outline

Description of ECHO [BBG+08]



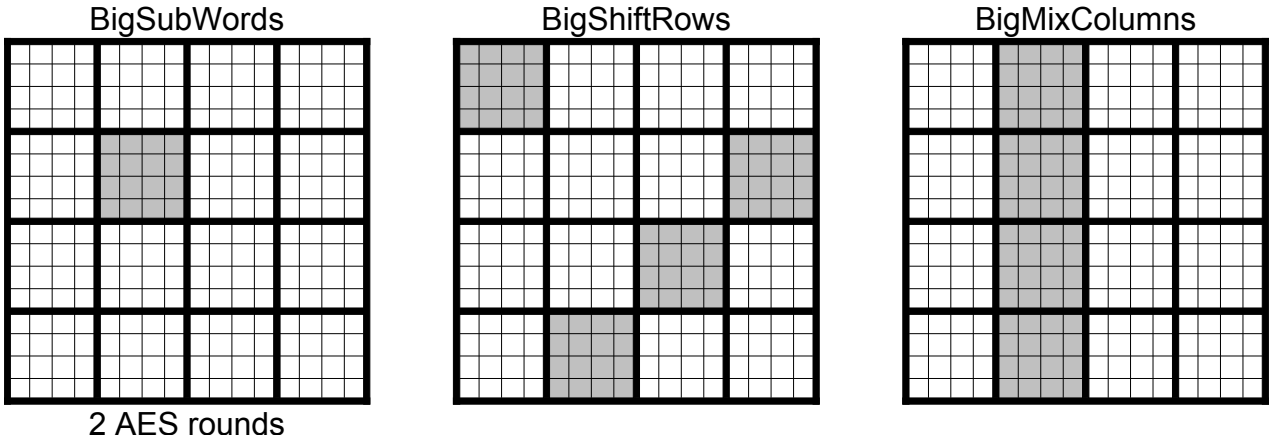
- Hash function submitted to the NIST SHA-3 competition
  - Round 2 candidate
  - AES based design
- Iterated hash function
  - Haifa design principle
  - double-pipe construction
  - output transformation: simple truncation
  - security based on non-randomness of compression function

The Compression Function of ECHO-256



- ECHO Permutation P
  - large 2048-bit permutation
  - tweak inputs (counter, salt)
  - AES based round transformations
- Finalization:
  - feed-forward of (H, M)
  - compression of 1024-bit columns

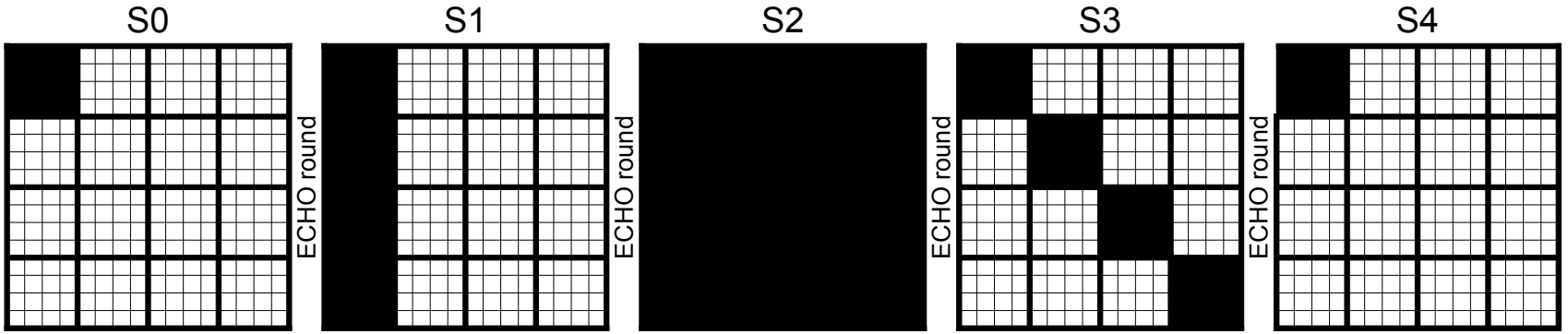
Round Transformations of ECHO



- ECHO state:
  - 4×4 AES states
  - AES inside AES
  - 8 rounds (ECHO-256) or 10 rounds (ECHO-512)
- Round transformations:
  - 128-bit BigSubWords (two standard AES rounds)
  - BigShiftRows shifts AES states
  - BigMixColumns mixes AES states (16 parallel MixColumns)

Outline

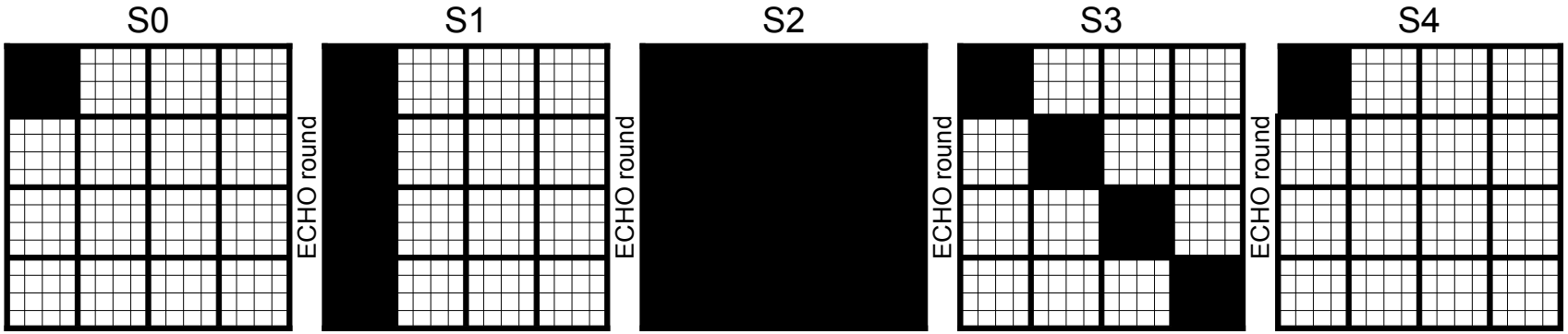
Simple 4-Round Truncated Differential Path



- Minimal 4-round AES path at word level [MPRS09]
  - number of active words: $1 \rightarrow 4 \rightarrow 16 \rightarrow 4 \rightarrow 1$
  - full active SubBytes layer
  - 800 active S-boxes

$\Rightarrow$ just enough freedom for Rebound Attack with one inbound phase

Simple 4-Round Truncated Differential Path

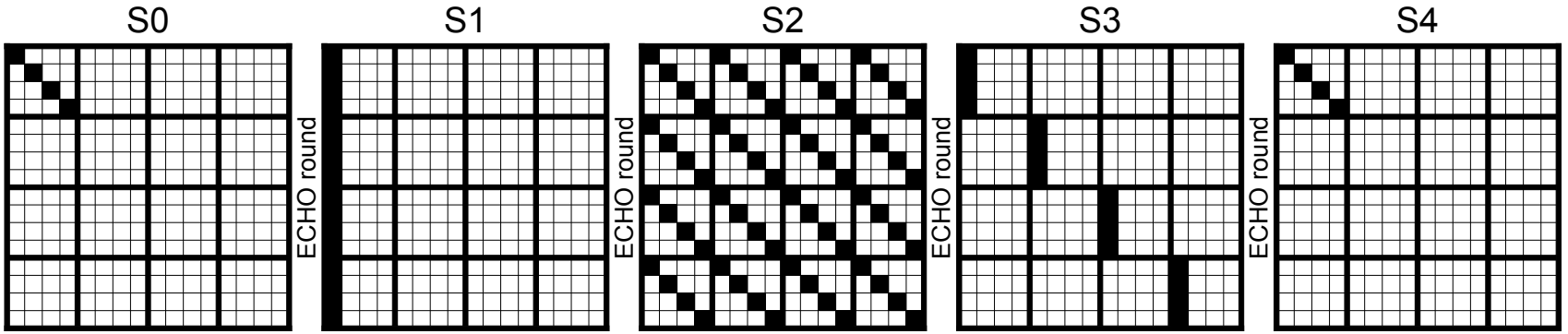


- Minimal 4-round AES path at word level [MPRS09]
  - number of active words: $1 \rightarrow 4 \rightarrow 16 \rightarrow 4 \rightarrow 1$
  - full active SubBytes layer
  - 800 active S-boxes

$\Rightarrow$ just enough freedom for Rebound Attack with one inbound phase

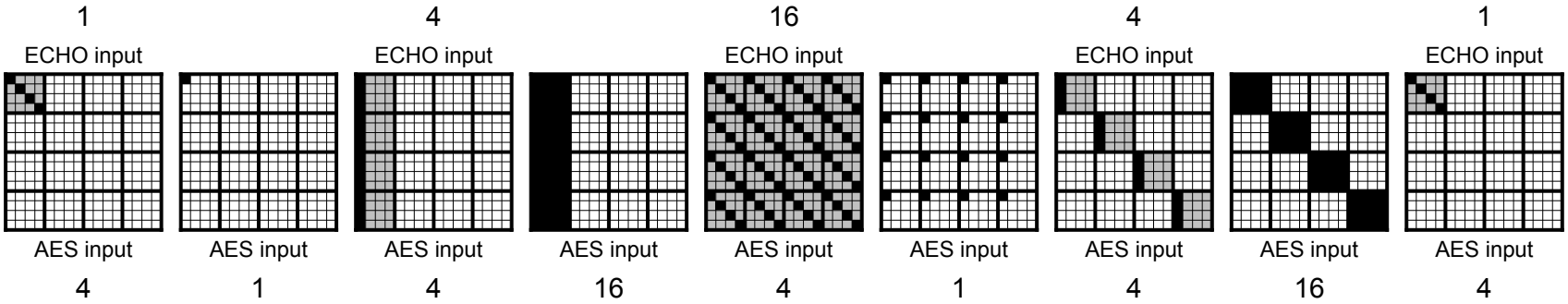
- Nevertheless: distinguishers for full ECHO-256 permutation
  - due to many improvements in the last year [GP10, Pey10a, Pey10b, SLW+10]

Sparse 4-Round Truncated Differential Path



- Sparse AES states in full active ECHO states
  - same number of active words: $1 \rightarrow 4 \rightarrow 16 \rightarrow 4 \rightarrow 1$
  - only 245 active S-boxes (lower bound: 200 [BBG+08])

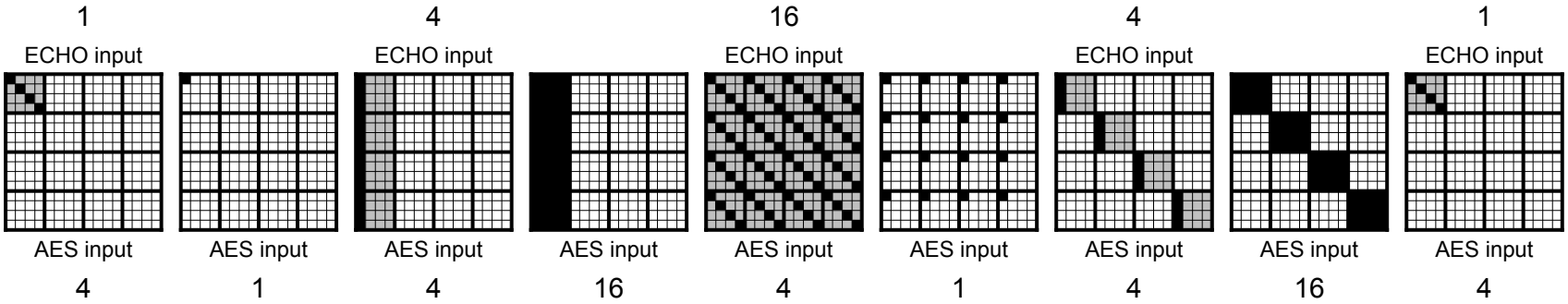
Sparse 4-Round Truncated Differential Path



- Sparse AES states in full active ECHO states
  - same number of active words: $1 \rightarrow 4 \rightarrow 16 \rightarrow 4 \rightarrow 1$
  - only 245 active S-boxes (lower bound: 200 [BBG+08])
  - at most 1/4 of all S-boxes are active (per state)

$\Rightarrow$ more freedom: better attacks

Sparse 4-Round Truncated Differential Path

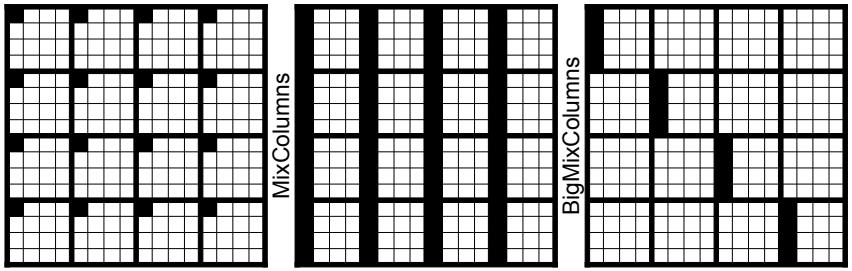


- Sparse AES states in full active ECHO states
  - same number of active words: $1 \rightarrow 4 \rightarrow 16 \rightarrow 4 \rightarrow 1$
  - only 245 active S-boxes (lower bound: 200 [BBG+08])
  - at most 1/4 of all S-boxes are active (per state)

$\Rightarrow$ more freedom: better attacks

- Rebound Attack with multiple inbound phases
  - Whirlpool, LANE [LMR+09, MNPN+09]
  - less complexity
  - longer paths

MixColumns and BigMixColumns



- Properties of MC and BMC:
  - both have branch number 5
  - applied to 1x16 column of state
- Combined SuperMixColumns transformation (also [SLW+10]):
  - has branch number 8
  - ideal case: branch number 17

$\Rightarrow$ Sparse truncated differential paths

Computing the Available Degrees of Freedom

- For a given (truncated) differential path
  - determine probability of this path P_{dp}
  - total number of possible input pairs N_{in}
  - estimate the number of "surviving" output pairs:

$$N_{out} = N_{in} \cdot P_{dp}$$

  - degrees of freedom: $log_2(N_{out})$

Computing the Available Degrees of Freedom

- For a given (truncated) differential path
  - determine probability of this path P_{dp}
  - total number of possible input pairs N_{in}
  - estimate the number of "surviving" output pairs:

$$N_{out} = N_{in} \cdot P_{dp}$$

  - degrees of freedom: $log_2(N_{out})$
- $log_2(N_{out}) << 0$:
  - attack does not work with high probability
  - cannot be repeated with new input pair
  - restart with new (probably less efficient) path
- $log_2(N_{out}) >> 0$:
  - many right pairs exist (we need to find just one)
  - lots of "freedom" left for attacker to find a pair

Computing the Available Degrees of Freedom

- For a given (truncated) differential path
  - determine probability of this path P_{dp}
  - total number of possible input pairs N_{in}
  - estimate the number of "surviving" output pairs:

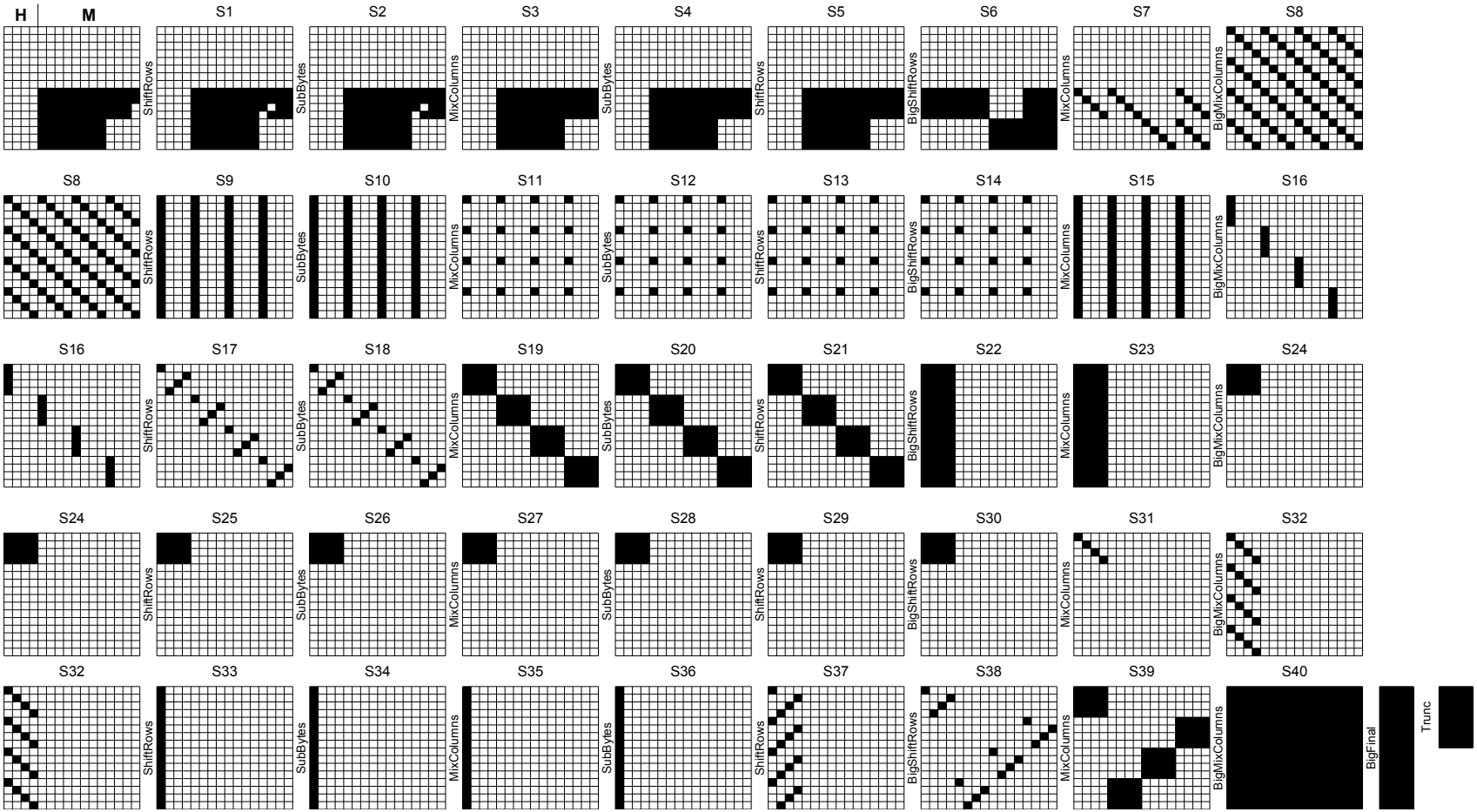
$$N_{out} = N_{in} \cdot P_{dp}$$

  - degrees of freedom: $log_2(N_{out})$
- $log_2(N_{out}) << 0$:
  - attack does not work with high probability
  - cannot be repeated with new input pair
  - restart with new (probably less efficient) path
- $log_2(N_{out}) >> 0$:
  - many right pairs exist (we need to find just one)
  - lots of "freedom" left for attacker to find a pair

This evaluation is independent of an actual Attack!

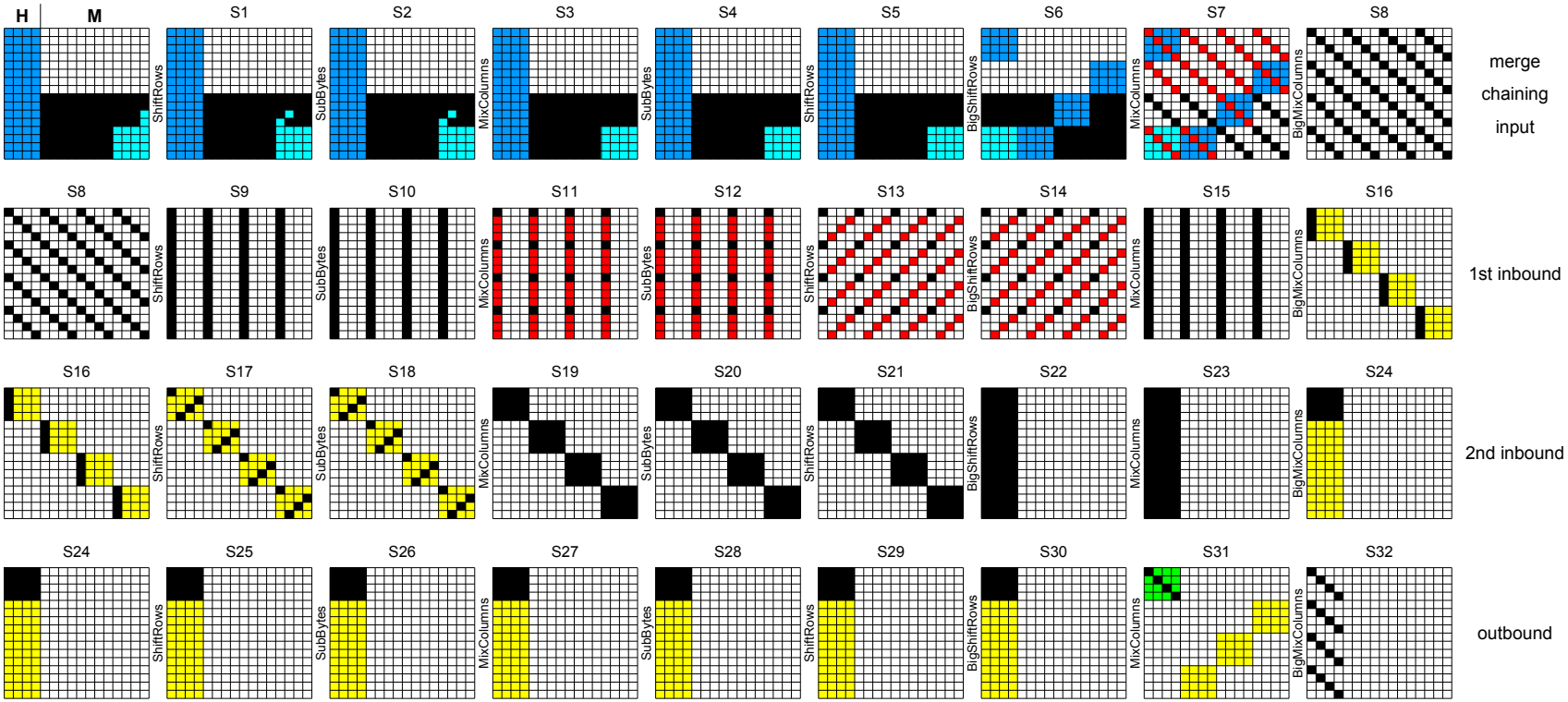
Outline

Rebound Attack with Multiple Inbound Phases



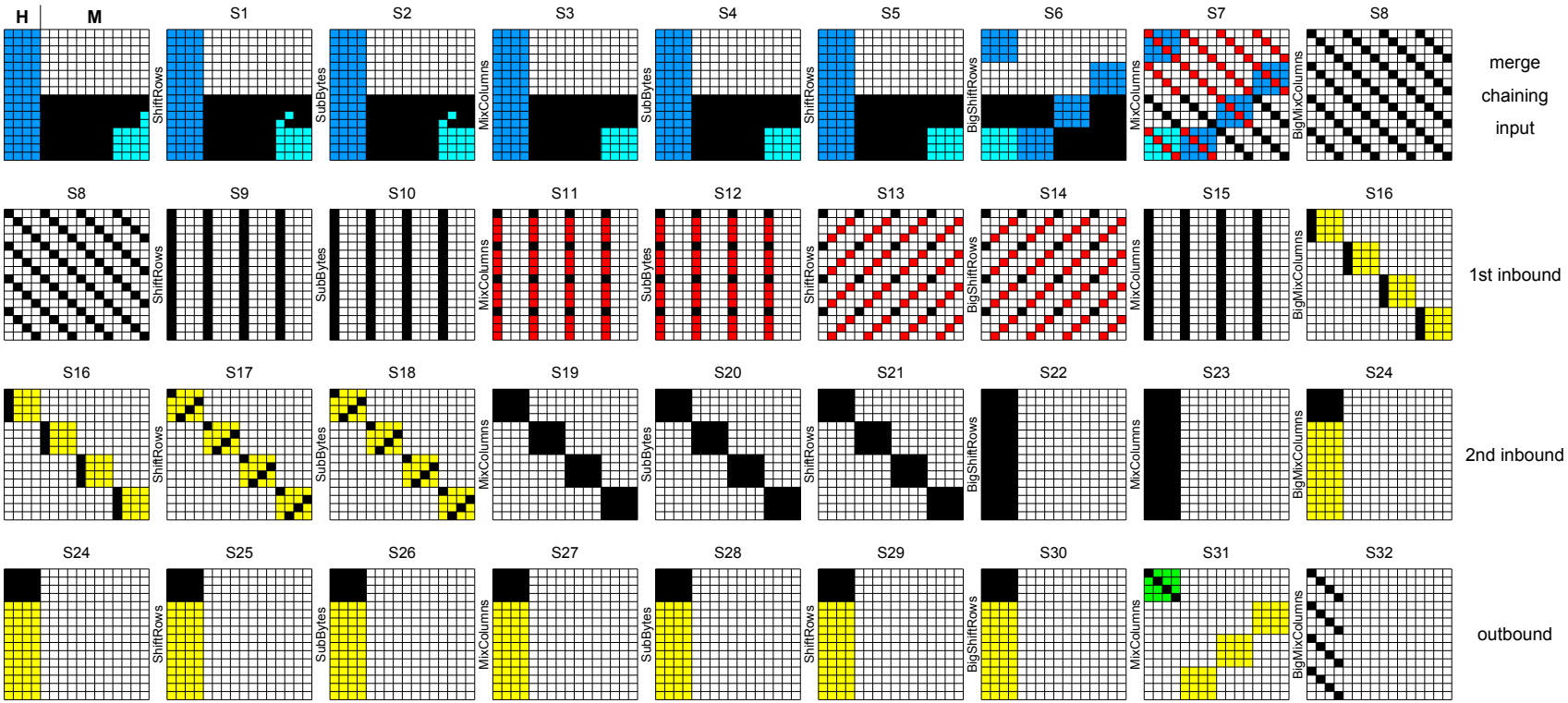
- enough freedom for more than one inbound phase
- merge independent solutions using birthday effect

Outline of the Rebound Attack



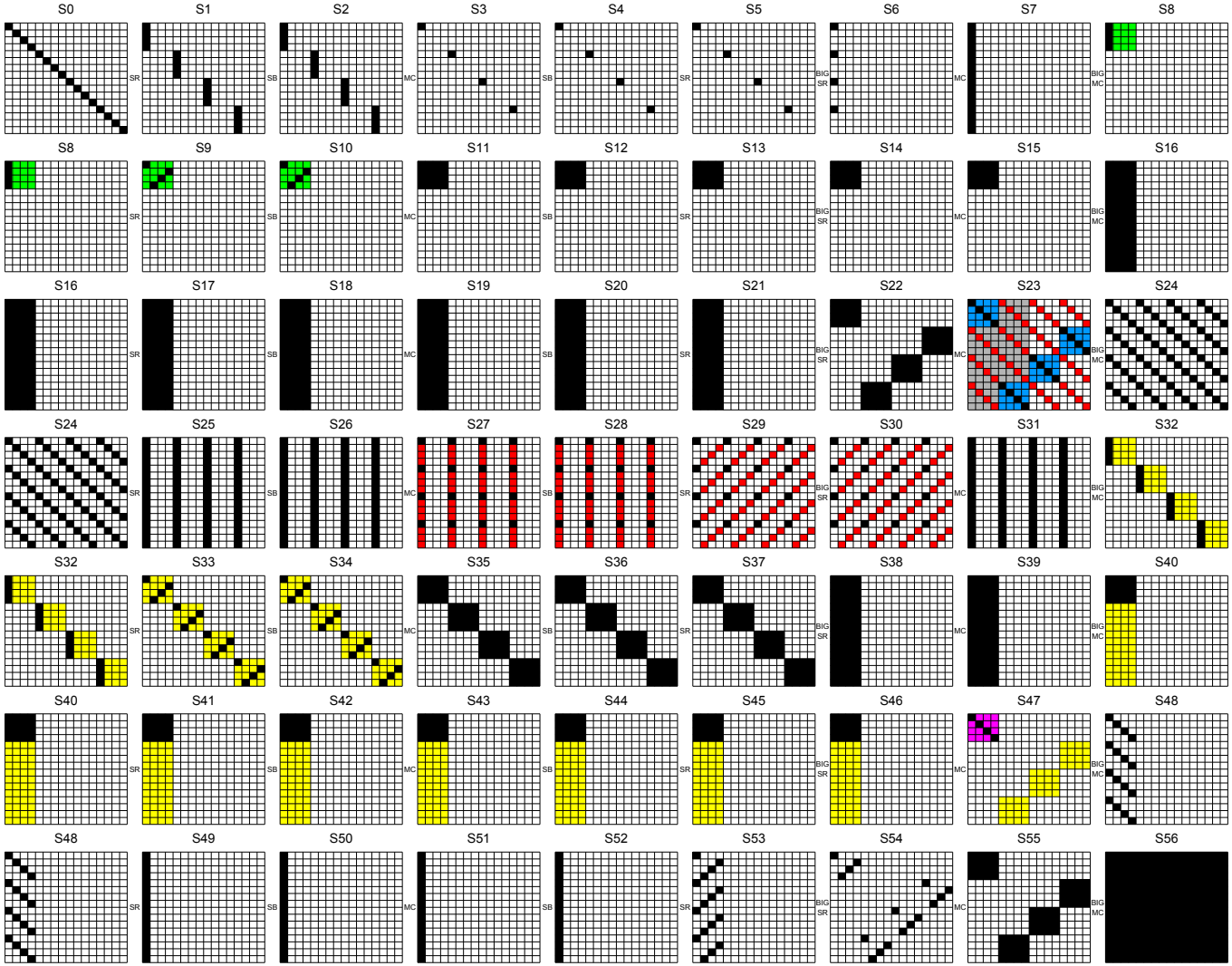
- 1st inbound phase (■, complexity 2^{32}, average 1)
- merge chaining input (■, complexity 2^{32})
- 2nd inbound phase (■, complexity 2^{64}, average 1)
- outbound phase (■, complexity 2^{96})
- merge inbound phases (□, complexity 2^{64})

Rough Estimation of Degrees of Freedom



- 1st inbound phase $\sim 1/4$ of the state (■)
- chaining input $\sim 1/4$ of the state (■)
- 2nd inbound phase $\sim 1/4$ of the state (■)
- available freedom $\sim 1/4$ of the state
  also freedom in differences (■)

Compression Function Dist. for 7/8 Rounds



$\sim$800 degrees of freedom ($\sim 2^{800}$ pairs for this path exist)

Outline

Results and Related Analysis

Function	Target	Rounds	Time	Memory	Type	Ref.
ECHO-256 (8 Rounds)	hash function	5	2^{96}	2^{64}	distinguisher	this work
		4.5	2^{96}	2^{64}	near-collision	this work
		4	2^{64}	2^{64}	collision	this work
	compression function	7	2^{152}	2^{64}	distinguisher[1]	this work
		7	2^{107}	2^{64}	distinguisher*	this work
		6.5	2^{152}	2^{64}	free-start near-collision	this work
		6.5	2^{96}	2^{64}	free-start near-collision*	this work
		4.5	2^{96}	2^{32}	distinguisher	[Pey10b]
		3	2^{64}	2^{32}	free-start collision	[Pey10b]
		3	2^{96}	2^{32}	semi-free-start collision*	[Pey10b]
ECHO-512 (10 Rounds)	compression function	7	2^{162}	2^{64}	distinguisher	this work
		7	2^{106}	2^{64}	distinguisher*	this work
		6.5	2^{152}	2^{64}	free-start near-collision	this work
		6.5	2^{96}	2^{64}	free-start near-collision*	this work
		6.5	2^{96}	2^{32}	distinguisher	[Pey10b]
		3	2^{96}	2^{32}	(semi-)free-start collision*	[Pey10b]

* chosen salt

[1] limited birthday distinguisher

(all attacks also without chosen salt)

Conclusions

- Easy analysis of AES-based hash functions
  - construct good paths by hand
  - no complicated path search tools needed
  - get good results in short amount of time

  $\Rightarrow$ quickley see how far we can go

Conclusions

- Easy analysis of AES-based hash functions
  - construct good paths by hand
  - no complicated path search tools needed
  - get good results in short amount of time

  $\Rightarrow$ quickley see how far we can go
- Sparse truncated differential paths in ECHO:
  - only 1/4 of the state is active
  - sparse also at input and output (“survives” final folding)
  - still leaving large ($>$ 800) degrees of freedom

  $\Rightarrow$ Rebound Attack with multiple inbound phases

Conclusions

- Easy analysis of AES-based hash functions
  - construct good paths by hand
  - no complicated path search tools needed
  - get good results in short amount of time

  $\Rightarrow$ quickley see how far we can go
- Sparse truncated differential paths in ECHO:
  - only 1/4 of the state is active
  - sparse also at input and output ("survives" final folding)
  - still leaving large (> 800) degrees of freedom

  $\Rightarrow$ Rebound Attack with multiple inbound phases
- How to protect against this type of attack?
  - avoid sparse (truncated) differential paths
  - strictly follow wide-trail design strategy: fixed-key AES, Grøstl, ... (best attacks on Grøstl: 0-3 degrees of freedom)

  $\Rightarrow$ not enough freedom for multiple inbound phases

Thank you for your Attention!

Questions?

References I

[BBG+08] Ryad Benadjila, Olivier Billet, Henri Gilbert, Gilles Macario-Rat, Thomas Peyrin, Matt Robshaw, and Yannick Seurin.
SHA-3 Proposal: ECHO.
Submission to NIST, 2008.
Available online: `http://crypto.rd.francetelecom.com/echo`.

[GP10] Henri Gilbert and Thomas Peyrin.
Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations.
In Seokhie Hong and Tetsu Iwata, editors, *FSE*, volume 6147 of *LNCS*, pages 365–383. Springer, 2010.

[LMR+09] Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer.
Rebound Distinguishers: Results on the Full Whirlpool Compression Function.
In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 126–143. Springer, 2009.

[MNPN+09] Krystian Matusiewicz, María Naya-Plasencia, Ivica Nikolic, Yu Sasaki, and Martin Schläffer.
Rebound Attack on the Full Lane Compression Function.
In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 106–125. Springer, 2009.

References II

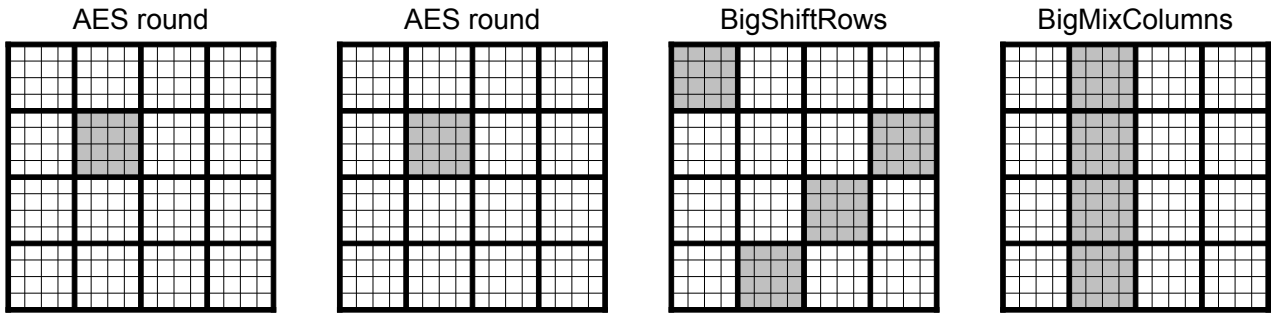
[MPRS09] Florian Mendel, Thomas Peyrin, Christian Rechberger, and Martin Schläffer.
Improved Cryptanalysis of the Reduced Grøstl Compression Function, ECHO Permutation and AES Block Cipher.
In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *LNCS*, pages 16–35. Springer, 2009.

[Pey10a] Thomas Peyrin.
Improved Differential Attacks for ECHO and Grøstl.
In *CRYPTO*, 2010.
to appear.

[Pey10b] Thomas Peyrin.
Improved Differential Attacks for ECHO and Grøstl.
Cryptology ePrint Archive, Report 2010/223, 2010.
Available online: `http://eprint.iacr.org/2010/223`.

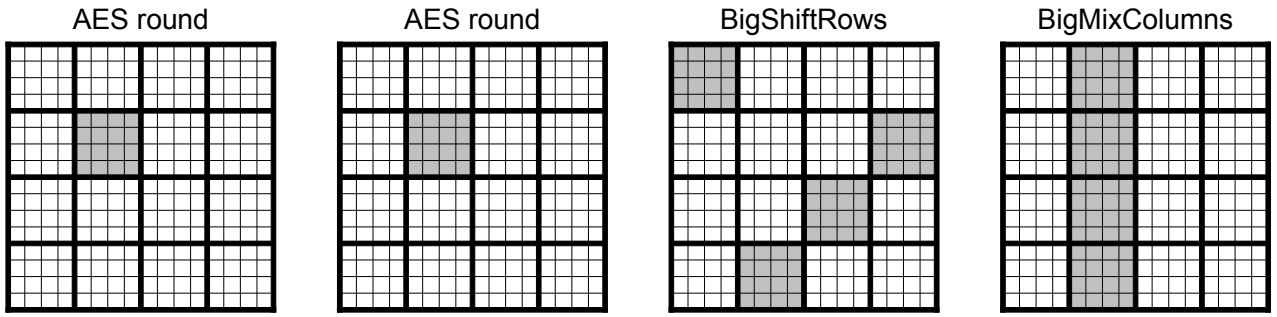
[SLW+10] Yu Sasaki, Yang Li, Lei Wang, Kazuo Sakiyama, and Kazuo Ohta.
New Non-Ideal Properties of AES-Based Permutations: Applications to ECHO and Grøstl.
In *Second SHA-3 Candidate Conference*, 2010.
to appear.

Internal AES Round Transformations



- 10 AES-like round transformations:
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey (counter)
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey (salt)
  - BigShiftRows
  - BigMixColumns

Internal AES Round Transformations

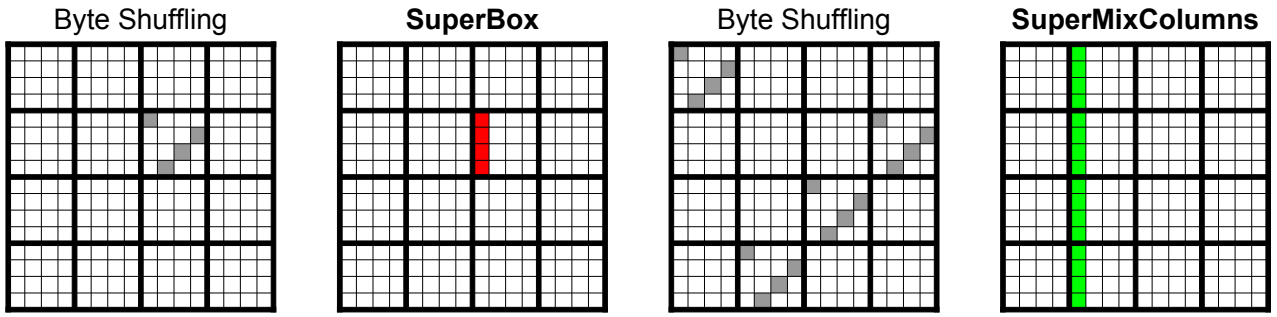


- 10 AES-like round transformations:
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey (counter)
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey (salt)
  - BigShiftRows
  - BigMixColumns

⇐ non-linear

⇐ linear

Internal AES Round Transformations

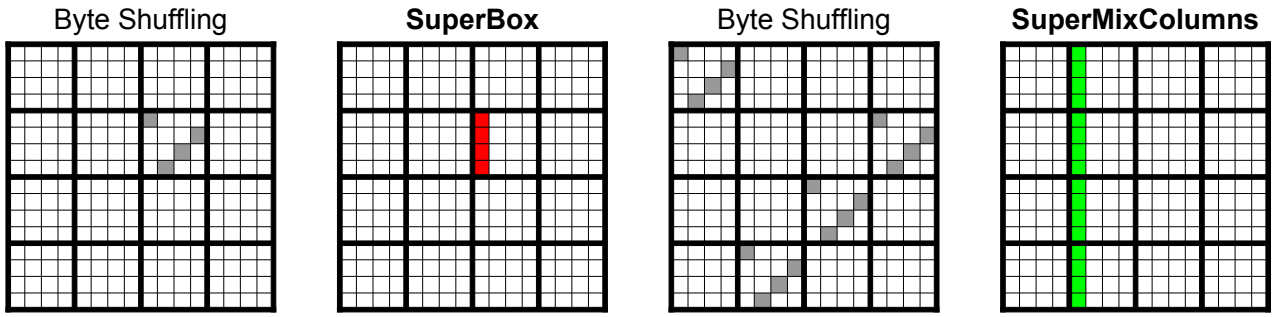


- 10 AES-like round transformations:
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey (counter)
  - SubBytes

  ⇐ non-linear SuperBox
  - ShiftRows
  - MixColumns
  - AddRoundKey (salt)
  - BigShiftRows
  - BigMixColumns

  ⇐ linear SuperMixColumns

Internal AES Round Transformations

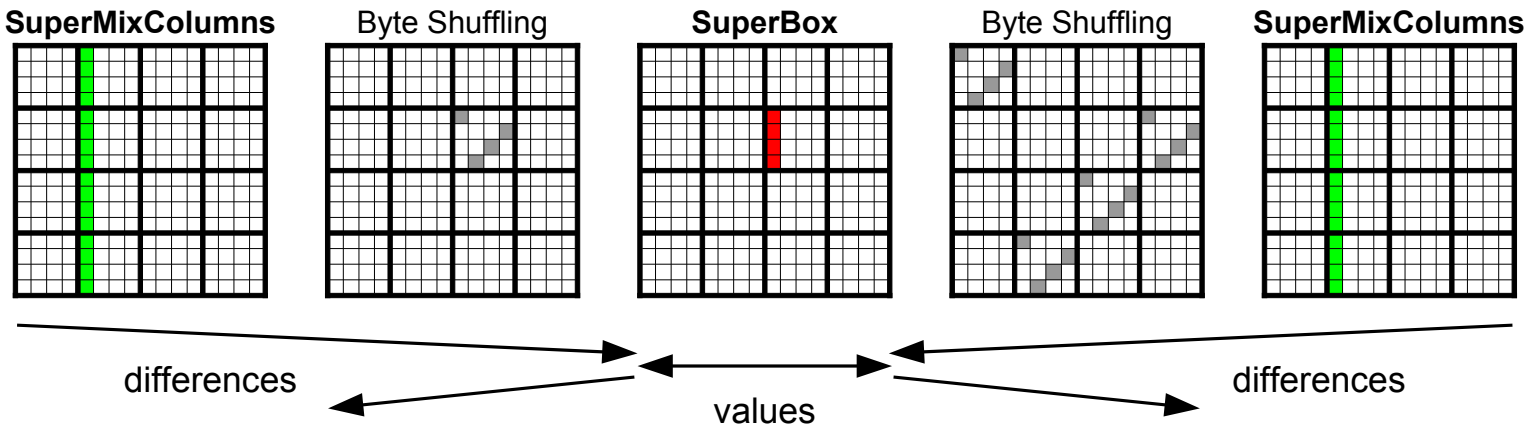


- 10 AES-like round transformations:
  - ShiftRows
  - SubBytes
  - MixColumns
  - AddRoundKey (counter)
  - SubBytes

  ⇐ non-linear SuperBox
  - ShiftRows
  - AddRoundKey (salt)
  - BigShiftRows
  - MixColumns
  - BigMixColumns

  ⇐ linear SuperMixColumns

Rebound Attack on ECHO



- Inbound phase:
  - choosing valid differences for SuperMixColumns
  - matching resulting differences in SuperBoxes
- SuperBox match:
  - to get input values of SuperBoxes (complexity 1)
  - using differential distribution table (DDT) of size 2^{64}
- Rebound Attack with multiple inbound phases:
  - enough freedom for more than one inbound phase
  - merge independent solutions using birthday effect