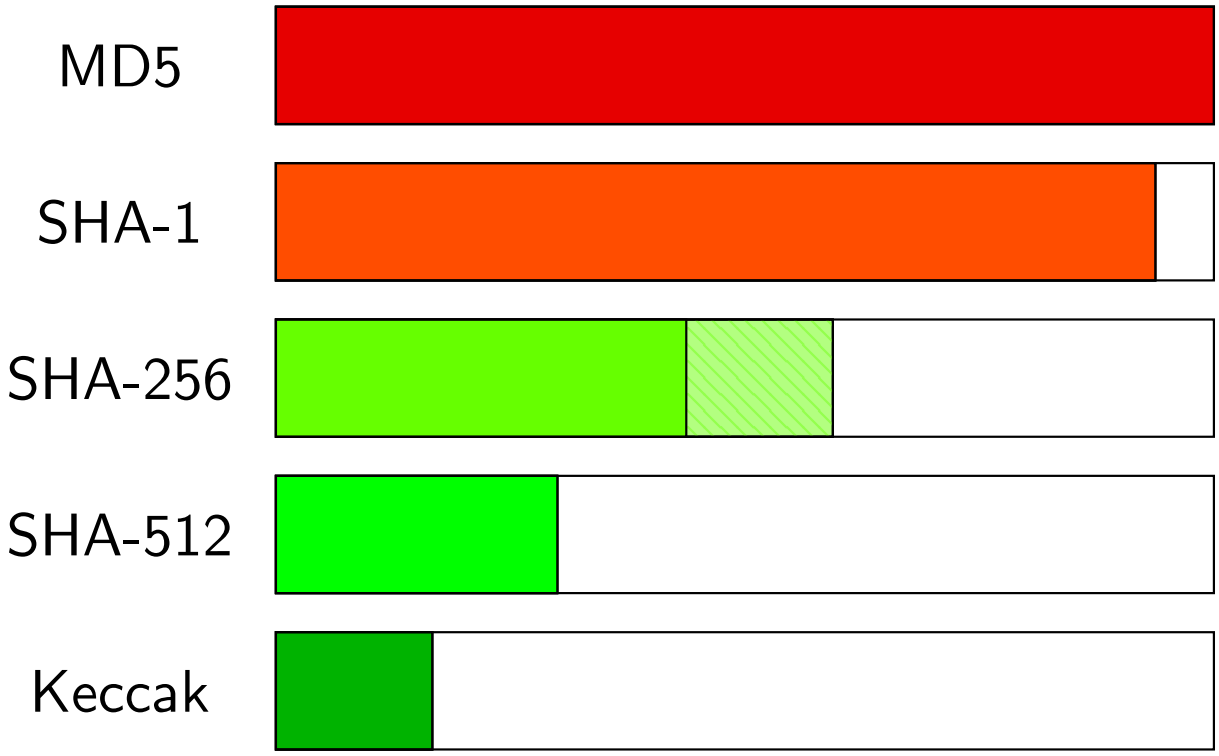
Branching Heuristics in Differential Collision Search: Application to SHA-512

Maria Eichlseder Florian Mendel Martin Schläffer

IAIK, Graz University of Technology, Austria

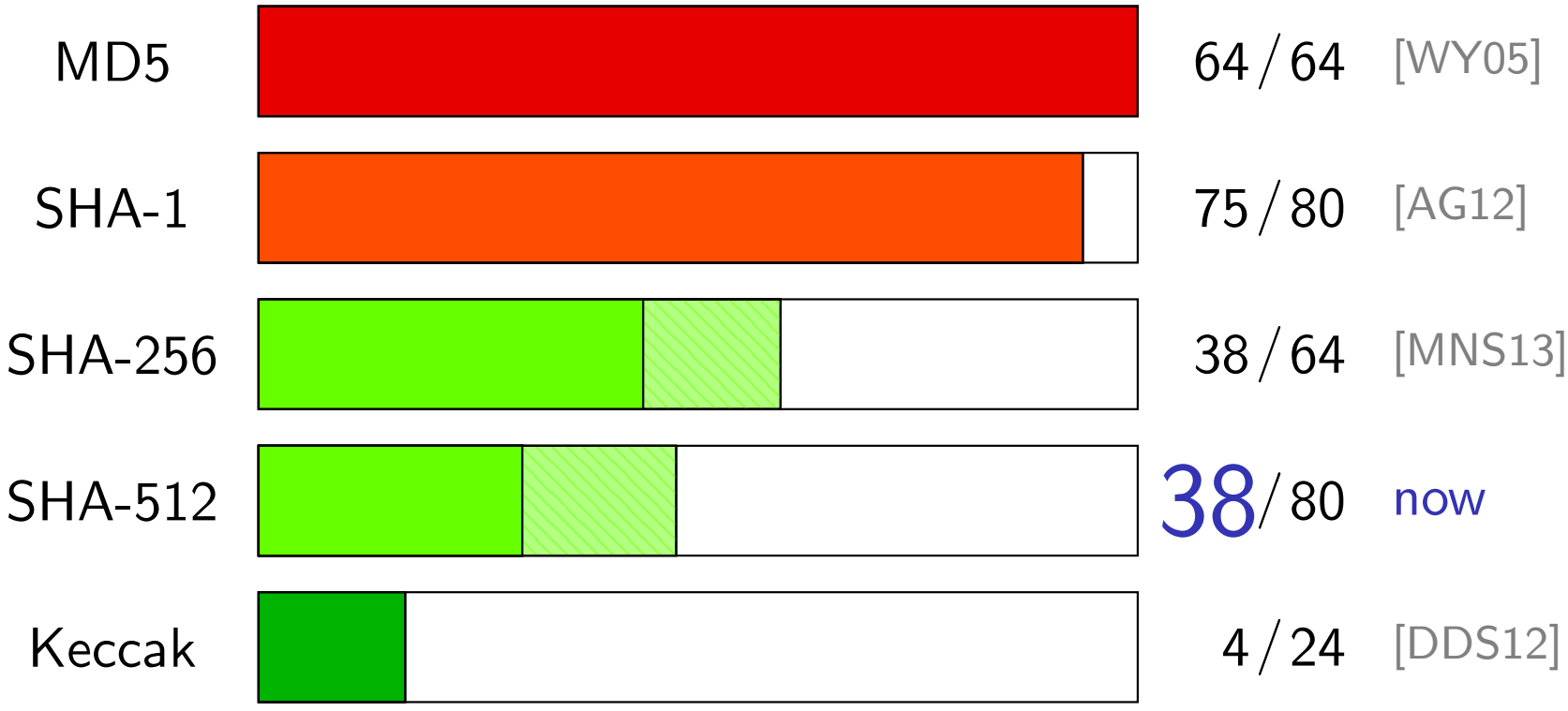
FSE 2014

Practical Collisions for Round-Reduced Hash Functions



Contribution: { semi-free-start collision for 38 steps of SHA-512
using improved automatic search tools

Practical Collisions for Round-Reduced Hash Functions

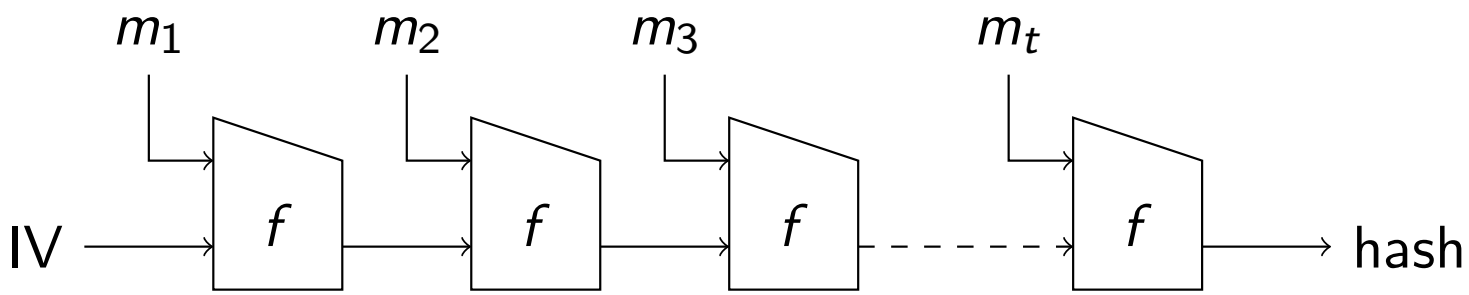


Contribution:
- semi-free-start collision for 38 steps of SHA-512
- using improved automatic search tools

SHA-2 Family – SHA-256 / SHA-512

Iterated hash function

- 32-bit/64-bit words
- 16-word message blocks (= 512/1024 bits)
- 8-word hash value and chaining value (= 256/512 bits)



Compression function f

- Message expansion: expand 16 words M_i to 64/80 words W_i
- State update: 64/80 steps with status words A_i, E_i

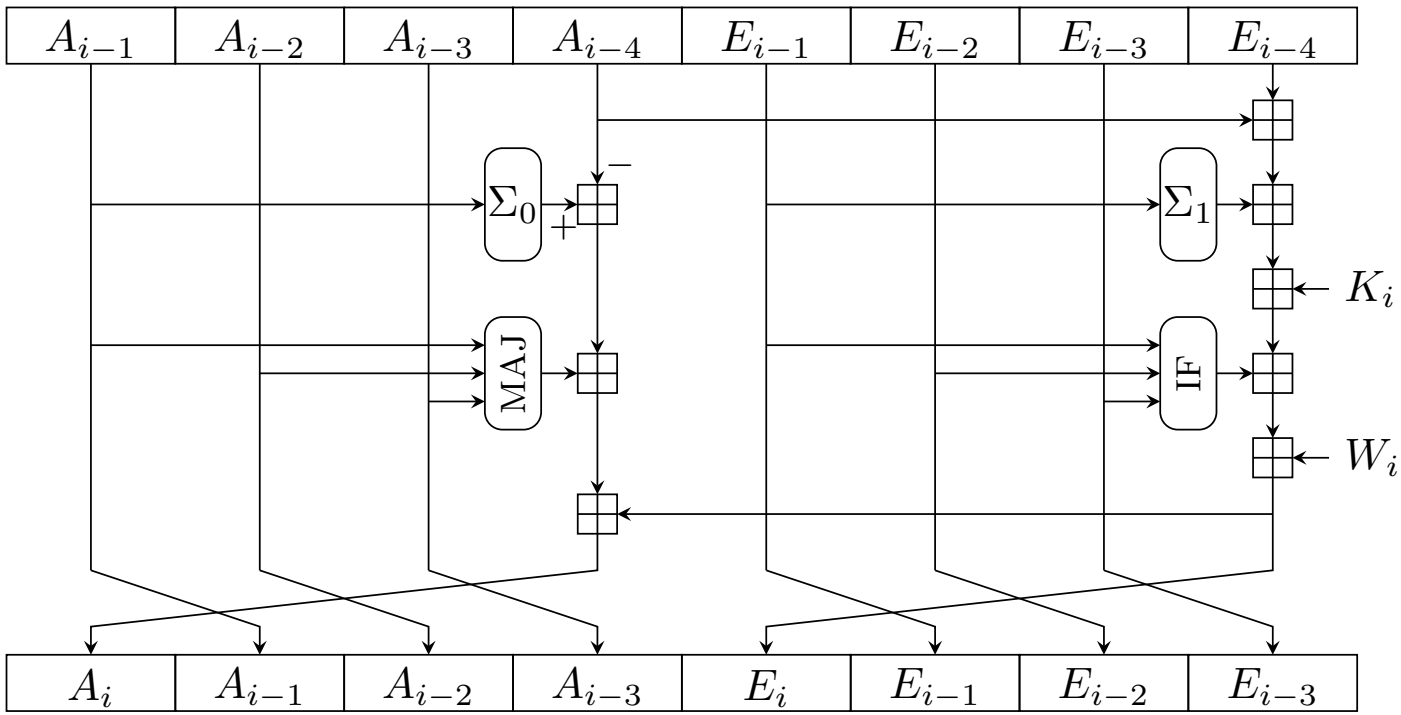
SHA-2 Compression Function

- Message expansion: expand 16 words M_i to 64/80 words W_i

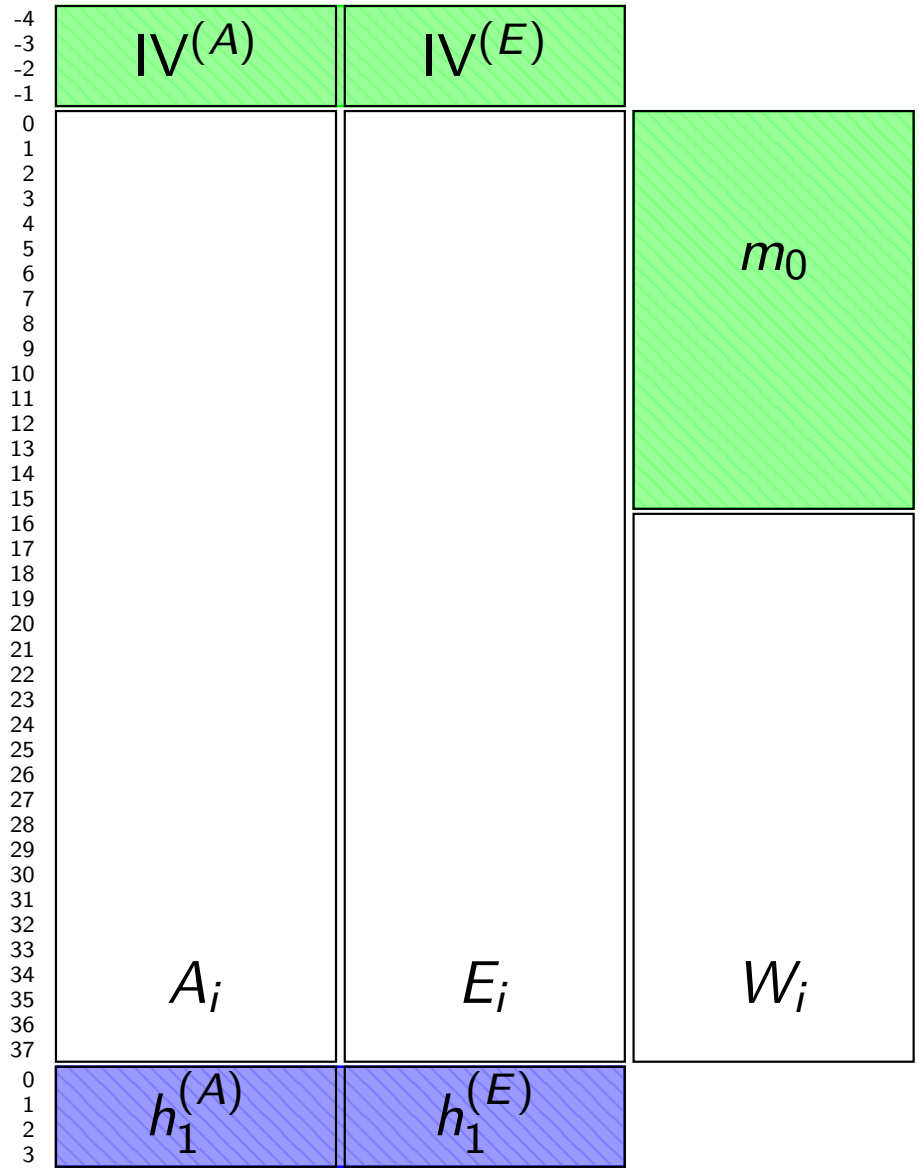
$$W_i = f_W(W_{i-2}, W_{i-7}, W_{i-15}, W_{i-16}) \text{ for } i \geq 16$$

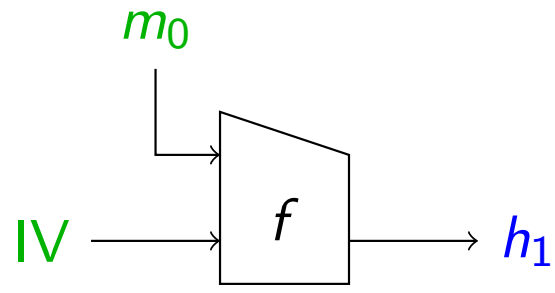
- State update: 64/80 steps with status words A_i, E_i

$$E_i = f_E(A_{i-4}, E_{i-1}, \ldots, E_{i-4}, K_i, W_i),$$
$$A_i = f_A(E_i, A_{i-1}, \ldots, A_{i-4})$$



SHA-2 Compression Function State

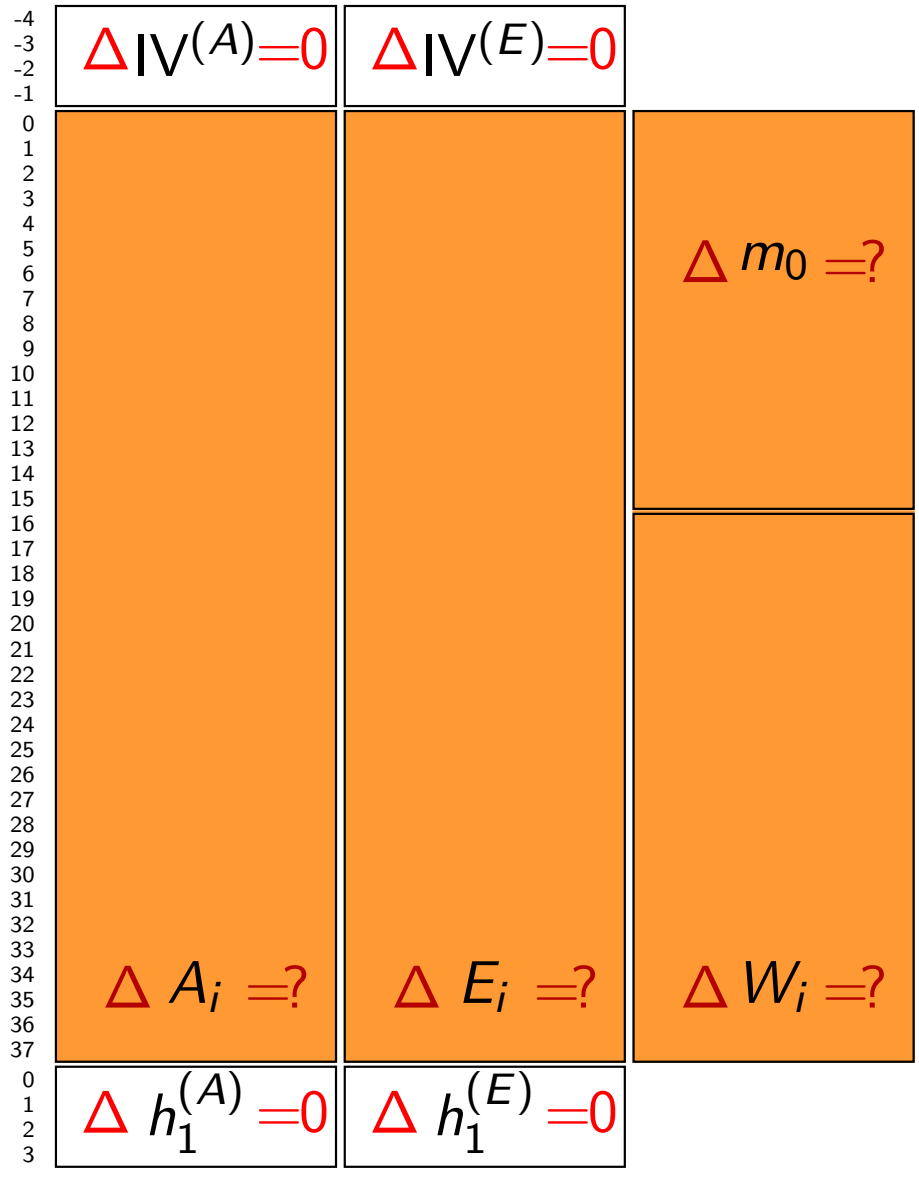




SHA-2 compression function:

- shows state words A_i, E_i, W_i
- inputs IV, m_0
- output h_1

Previous Collision Attack on SHA-256 [MNS13]



Starting point

- Few message words different
- High probability
- Local collisions

Differential characteristic

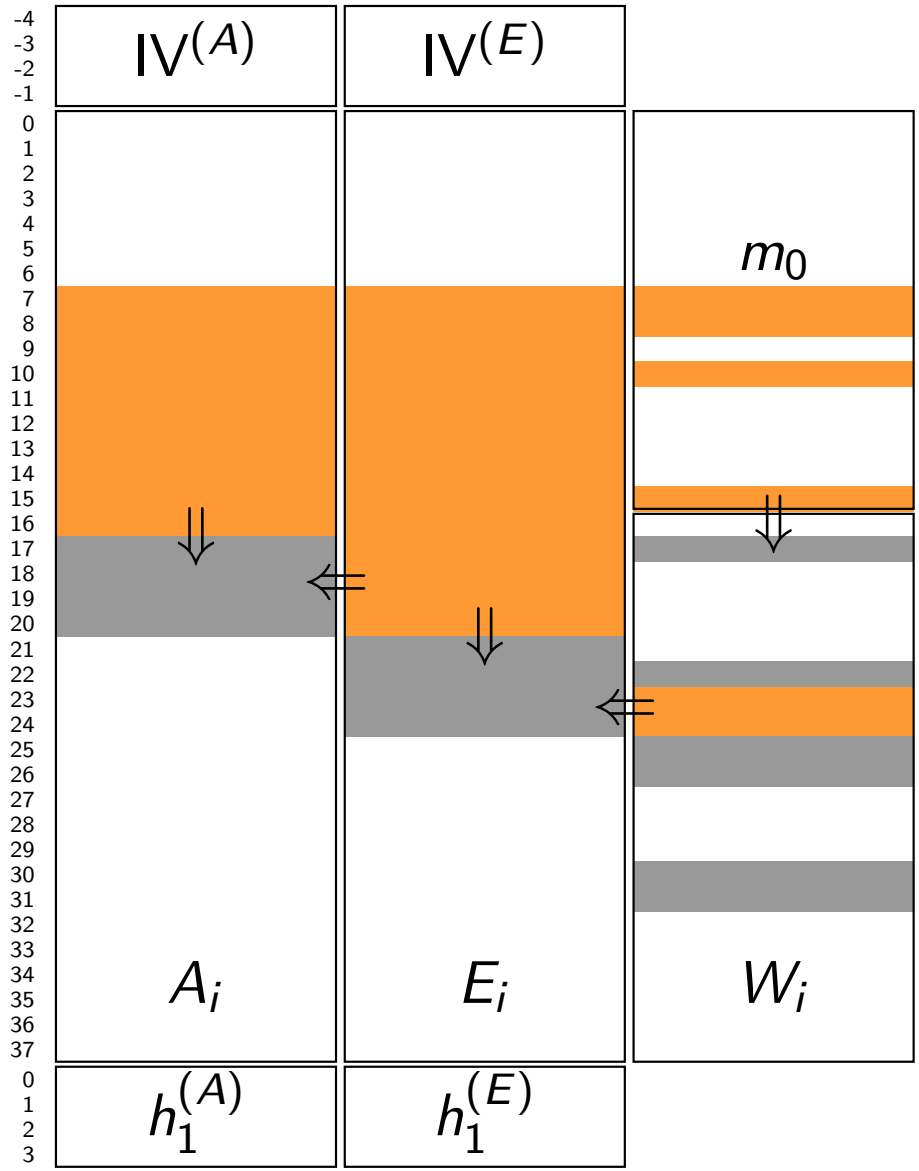
Automated search tool [DR06]

1. Guess undetermined bits
2. Determine consequences
3. Backtrack if contradiction

Message Pair

Automated search tool

Previous Collision Attack on SHA-256 [MNS13]



Starting point

- Few message words different
- High probability
- Local collisions

Differential characteristic

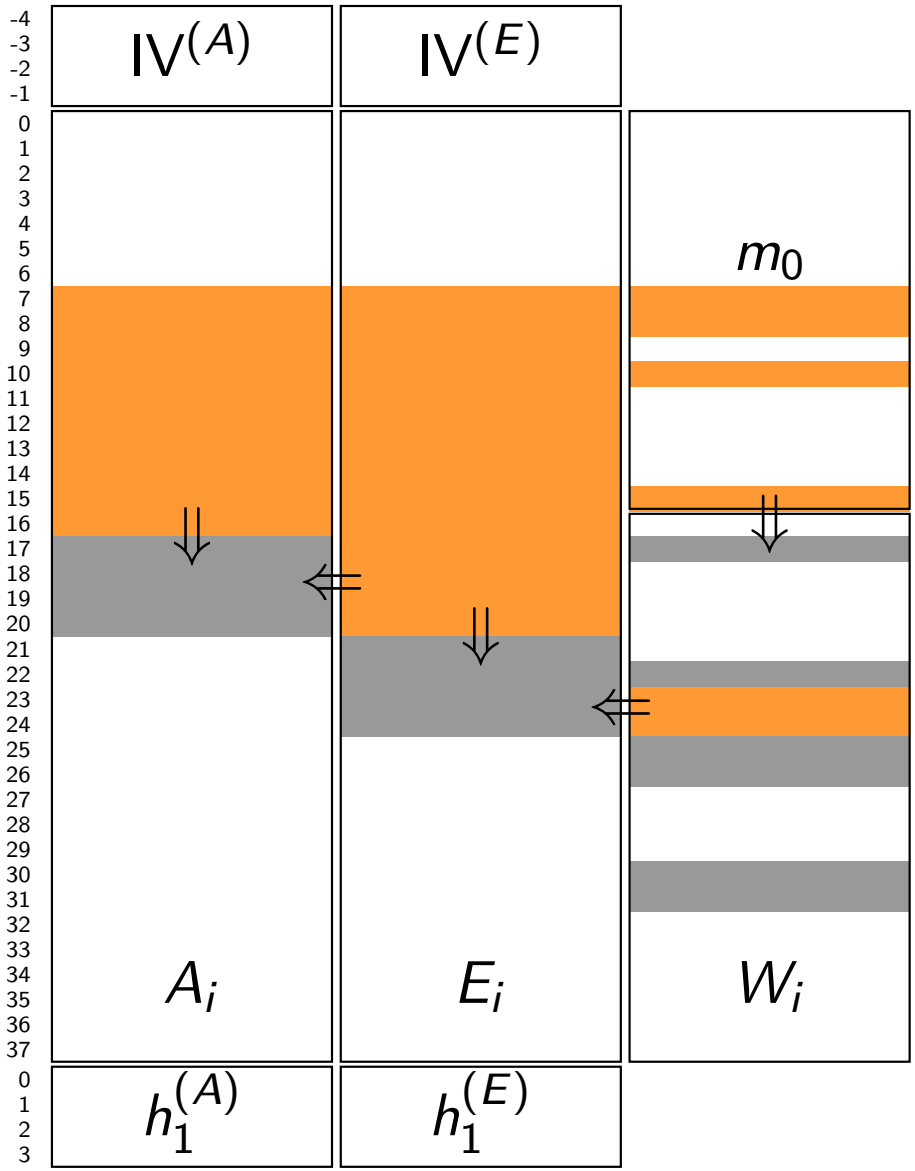
Automated search tool [DR06]

1. Guess undetermined bits
2. Determine consequences
3. Backtrack if contradiction

Message Pair

Automated search tool

Previous Collision Attack on SHA-256 [MNS13]



Starting point

- Few message words different
- High probability
- Local collisions

Differential characteristic

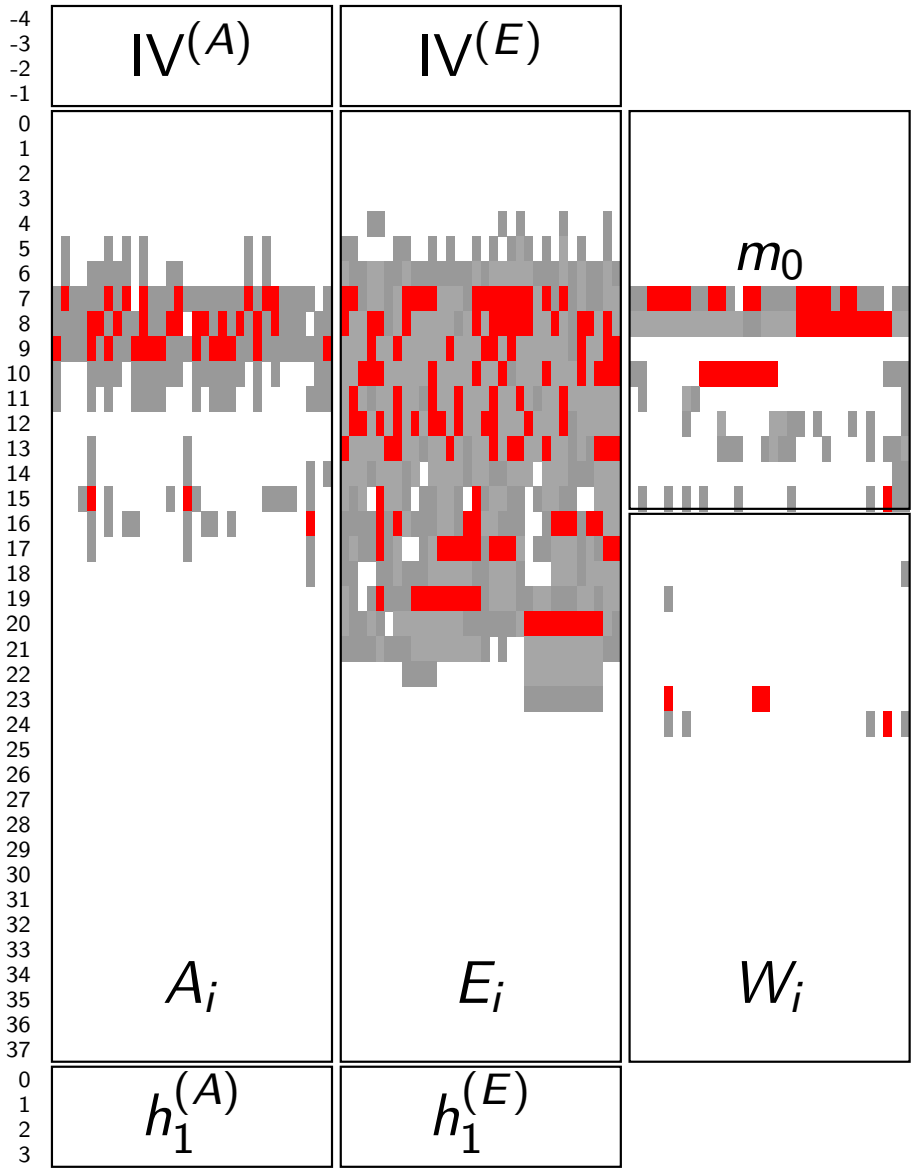
Automated search tool [DR06]

1. Guess undetermined bits
2. Determine consequences
3. Backtrack if contradiction

Message Pair

Automated search tool

Previous Collision Attack on SHA-256 [MNS13]



Starting point

- Few message words different
- High probability
- Local collisions

Differential characteristic

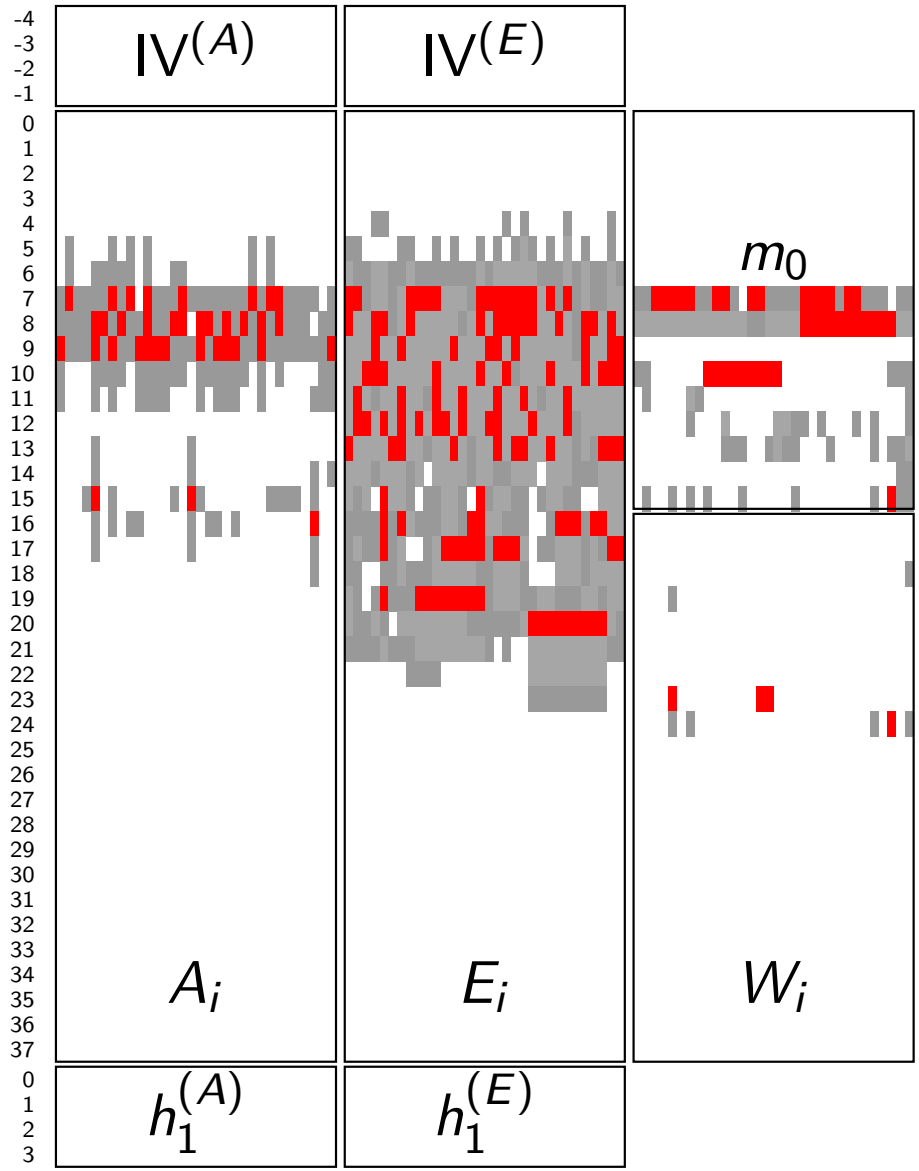
Automated search tool [DR06]

1. Guess undetermined bits
2. Determine consequences
3. Backtrack if contradiction

Message Pair

Automated search tool

Previous Collision Attack on SHA-256 [MNS13]



Starting point

- Few message words different
- High probability
- Local collisions

Differential characteristic

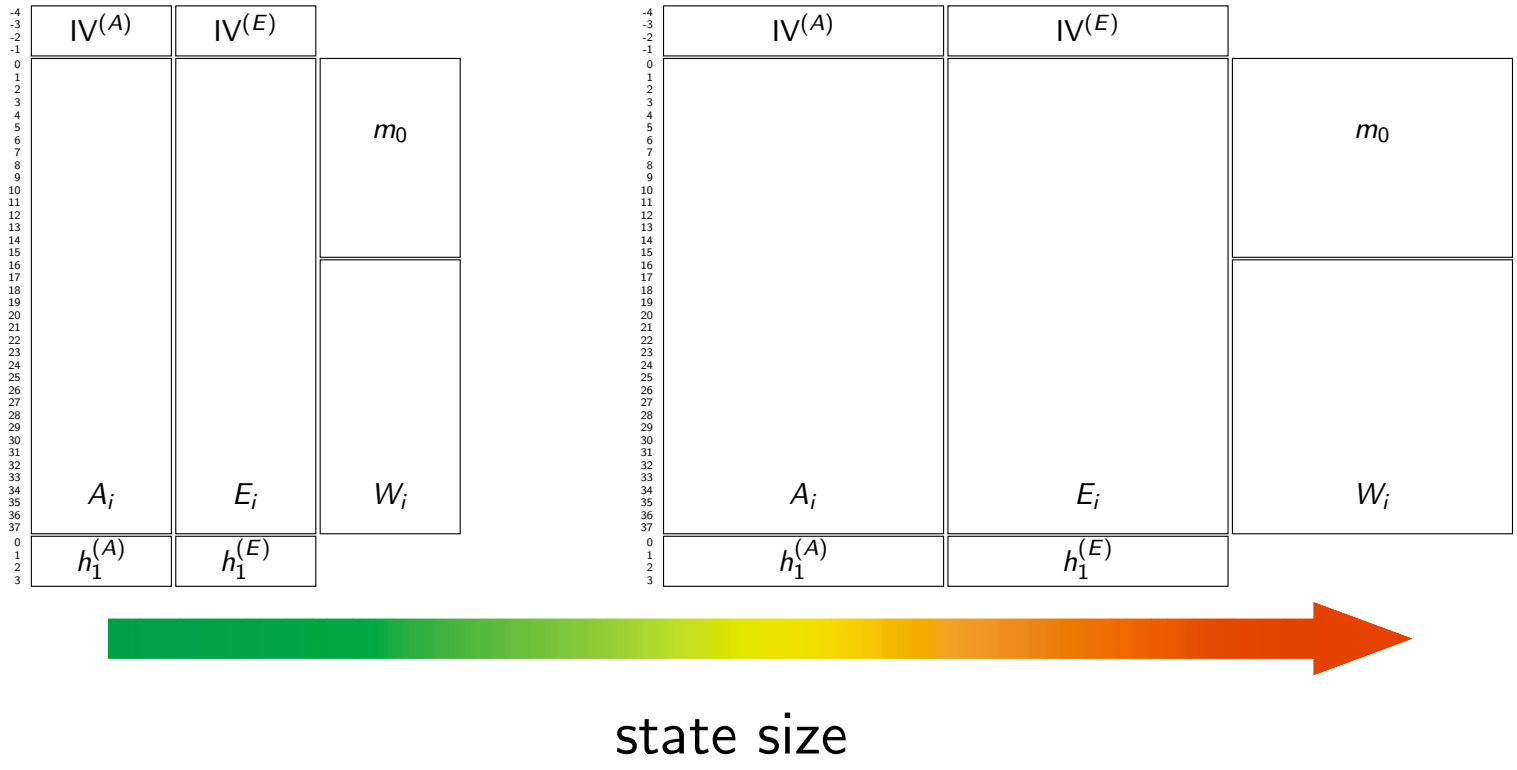
Automated search tool [DR06]

1. Guess undetermined bits
2. Determine consequences
3. Backtrack if contradiction

Message Pair

Automated search tool

Problem – SHA-256 vs. SHA-512



Consequences:

- Larger search space
- Contradictions take longer to detect
- More conditions to fulfill

Improving Guess & Determine?

- Problem description [MNS13]
  - Starting point
  - Hash function description
  - High-level strategy
- Guessing strategy, branching rules [MNS11]
  - Which variable to pick first?
  - Which value to guess first for this variable?
- Propagation [MNS11, EMN+13, Leu12, Leu13]
  - How to detect contradictions?
  - How to determine implications of a guess?
- Backtracking [MNS11]
  - How many guesses to undo?
  - Restart?

Improving Guess & Determine?

- Problem description [MNS13]
  - Starting point
  - Hash function description
  - High-level strategy
- Guessing strategy, branching rules [MNS11]
  - Which variable to pick first?
  - Which value to guess first for this variable?
- Propagation [MNS11, EMN+13, Leu12, Leu13]
  - How to detect contradictions?
  - How to determine implications of a guess?
- Backtracking [MNS11]
  - How many guesses to undo?
  - Restart?

Branching: Inspiration from SAT Solvers...

SAT Solvers (Guess-and-Determine for CNF formulas)

Different strategies and paradigms:

- Many small clauses first (Böhm, MOM, JW)
- Many clauses first (DLCS, DLIS)

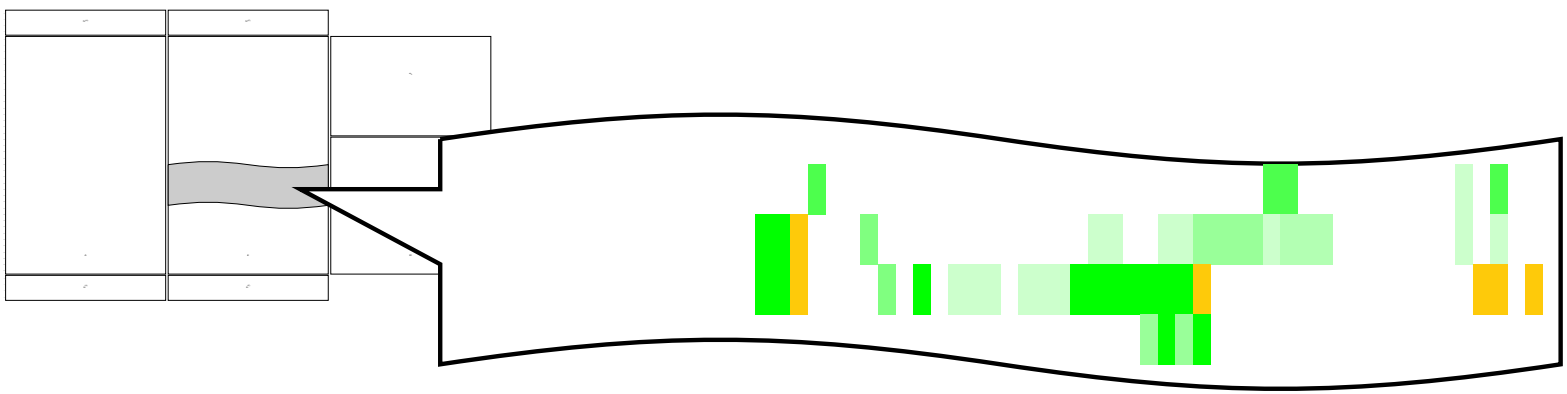
- Conflict-driven, recent conflicts first (VSIDS)
- Localized, recently updated clauses first

- Preview consequences (UPLA)

Look-Ahead Branching Heuristic

Rationale:

- Propagation is good
  - Reduce search space
  - Better explicit than implicit conditions
- Contradictions are good
  - Better handle them sooner rather than later



$\Rightarrow$ simulate outcome for candidate guessing variables and pick best

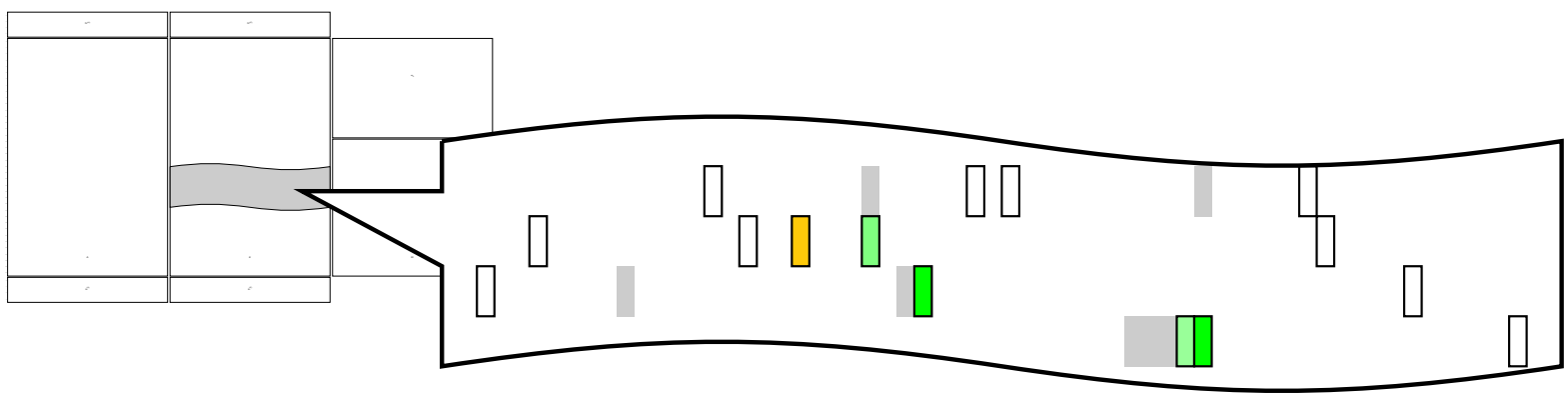
Randomized Look-Ahead

Problems of basic approach:

- Simulating for many candidates is very costly
- Search is not well randomized – essential after restarts

Solution:

- Limit absolute candidate set size
- Limit relative set size
- Avoid redundant evaluation of candidates



Effect of Branching Heuristic (16 Candidates)

Semi-free-start collisions:

27 or 38 steps of SHA-256

- with heuristic: about 5–50 times faster

27 steps of SHA-512

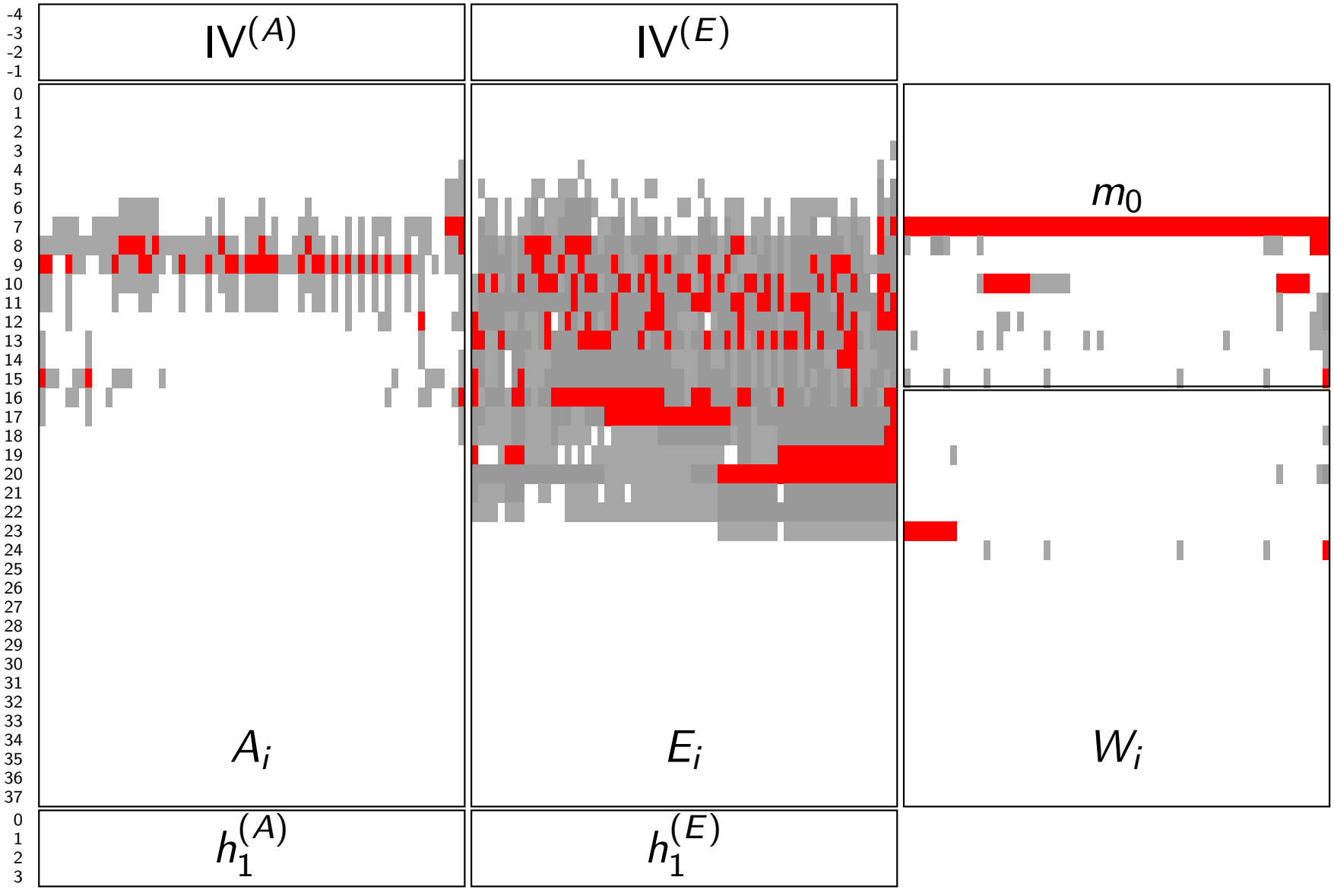
- without heuristic: 4 days on 40 CPUs
- with heuristic: seconds on standard PC

38 steps of SHA-512

- without heuristic: no results
- with heuristic: $\approx 1.5\,\mathrm{h}$ on 40 CPUs

Collisions with correct IV: not enough freedom in message left

Application to 38 steps of SHA-512 – Characteristic



Application to 38 steps of SHA-512 – Result

Semi-free-start collision for 38 of 80 steps ($\approx 1.5\,\mathrm{h}$ on 40 CPUs):

h_0	e8626f53a3771964	2ae427b8c5065790	c8fd5a1628fc3337	0f362d297f82f987
	89166a0c022ffc40	c2c49c30e629239f	d1fa8bd692843025	ad4bba64c797e6ec
m	610519a88f0d2809	3addc83f01c8b179	84afa7a2772c6141	ad539854e64c9cce
	85450b73549b2085	7296b5291f31c0d9	fc978d9624e2c2cc	fffffffffffffffe
	92114cb9d2f4cd9b	34a3198b79871212	cca7f43154e38081	ac0598a589168fe1
	f32ae6a0070a8d2e	755aa5cada87e894	4b9bd7df3c94b667	65291f2b80cc8c51
m^*	610519a88f0d2809	3addc83f01c8b179	84afa7a2772c6141	ad539854e64c9cce
	85450b73549b2085	7296b5291f31c0d9	fc978d9624e2c2cc	0000000000000001
	92114cb9d2f4cd9c	34a3198b79871212	cca8143154e38079	ac0598a589168fe1
	f32ae6a0070a8d2e	755aa5cada87e894	4b9bd7df3c94b667	65291f2b80cc8c50
Δm	0000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000	0000000000000000	ffffffffffffffff
	0000000000000007	0000000000000000	000fe000000000f8	0000000000000000
	0000000000000000	0000000000000000	0000000000000000	0000000000000001
h_1	946a28eedc3b2ff6	c4573d0a13ea6268	11f07b04b06900dd	897c606e4053bbe4
	2406aae9d58504b4	89b237932b061ba8	663402cb4bb1972c	d99c062dce945423

Conclusion

SHA-512

- Larger state size is a problem for automated tools
- Requires better branching strategy to apply SHA-256 attacks
- Semi-free-start collision on 38 steps

Look-ahead branching heuristic

- To navigate through larger search spaces
- Evaluates randomly selected candidates
- Number of candidates and randomness critical

Future

- Extend to hash collision with fixed IV?
- Other SAT Solver techniques?

Bibliography I

Andrew V. Adinetz and Evgeny A. Grechnikov.

Building a collision for 75-round reduced SHA-1 using GPU clusters.
In Christos Kaklamanis, Theodore S. Papatheodorou, and Paul G. Spirakis, editors, *Euro-Par*, volume 7484 of *Lecture Notes in Computer Science*, pages 933–944. Springer, 2012.

Itai Dinur, Orr Dunkelman, and Adi Shamir.

New attacks on Keccak-224 and Keccak-256.
In Anne Canteaut, editor, *FSE*, volume 7549 of *Lecture Notes in Computer Science*, pages 442–461. Springer, 2012.

Christophe De Cannière and Christian Rechberger.

Finding SHA-1 characteristics: General results and applications.
In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006.

Maria Eichlseder, Florian Mendel, Tomislav Nad, Vincent Rijmen, and Martin Schläffer.

Linear propagation in efficient guess-and-determine attacks.
In Lilya Budaghyan, Tor Helleseth, and Matthew G. Parker, editors, *WCC*, 2013.
`http://www.selmer.uib.no/WCC2013/`.

Sebastiaan Indesteege, Florian Mendel, Bart Preneel, and Christian Rechberger.

Collisions and other non-random properties for step-reduced SHA-256.
In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *LNCS*, pages 276–293. Springer, 2008.

Gaëtan Leurent.

Analysis of differential attacks in ARX constructions.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *LNCS*, pages 226–243. Springer, 2012.

Bibliography II

Gaëtan Leurent.

Construction of differential characteristics in ARX designs: Application to Skein.
In Ran Canetti and Juan A. Garay, editors, *CRYPTO (1)*, volume 8042 of *LNCS*, pages 241–258. Springer, 2013.

Florian Mendel, Tomislav Nad, and Martin Schläffer.

Finding SHA-2 characteristics: Searching through a minefield of contradictions.
In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 288–307. Springer, 2011.

Florian Mendel, Tomislav Nad, and Martin Schläffer.

Improving local collisions: New attacks on reduced SHA-256.
In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 262–278. Springer, 2013.

Somitra Kumar Sanadhya and Palash Sarkar.

New collision attacks against up to 24-step SHA-2.
In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT*, volume 5365 of *LNCS*, pages 91–103. Springer, 2008.

Xiaoyun Wang and Hongbo Yu.

How to break MD5 and other hash functions.
In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.