Improved MITM Cryptanalysis on Streebog

Jialiang Hua, Xiaoyang Dong, Siwei Sun, Zhiyu Zhang,
Lei Hu, Xiaoyun Wang

Speaker: Jialiang Hua

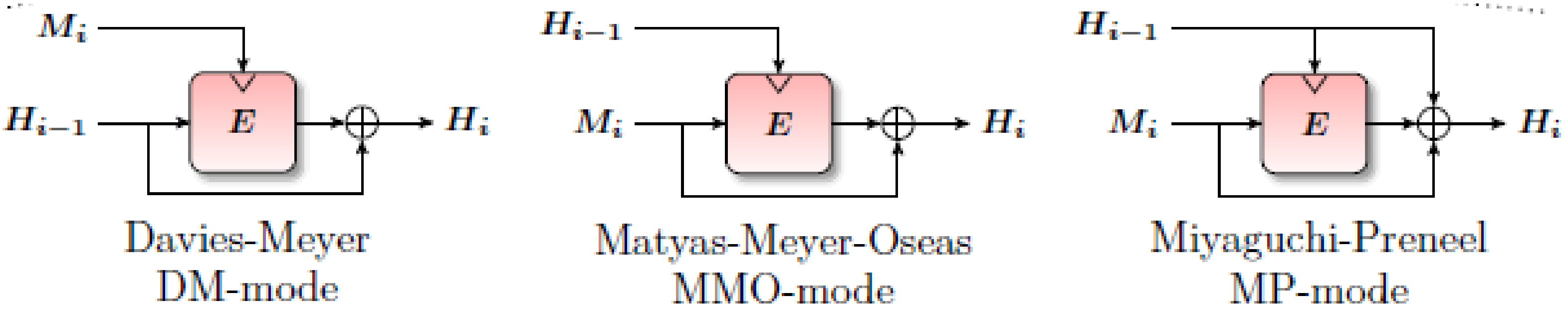
Email: huajl18@mails.tsinghua.edu.cn

2023/03/22

1. Background
2. Improved MILP model
3. Attack on Streebog

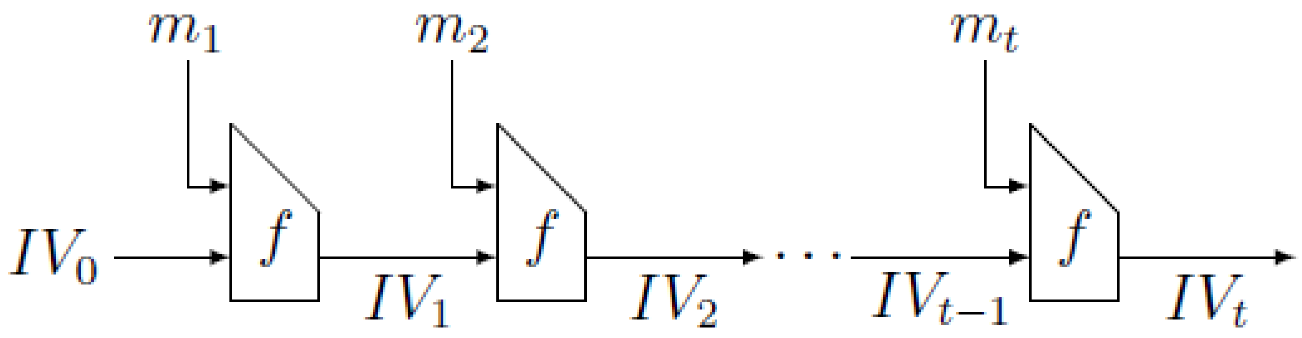
1 Background

- **Hash Function:** maps a message of arbitrary length into a short fixed length digest.
  - 1. Preimage Resistance 2. Second-Preimage Resistance 3. Collision Resistence
- Construction of Hash function: 1.Compression function 2.Domain extender
  - The compression function：
    - Based on block ciphers, there are 12 secure PGV modes[C:PGV93]. e.g. Streebog.

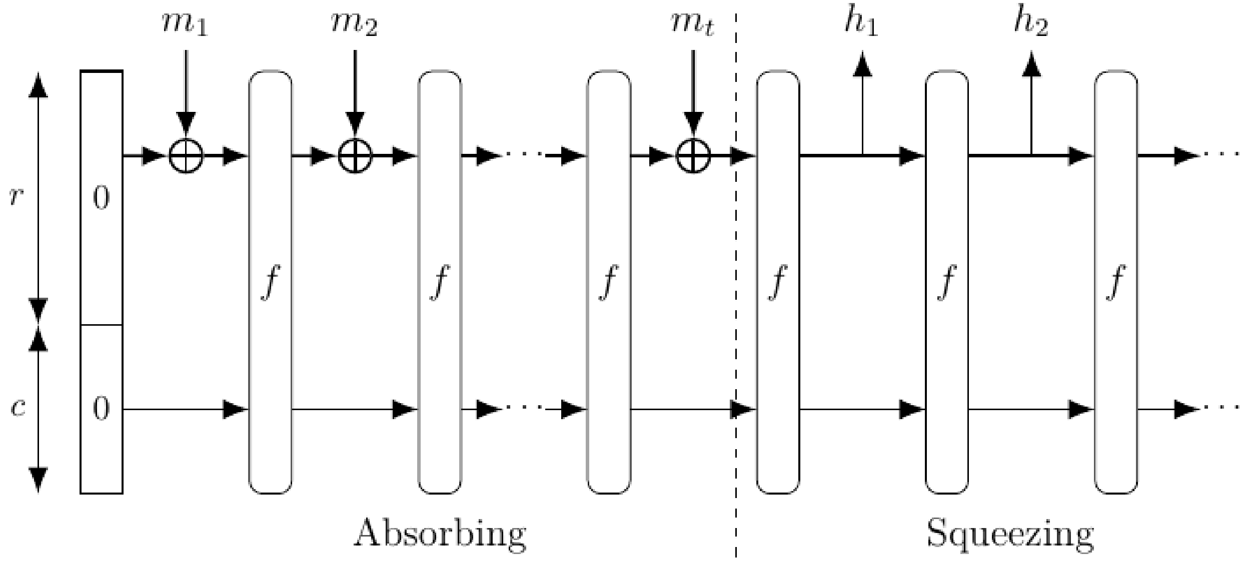


    - Directly construct. e.g. MD5, SHA-1, SHA-2.
    - Based on hard mathematic problems.

- Domain extender：
  - Merkel-Damgård（MD）structure

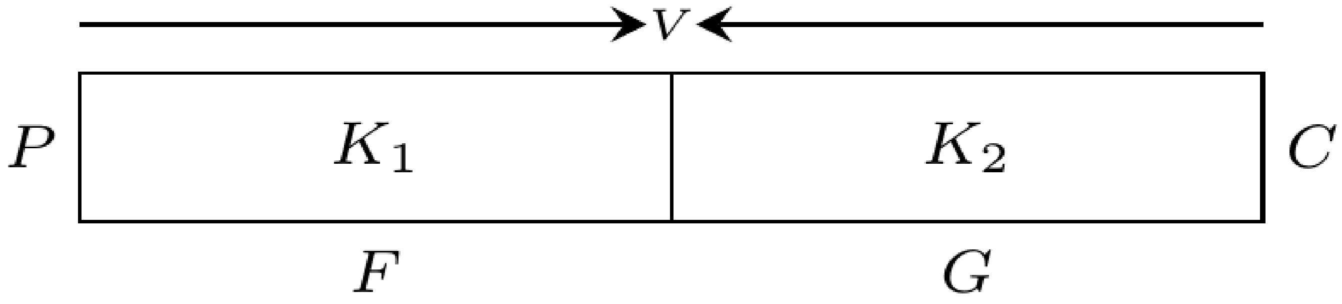


  - Sponge structure

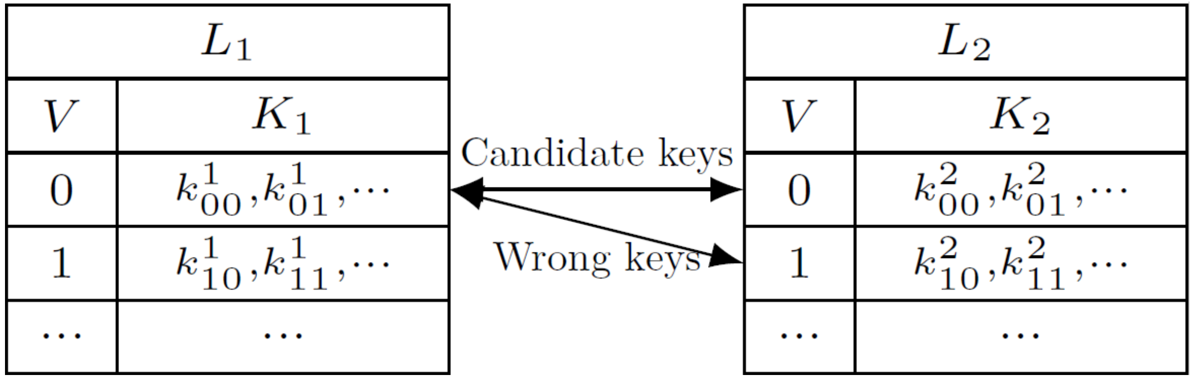


Meet-in-the-Middle(MITM) attack [DH77]

- Meet-in-the-Middle (MITM): introduced by Diffie and Hellman[DH77] in 1977.
- Encryption: $C = G_{K_1}(F_{K_2}(P))$, n-bit block size.



◆ Correct guess of K_1 and K_2 : $F_{K_1}(P)$ =V= $G_{K_2}^{-1}(C)$

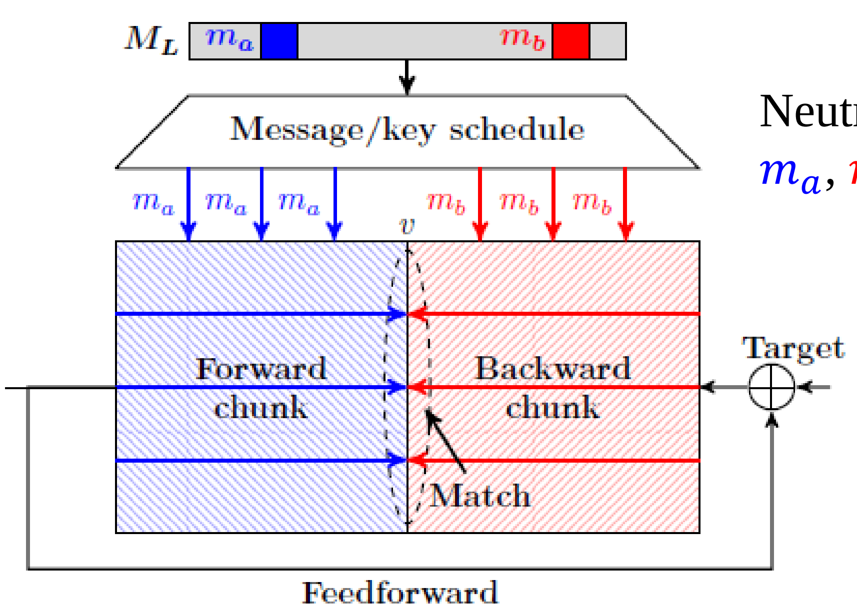


Key space: From $2^{|K_1|+|K_2|}$ to $2^{|K_1|+|K_2|-n}$

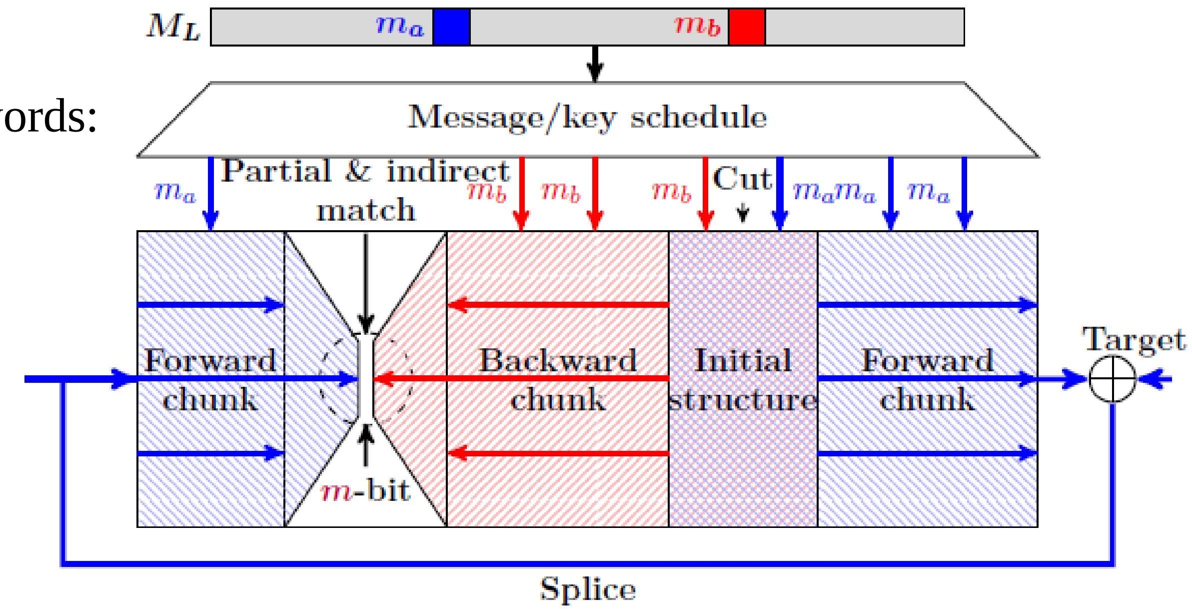
MITM Preimage attack on hash functions

- Sasaki and Aoki [SA08] introduced the Meet-in-the-Middle (MITM) preimage attack in 2008.
- MITM Preimage Attack is applied to many Hash functions.
  - MD4 [AC:GLRW10]
  - MD5 [EC:SasAok09]
  - Tiger [AC:GLRW10]
  - HAVAL [AC:SasAok08]
  - …
  - SHA-1 [C:KneKho12]
  - SHA-2 [AC:GLRW10]
  - Whirlpool [AC:SWWW12]
  - Grostl [IWSEC:MLHL15]

MITM Preimage attack on compression function



Neutral words:
m_a, m_b

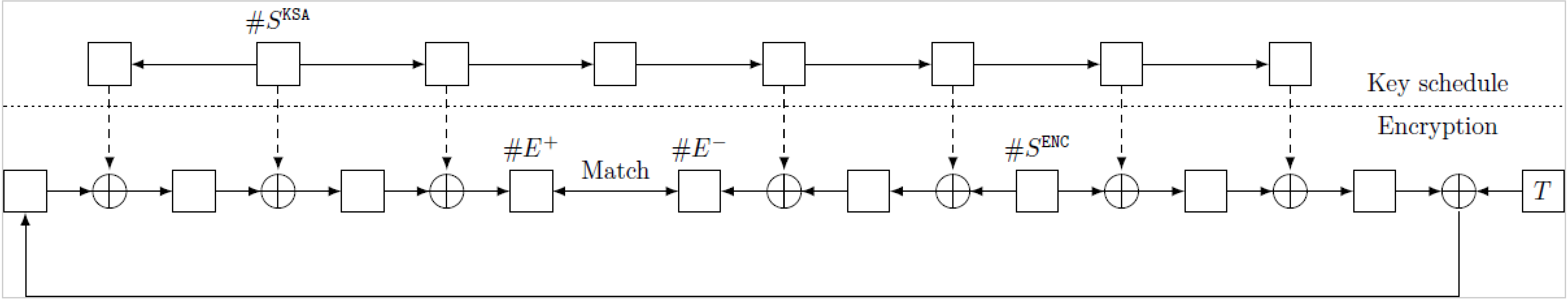


- For $2^{n-(d_1+d_2)}$ values of $M_L / \{m_a, m_b\}$
  - For 2^{d_1} values of m_a, forward compute to get a list $\overrightarrow{\mathcal{L}}$ of v.
  - For 2^{d_2} values of m_b, backward compute to get a list $\overleftarrow{\mathcal{L}}$ of v.
  - If find a match between $\overrightarrow{\mathcal{L}}$ and $\overleftarrow{\mathcal{L}}$, return the correspondence M_L.

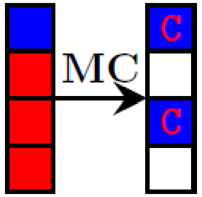
- Splice-and-cut: better chunk separations
- Initial structure: more rounds
  - neutral words appear simultaneously
  - local-collision-like cancellation of impact
- Partial & indirect matching: more rounds
  - filtering using partial state ($m < n$ bits)
  - indirect matching via linear relations.

Time complexity: $2^{n-(d_1+d_2)} \cdot (2^{\max(d_1,d_2)} + 2^{d_1+d_2-m}) \simeq 2^{n-\min(d_1,d_2,m)}$.

Automatic search of MITM preimage attacks with MILP [EC:BDGLSSW21]



- Select neutral words from both encryption and key states for both chunks.
- Apply the essential idea behind initial structure to every possible round. e.g. Add constraints on neutral words to cancel impact in every round.



- Starting states: S^{ENC} and S^{KSA}
- Ending states: E^+ and E^-
- For each combinations of total round, starting states and ending states, build an individual MILP model and solve.

MILP model for MITM preimage attack [EC:BDGLSSW21]

- Each cell of state S : encoded by a pair of 0-1 variables (x_i^S, y_i^S).

$(x_i^S, y_i^S) =$
- (1,1) , Gray , computable in both chunks
- (1,0) , Blue , computable only in forward chunk
- (0,1) , Red , computable only in backward chunk
- (0,0) , White , incomputable in both chunks

Constraints for the Starting States.

$\alpha_i = 1$ if and only if $(x_i^S, y_i^S) = (1,0)$ ⇒

$$\begin{cases} x_i^S - \alpha_i \geq 0 \\ y_i^S - x_i^S + \alpha_i \geq 0 \\ y_i^S + \alpha_i \leq 1 \end{cases}$$

$\beta_i = 1$ if and only if $(x_i^S, y_i^S) = (0,1)$ ⇒

$$\begin{cases} y_i^S - \beta_i \geq 0 \\ x_i^S - y_i^S + \beta_i \geq 0 \\ x_i^S + \beta_i \leq 1 \end{cases}$$

⇒

- initial degrees of freedom(DoF) of Blue cells:

$$\lambda^+ = \sum_i \alpha_i$$

- initial DoF of Red cells:

$$\lambda^- = \sum_i \beta_i$$

- **Constraints for the states in computation paths**
  - σ^{+}: accumulated of consumed DoF in the backward computation
  - σ^{-}: accumulated of consumed DoF in the forward computation

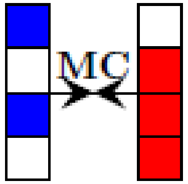
$$d_1 = \lambda^{+} - \sigma^{+}$$

$$d_2 = \lambda^{-} - \sigma^{-}$$

- **Degree of Match(DoM): *m***

m_i: DoM of each pair of rows of E^{+} and E^{-}

$$m = \sum m_i$$



- **Objective Function**

Time complexity: $2^{n - min(d_1, d_2, m)}$

$$\begin{cases} V_{obj} \le d_1 \\ V_{obj} \le d_2 \\ V_{obj} \le m \end{cases}$$

Objective: Maximize V_{obj}

MILP model for MITM preimage attack [EC:BDGLSSW21]

- Constraints for the States in the Computation Path.

Translate the rules of attribute propagation into MILP: e.g., XOR-RULE

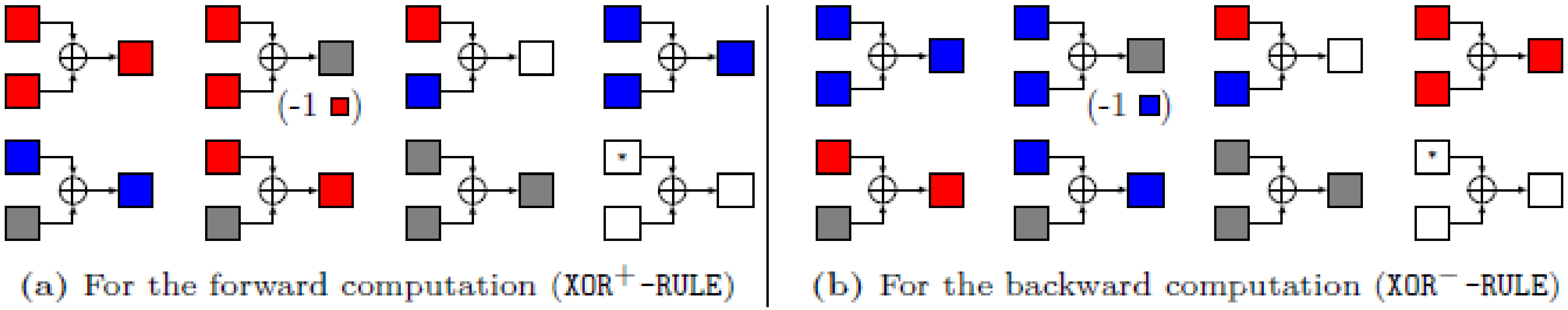
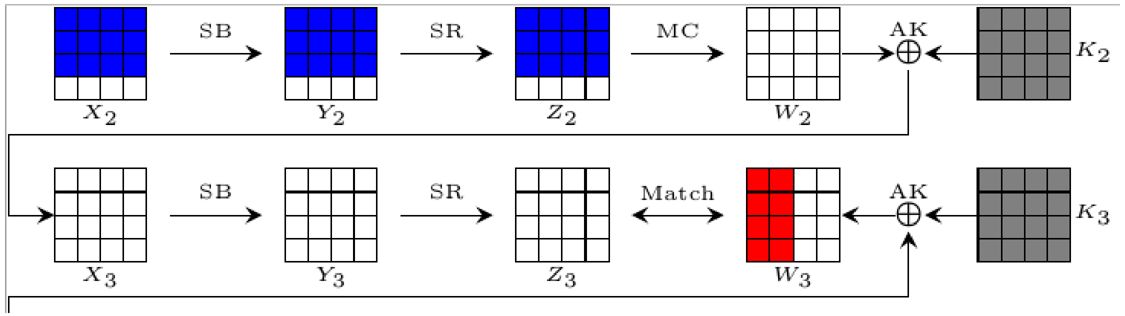


Fig. 3: Rules for XOR operations, where a "*" means that the cell can be any color

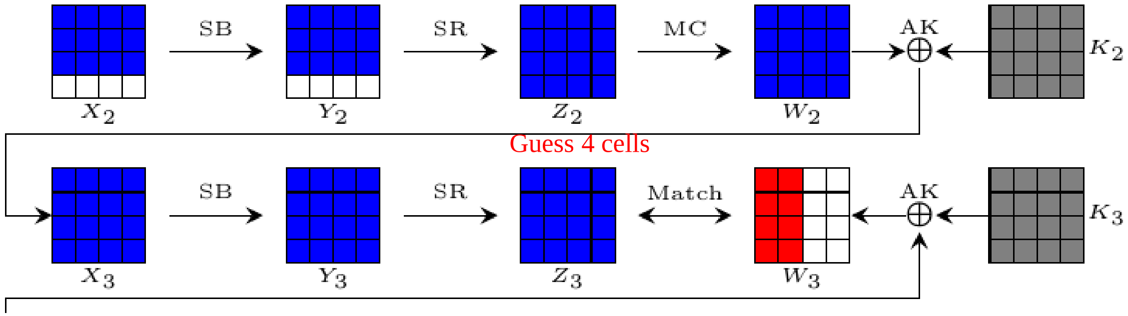
- Let A,B be the input cells and C be the output cell.
- The set rules $\mathrm{XOR}^+ - \mathrm{RULE}$ restricts $(x^A, y^A, x^B, y^B, x^C, y^C, d)$ to a subset of $\mathbb{F}_2^7$, which can be described by a system of linear inequalities.

[1]Bao, Z., Dong, X., Guo, J., Li, Z., Shi, D., Sun, S., & Wang, X. (2021). Automatic search of meet-in-the-middle preimage attacks on AES-like hashing. In *Advances in Cryptology–EUROCRYPT 2021. Part I 40* (pp. 771-804).

2 Improved MILP model



AES Hashing



AES Hashing with guess and determine

- Guess values of a few unknown cells to continue the computation;
- After matching, check the consistency of the guessed cells.

For Gray cells:
    For Blue cells in V:
        For Guessed cells:
            Compute forward...
    For Red cells in U:
        Compute backward find matching
        Check if the guessed cells is correct.

Guess-and-Determine [AC: SWWW12]

MITM preimage attack with guess-and-determine

For Gray cells:
- For Blue cells in V: (2^{d_1})
  - For Guessed cells: (2^{d_b})
    - Compute forward...
- For Red cells in U: (2^{d_2})
  - For Guessed cells: (2^{d_r})
    - Compute backward find matching (2^m)
    - Check the guessed cells.

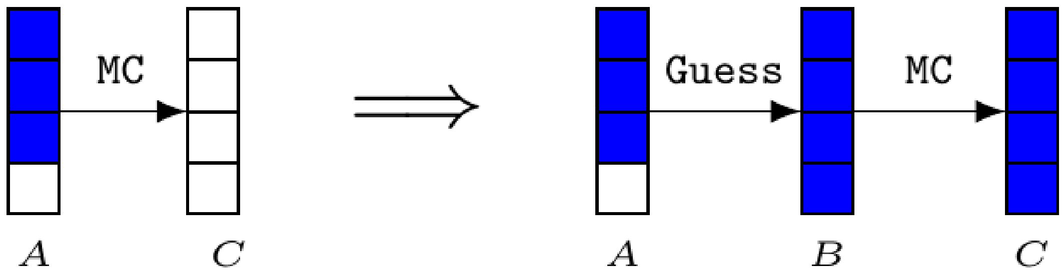
- Test $2^{d_1+d_b+d_2+d_r}$ messages
- Partial matching: $2^{d_1+d_b+d_2+d_r-m}$
- Probability of a correct guess : $2^{-(d_b+d_r)}$
- Valid partial matching: $2^{d_1+d_2-m}$
- Find full match need to Repeat $2^{n-(d_1+d_2)}$ times

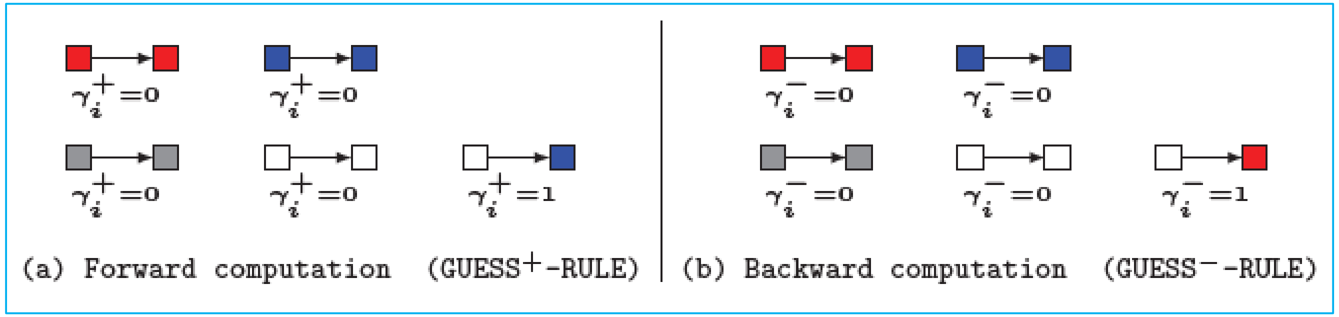
Time complexity:

$$2^{n-(d_1+d_2)} \cdot (2^{d_1+d_b} + 2^{d_2+d_r} + 2^{d_1+d_b+d_2+d_r-m}) \approx 2^{n-min(d_1-d_r, d_2-d_b, m-d_r-d_b)}$$

Model the Guess-and-Determine in the MILP

- New state: **B,** between A and C.
- New Operation: **Guess**





- $Guess^+ - RULE$ restricts $(x^A, y^A, x^B, y^B, \gamma_i^+)$, to a subset of $\mathbb{F}_2^5$, which can be described by a system of linear inequalities.

$$d_b = \sum \gamma_i^+, d_r = \sum \gamma_i^-$$

- **Objective Function**

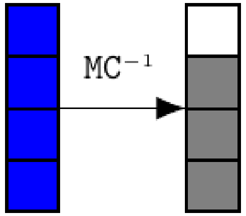
Time complexity: $2^{n-min(d_1-d_r, d_2-d_b, m-d_r-d_b)}$

$$\begin{cases} V_{obj} \le d_1 - d_r \\ V_{obj} \le d_2 - d_b \\ V_{obj} \le m - d_r - d_b \end{cases}$$

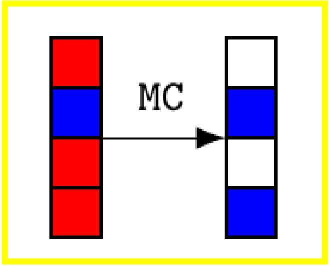
Objective: Maximize V_{obj}

Nonlinear Constrained Neutral Words [C:DHALWH21]

- **Neutral Words are linearly constrained**



$$MC^{-1}\cdot\begin{bmatrix}B_1\\B_2\\B_3\\B_4\end{bmatrix}=\begin{bmatrix}c_3\\c_4\\c_5\\c_6\end{bmatrix}$$



$$MC\cdot\begin{bmatrix}R_1\\0\\R_2\\R_3\end{bmatrix}=\begin{bmatrix}-\\c_1\\-\\c_2\end{bmatrix}$$

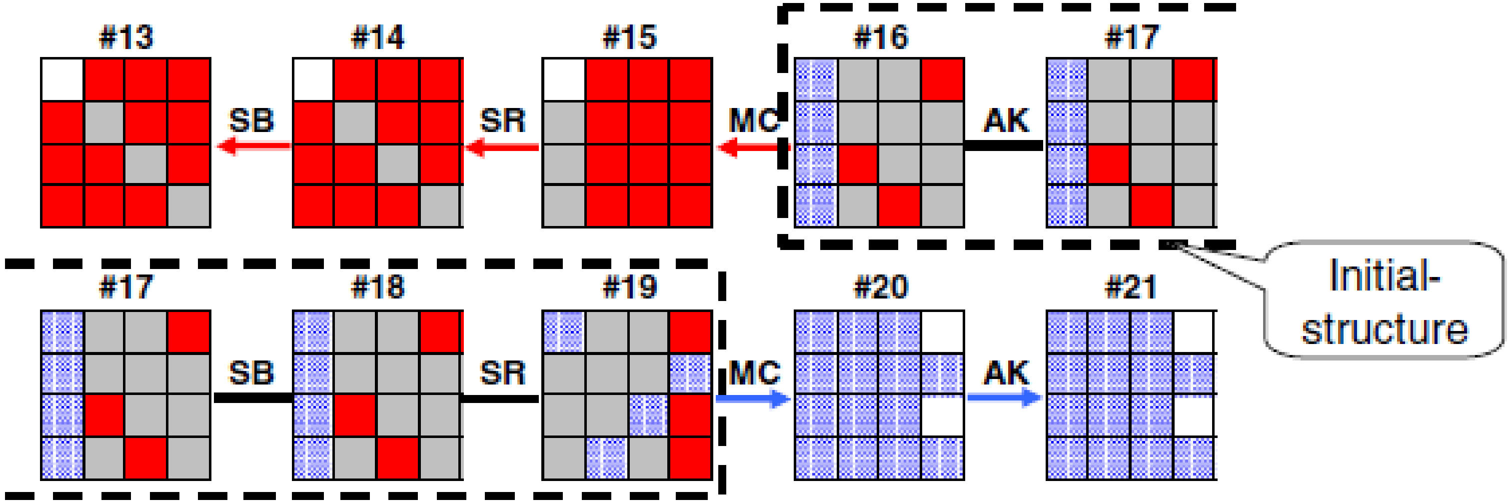
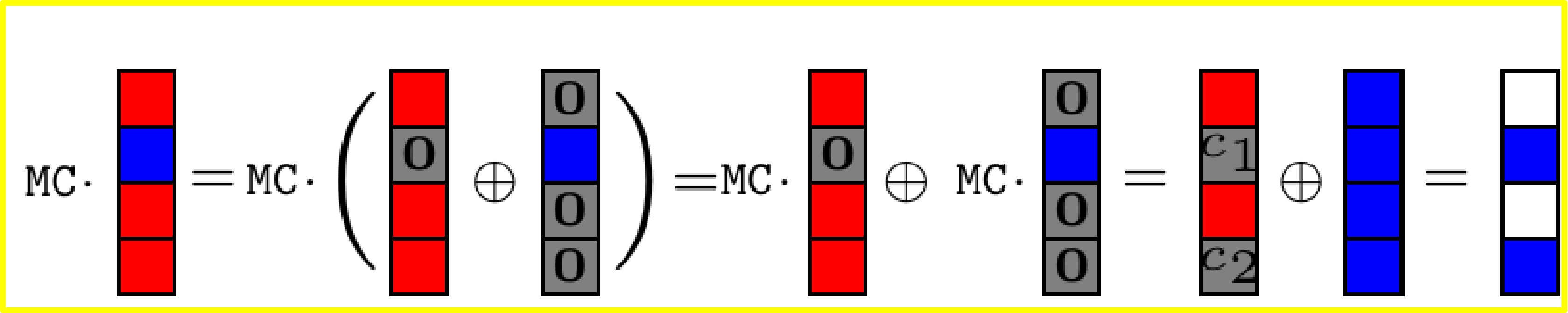
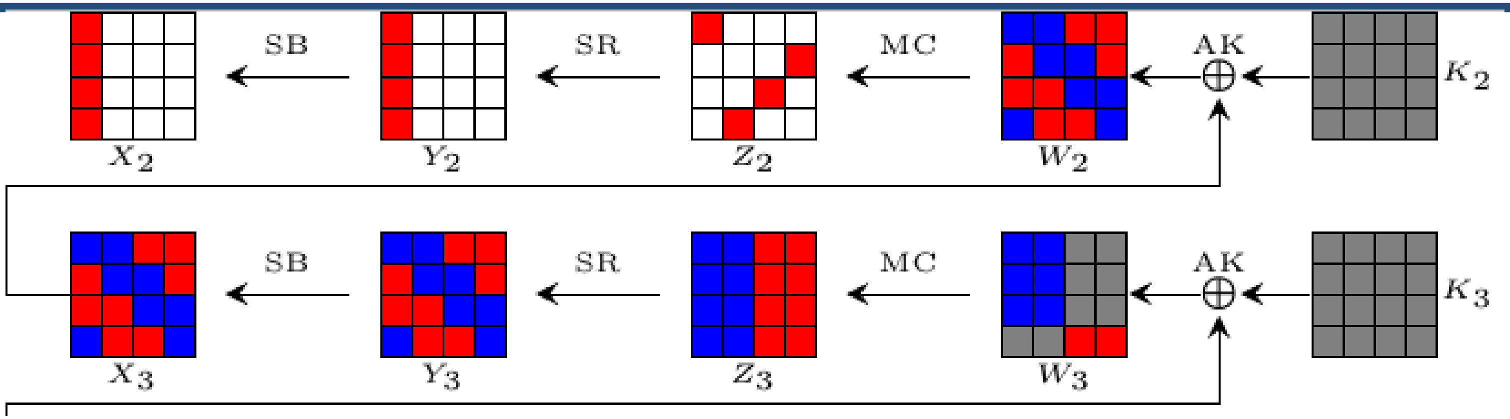


Fig: Initial Structure of a MITM preimage attack on AES-128 Hashing [FSE: Sasaki11]

- Compute the solution space of Blue and Red cells by solving the linear equations.



Nonlinear Constrained Neutral Words [C:DHALWH21]



1. From W_2 to Z_2: $MC^{-1} \cdot \begin{bmatrix} B_1 & B_2 & 0 & 0 \\ 0 & B_3 & B_4 & 0 \\ 0 & 0 & B_5 & B_6 \\ B_7 & 0 & 0 & B_8 \end{bmatrix} = \begin{bmatrix} c_1 & - & - & - \\ - & - & - & c_2 \\ - & - & c_3 & - \\ - & c_4 & - & - \end{bmatrix}$

2. From W_2 to W_3: $MC \cdot \begin{bmatrix} S(B_1) & S(B_2) \\ S(B_3) & S(B_4) \\ S(B_5) & S(B_6) \\ S(B_7) & S(B_8) \end{bmatrix} = \begin{bmatrix} - & - \\ - & - \\ - & - \\ c_5 & c_6 \end{bmatrix}$

◆ Neutral Words are nonlinearly constrained

- Compute the solution of Blue by solving the nonlinear equations.
- It is difficult to solve the nonlinear equations.

Nonlinear Constrained Neutral Words [C:DHALWH21]

Table based technique

Traverse 8 Blue cells :

- Compute to get the values of $(c_1, c_2, \cdots, c_6)$ and store in a list V.

$$MC^{-1} \cdot \begin{bmatrix} B_1 & B_2 & 0 & 0 \\ 0 & B_3 & B_4 & 0 \\ 0 & 0 & B_5 & B_6 \\ B_7 & 0 & 0 & B_8 \end{bmatrix} = \begin{bmatrix} c_1 & - & - & - \\ - & - & - & c_2 \\ - & - & c_3 & - \\ - & c_4 & - & - \end{bmatrix}$$

$$MC \cdot \begin{bmatrix} S(B_1) & S(B_2) \\ S(B_3) & S(B_4) \\ S(B_5) & S(B_6) \\ S(B_7) & S(B_8) \end{bmatrix} = \begin{bmatrix} - & - \\ - & - \\ - & - \\ c_5 & c_6 \end{bmatrix}$$

List V	
$(c_1, c_2, \cdots, c_6)$	Values of 8 Blue cell
$(0,0,\cdots,0)$	i_0, i_1 ,...
$(0,0,\cdots,1)$	j_0, j_1 ,...
...	...

MITM preimage attack with table based technique

For Gray cells:
    Build table V and U by table based technique
    For $c^+ = (c_1, c_2, \cdots, c_{l_1}) \in \mathbb{F}_2^{w \cdot l_1}$:
        For $c^- = (c'_1, c'_2, \cdots, c'_{l_2}) \in \mathbb{F}_2^{w \cdot l_2}$:
            For Blue cells in V$[c^+]$:
                Compute forward...
            For Red cells in U$[c^-]$:
                Compute backward find matching

[1] Dong, X., Hua, J., Sun, S., Li, Z., Wang, X., & Hu, L. (2021). Meet-in-the-middle attacks revisited: key-recovery, collision, and preimage attacks. In *Advances in Cryptology–CRYPTO 2021. Part III 41* (pp. 278-308).

3 Attack on Streebog

- **Streebog** is a Russian national standard hash function.

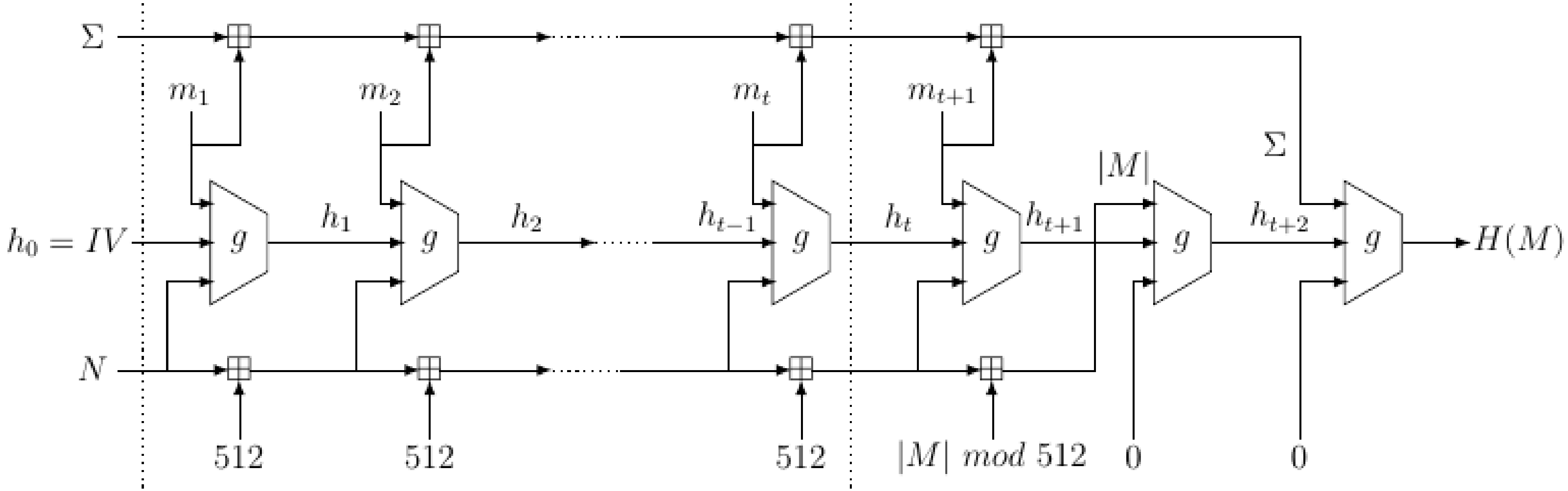


Figure 2: The Streebog hash function

Compression function of Streebog

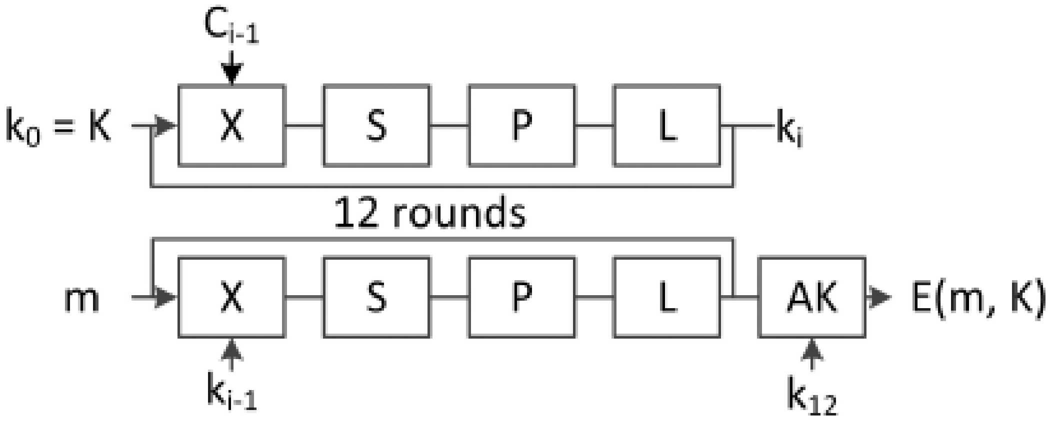


Fig. 2. The internal block cipher (E)

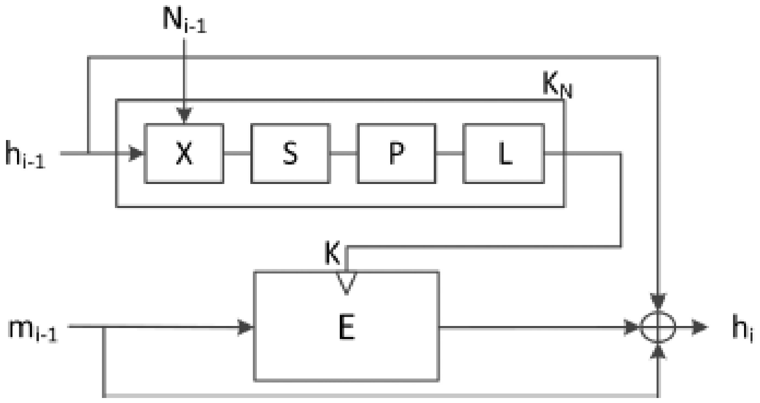
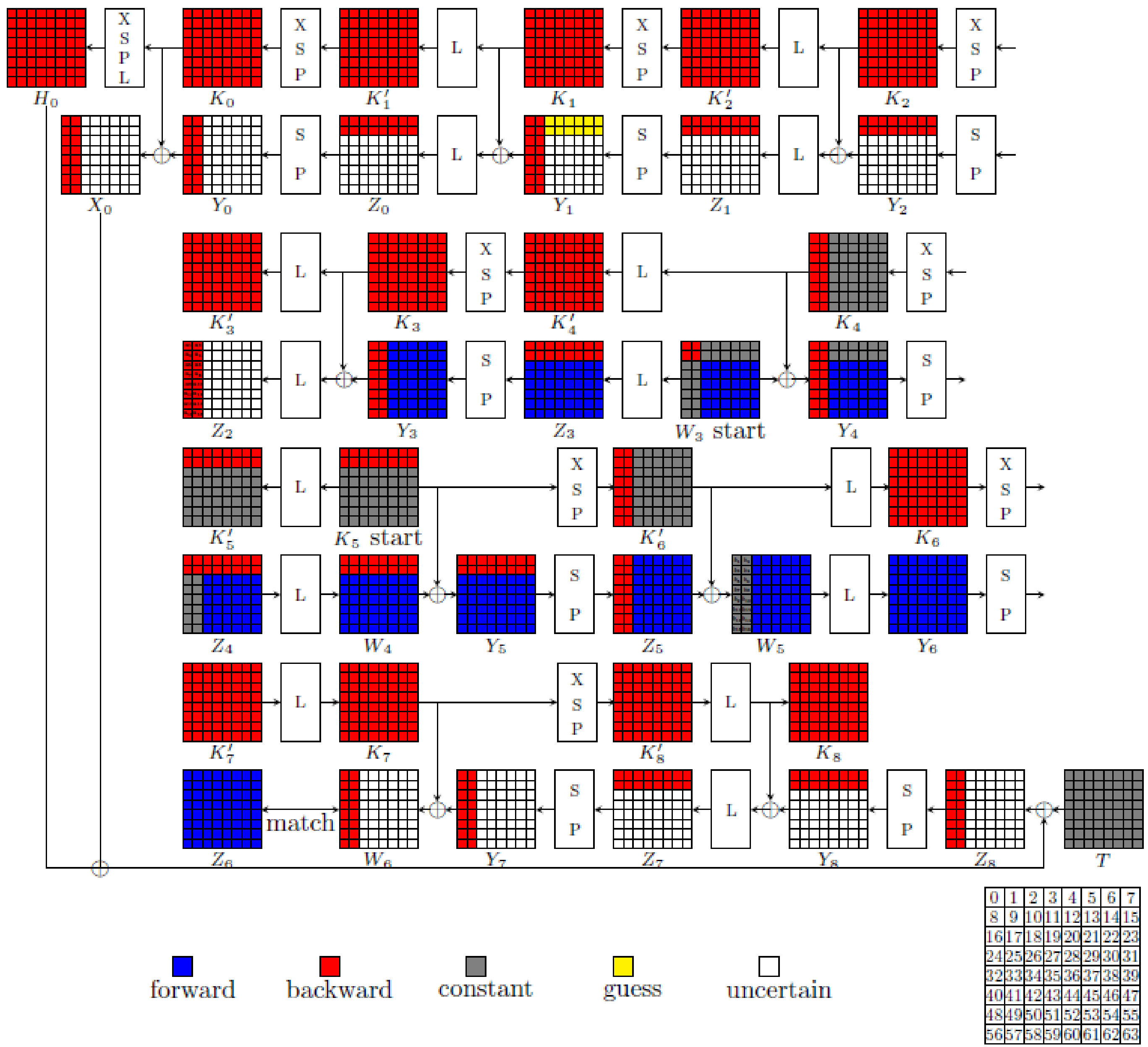


Fig. 1. Stribog's compression function g_N

MP mode: $CF(h_{i-1}, m_{i-1}) = E_{h_{i-1}}(m_{i-1}) \oplus m_{i-1} \oplus h_{i-1}$

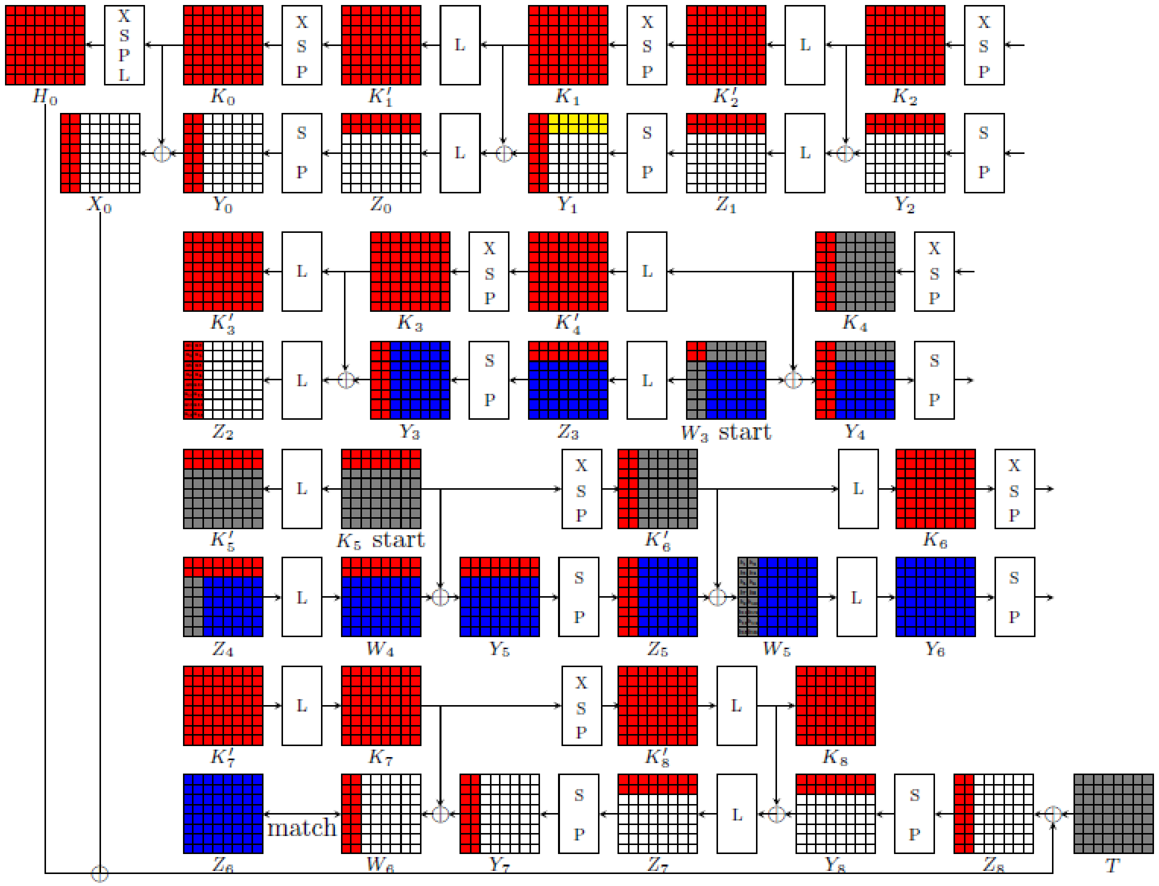
- AddKey(X): XOR with either a round key, a constant, or the counter of bits hashed so far (N).
- SubBytes (S): A nonlinear byte bijective mapping.
- Transposition (P): Byte permutation.
- Linear Transformation (L): Row multiplication by an MDS matrix in GF(2).

8.5-round preimage attack on Streebog-512 compression function



- Blue cell: 36-16=20
- Red cell: 20-16=4
- Matching: 16
- Guessed cell: 12

8.5-round preimage attack on Streebog-512 compression function



Algorithm 4: The MITM preimage attack on 8.5-round `Streebog-512` compression function

Fix all ■ cells of K_5 to 0 and arbitrary 16 ■ cells of W_3 to 0.
for *All 8 no fixed* ■ *cells in* W_3 **do**
  Call Algorithm 2 to build V and U.
  for $\mathfrak{c}^+ = (a_1, a_2, \cdots, a_{16}) \in \mathbb{F}_2^{8\times 16}$ **do**
    for $\mathfrak{c}^- = (b_1, b_2, \cdots, b_{16}) \in \mathbb{F}_2^{8\times 16}$ **do**
      for *all values in* $V[\mathfrak{c}^+]$ **do**
        Compute forward to get the full state of Z_6 and store it in a table L.
      for $\mathcal{Y}_{-}^{\text{ENC}} \in \mathbb{F}_2^{8\times 12}$ *(■ cells of* Y_1*)* **do**
        for *all values in* $U[\mathfrak{c}^-]$ **do**
          Compute backward to get the first two columns of W_6 and search L to find matching.
          Use the matching pairs to compute and check if the guessed values $\mathcal{Y}_{-}^{\text{ENC}}$ are correct.
          if *The guessed values* $\mathcal{Y}_{-}^{\text{ENC}}$ *are correct* **then**
            Test the full preimage.
            if *The full preimage is found* **then**
              Output and stop.

- Forward matching values : $2^{8\times 20}$
- 16-cell matching: $2^{8\times 16}$
- Correct partial matching: $2^{8\times(20+16-16-12)} = 2^{8\times 8}$

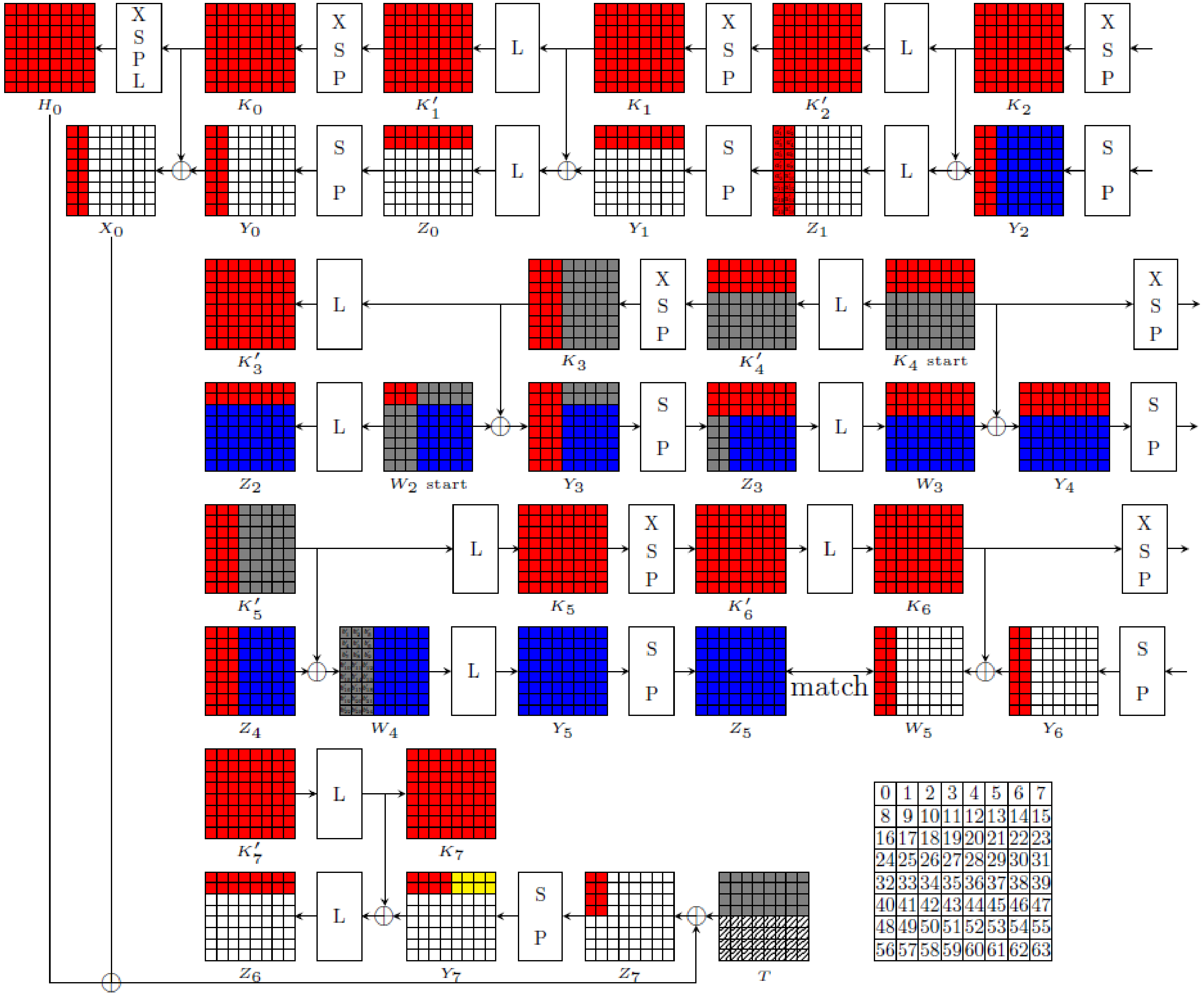
Backward matching values : $2^{8\times 16}$

Probability of a correct guess : $2^{-8\times 12}$

Find full match : Repeat $2^{8\times 40}$

Time complexity: $2^{8\times 40}(2^{8\times 20} + 2^{8\times 16} + 2^{8\times 8}) \approx 2^{480}$

7.5-round preimage attack on Streebog-256 compression function



- Blue cell: 30-16=14
- Red cell: 30-24=6
- Matching: 16
- Guessed cell: 8

Time complexity:

$$2^{8\times 12}(2^{8\times 14} + 2^{8\times 14} + 2^{8\times 4}) \approx 2^{208}$$

Our contribution

Table 1: Summary of preimage attack results on Streebog

Algorithm	Target	Rounds	Time	Memory	Ref.
Streebog-256 (12 rounds)	Compression Function	6.5	2^{232}	2^{120}	[MLHL15b]
		6.5	2^{209}	2^{160}	Sect. 7
		7.5	2^{209}	2^{192}	Sect. 5.3
	Hash Function	5	2^{192}	2^{64}	[MLHL15b]
		5	2^{208}	2^{12}	[MLHL15b]
		6.5	2^{232}	2^{120}	[MLHL15b]
		6.5	2^{209}	2^{160}	Sect. 7
Streebog-512 (12 rounds)	Compression Function	6	2^{496}	2^{64}	[ZWW13]
		6	2^{496}	2^{112}	[AY14]
		7.5	2^{496}	2^{64}	[MLHL15b]
		7.5	2^{441}	2^{192}	Sect. A
		8.5	2^{481}	2^{288}	Sect. 5.2
	Hash Function	6	2^{505}	2^{64}	[ZWW13]
		6	2^{505}	2^{256}	[AY14]
		6	2^{496}	2^{64}	[MLHL14]
		6	2^{504}	2^{11}	[MLHL14]
		7.5	2^{496}	2^{64}	[MLHL15b]
		7.5	2^{504}	2^{11}	[MLHL15b]
		7.5	$2^{478.25}$	2^{256}	Sect. 6
		8.5	$2^{498.25}$	2^{288}	Sect. 6

Thank you!

Preimage attack on Streebog-512 [AlTawy et al. AFRICACRYPT14]

1. From H(M), produce 2^{16} pseudo preimage for the last compression function. T: 2^{16} pairs of (h_{515}, Σ).

2. By using multicollisions, we construct 2^{512} messages which lead all to the same value of h_{512} . Specifically, $M_i = m_1^j || m_2^j \ldots || m_{512}^j$ $(j \in$

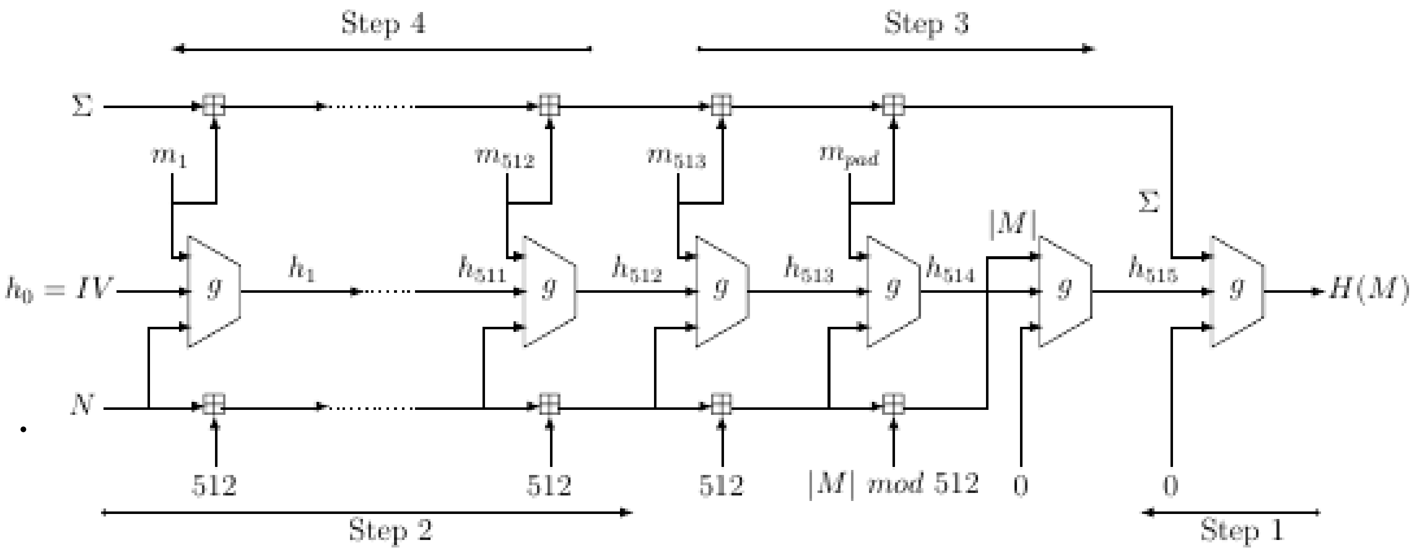


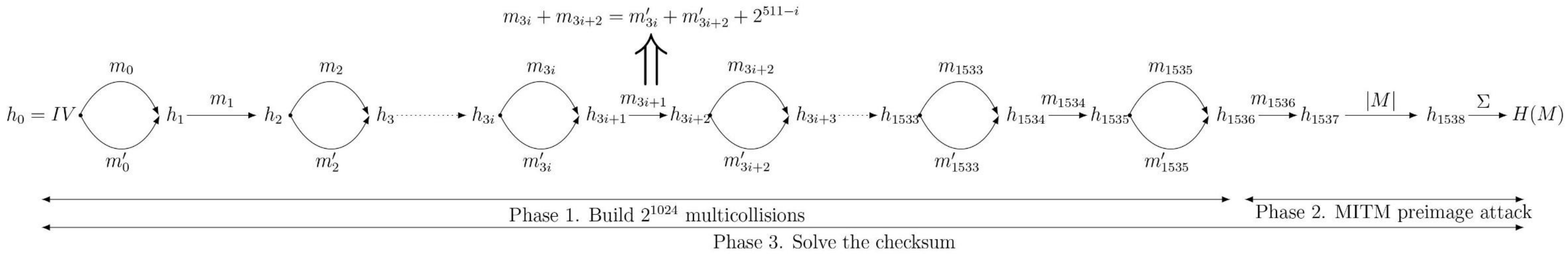
Figure 5: preimage attack on Streebog-512

3. Let the message is 513 complete blocks, $m_{pad}, |M|$ are known. Randomly choose m_{513} to compute h_{515} and check if it exists in T. We expect to find a match after 2^{496} guessing. Then Σ is known, $\Sigma_{M_i} = \Sigma - m_{pad} - m_{513}$.

4. Compute all the sums of all the $\Sigma_{M_1} = m_1^j + m_2^j + \cdots + m_{256}^j$, store them in T_1. Then, compute the sum of other messages $\Sigma_{M_2} = m_{257}^j + m_{258}^j + \cdots + m_{512}^j$ and check if $\Sigma - \Sigma_{M_2}$ is in T_1. Once we find a match, $M = m_1^j \Big|\Big| m_2^j \Big|\Big| \ldots || m_{513}^j$ is the preimage of the given \$H(M)\$.

Time complexity: $2^{16} * 2^{480} + 512 * 2^{256} + 3 * 2^{496} + 2^{256} \approx 2^{498}$

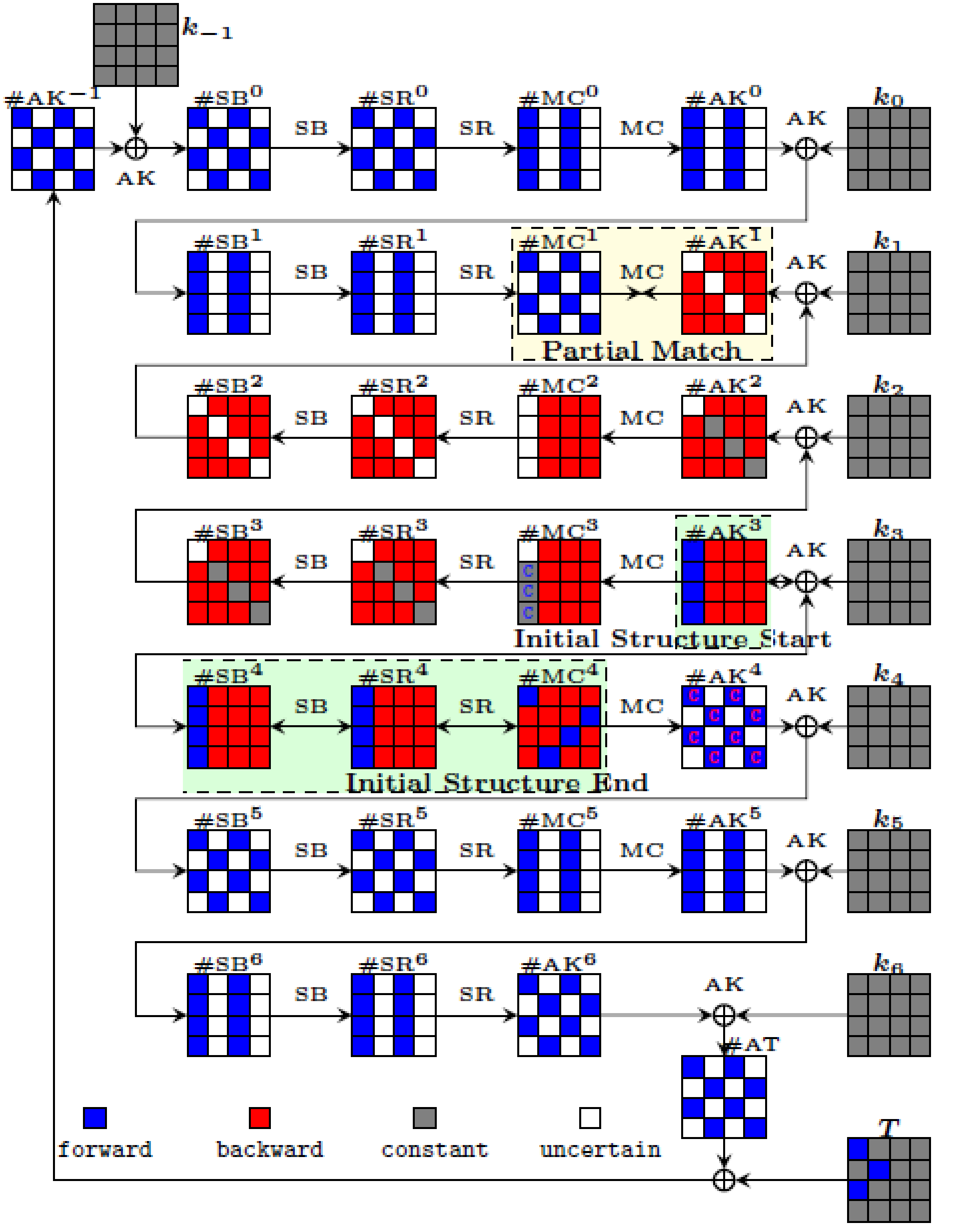
Preimage attack on 6.5-round Streebog-512 [Ma et al. IWSEC 2015]



- Phase 1: 2^{1024}-multicollisions are constructed with 512 cascacded 4-multicollisions pairs, $(M_{3i}, M'_{3i})||M_{3i+1}||(M_{3i+2}, M'_{3i+2})$, satisfy $M_{3i} + M_{3i+2} = M'_{3i} + M_{3i+2} + 2^{511-i}$
- Phase 2: With h_{1536}, randomly choose one more message block m_{1536} which satisfies padding, get $|M|$. Then can get h_{1538}, using preimage attack on compression function to generate Σ.
- Phase 3: Find desired checksum.

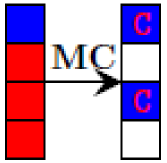
1. let $S = H(M) - m_{1536}$ denote the checksum which we are desired.
2. Compute $C = S - (\sum_{i=0}^{511}(m_{3i} + m_{3i+2})) = \sum_{i=0}^{511} k_i 2^i$, the k_i sequence is the binary representation of C.
3. Set M be an empty message.
4. For $i = 0$ to 511:
   (a) If $k_i = 0$, then $M = m||m_{3i}||m_{3i+1}||m_{3i+2}$.
   (b) Else $M = m||m'_{3i}||m_{3i+1}||m'_{3i+2}$
5. $M = M||m_{1536}$

MITM Preimage attack on AES-hash [FSE:Sasaki11]



- Initial structure: add constraints to cancel impact

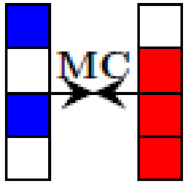
Constraints on $\#MC^4[1,2,3]$ to build the initial structure:



$$i.e., \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 0 \\ \#MC^4[1] \\ \#MC^4[2] \\ \#MC^4[3] \end{bmatrix} = \begin{bmatrix} c_0 \\ - \\ c_1 \\ - \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} 3\cdot\#MC^4[1] & \oplus\ 1\cdot\#MC^4[2] & \oplus\ 1\cdot\#MC^4[3] \\ 1\cdot\#MC^4[1] & \oplus\ 2\cdot\#MC^4[2] & \oplus\ 3\cdot\#MC^4[3] \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$$

- Indirect partial matching



Know any b bytes (b > 4) among the input and output of MixColumns on one column.

Get a filter of b − 4 bytes.

Our contribution

- Introduce some techniques on the MILP model.
- Build MILP model of Streebog and get some improved results.

Table 1: Summary of preimage attack results on Streebog

Algorithm	Target	Rounds	Time	Memory	Ref.
Streebog-256 (12 rounds)	Compression Function	6.5	2^{232}	2^{120}	[MLHL15b]
		6.5	2^{209}	2^{160}	Sect. 7
		7.5	2^{209}	2^{192}	Sect. 5.3
	Hash Function	5	2^{192}	2^{64}	[MLHL15b]
		5	2^{208}	2^{12}	[MLHL15b]
		6.5	2^{232}	2^{120}	[MLHL15b]
		6.5	2^{209}	2^{160}	Sect. 7
Streebog-512 (12 rounds)	Compression Function	6	2^{496}	2^{64}	[ZWW13]
		6	2^{496}	2^{112}	[AY14]
		7.5	2^{496}	2^{64}	[MLHL15b]
		7.5	2^{441}	2^{192}	Sect. A
		8.5	2^{481}	2^{288}	Sect. 5.2
	Hash Function	6	2^{505}	2^{64}	[ZWW13]
		6	2^{505}	2^{256}	[AY14]
		6	2^{496}	2^{64}	[MLHL14]
		6	2^{504}	2^{11}	[MLHL14]
		7.5	2^{496}	2^{64}	[MLHL15b]
		7.5	2^{504}	2^{11}	[MLHL15b]
		7.5	$2^{478.25}$	2^{256}	Sect. 6
		8.5	$2^{498.25}$	2^{288}	Sect. 6