The Rebound Attack

Halil İbrahim Kaplan

TDBY / Kripto Analiz Laboratuvarı

halil.kaplan@tubitak.gov.tr

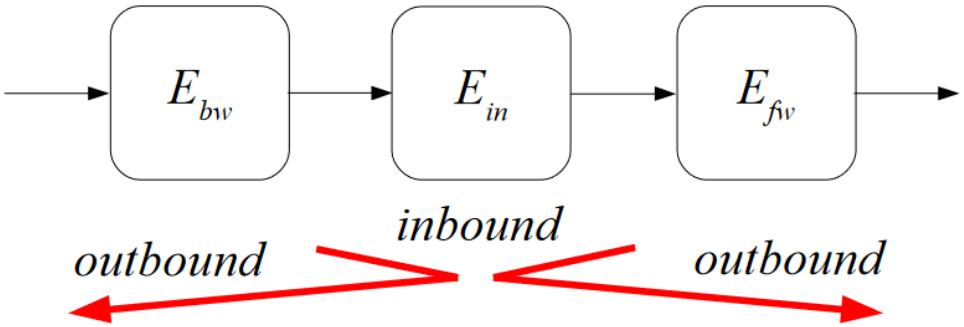
2021

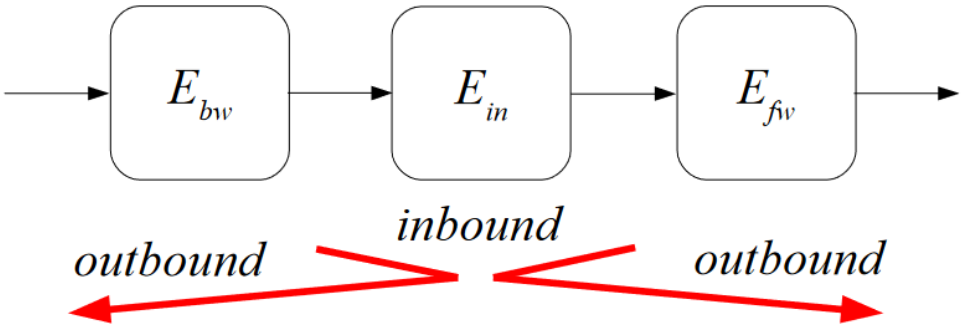
1. **Preview of Results**
2. **The Whirlypool Hash Function**
3. **Collision Attack on 4.5 Rounds Whirlpool**

IDEA : Use the available degrees of freedom in a collision attack to efficiently bypass the low probability parts of a differential trail.

Consider the internal cipher of a hash or compression function as a 3 sub-ciphers:

$$E = E_{fw} \, o \, E_{in} \, o \, E_{bw}$$



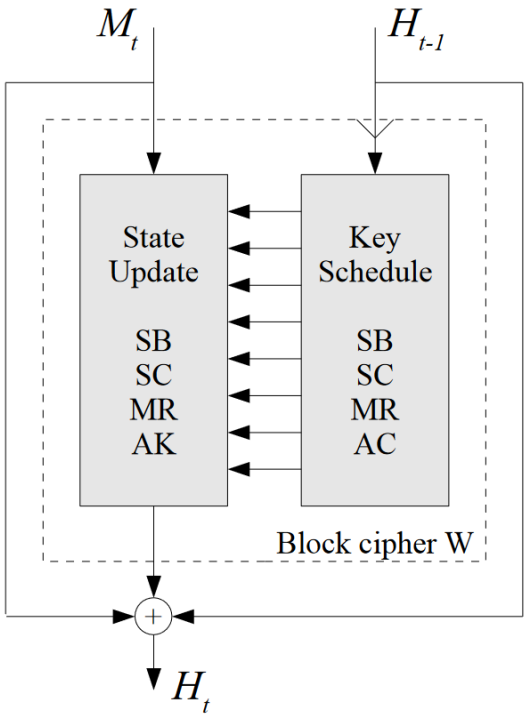


Inbound phase : Match in the middle approach in E_{in}

Outbound phase: We use truncated differentials in both forward- and backward direction through E_{fw} and E_{bw}

THE WHIRLPOOL HASH FUNCTION

- Iterated hash function.
- The underlying block cipher W operates in the Miyaguchi-Preneel mode



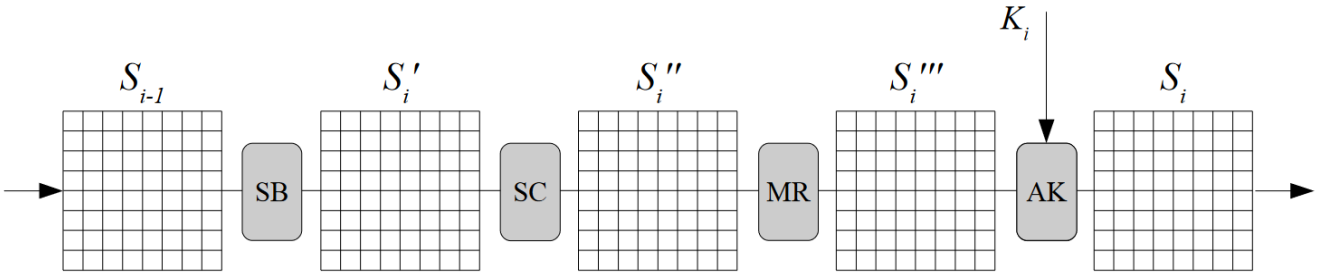
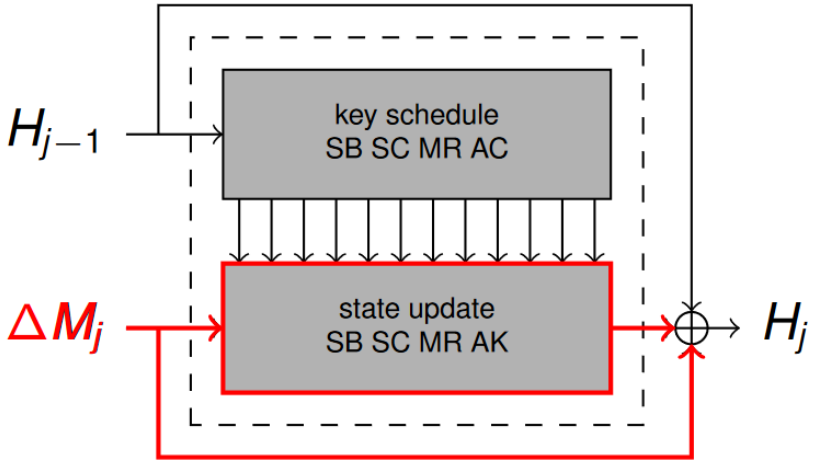


Fig. 3. One round r_i of the Whirlpool compression function with 8×8 states S_{i-1}, S'_i, S''_i, S'''_i, S_i and round key input K_i.

REBOUND ATTACK ON WHIRLPOOL

Overview:

If the differences in the message words are the same as in the output of the state update transformation, the differences cancel each other through the feed-forward



Overview:

The core of the attack is 4 round trail of the form

$$1 \rightarrow 8 \rightarrow 64 \rightarrow 8 \rightarrow 1$$

This trail has the minimum number of active S-boxes and has the best differential probability according to the wide trail design strategy.

Overview:

Split the block cipher W into three sub-ciphers

$$W = E_{fw} \circ E_{in} \circ E_{bw}$$

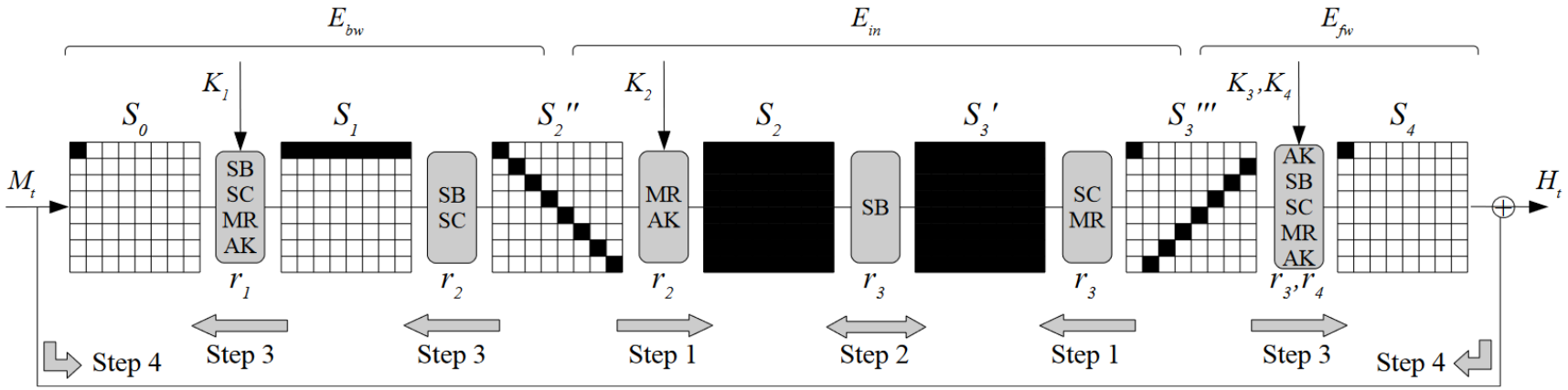
Most expensive part of the differential trail is covered

relatively low probability and can be fulfilled in a probabilistic way

$$E_{bw} = SC \circ SB \circ AK \circ MR \circ SC \circ SB$$
$$E_{in} = MR \circ SC \circ SB \circ AK \circ MR$$
$$E_{fw} = AK \circ MR \circ SC \circ SB \circ AK$$

Overview:

- **Inbound phase**
  Step 1: start with 8-byte truncated differences at the MixRows layer of round r_2 and r_3, and propagate forward and backward to the S-box layer of round r_3.
  Step 2: connect the input and output of the S-boxes of round r_3 to form the three middle states $8 \rightarrow 64 \rightarrow 8$ of the trail.
- **Outbound phase**
  Step 3: extend the trail both forward and backward to give the trail $1 \rightarrow 8 \rightarrow 64 \rightarrow 8 \rightarrow 1$ through MixRows in a probabilistic way.
  Step 4: link the beginning and the end of the trail using the feed-forward of the hash function.



Collision Attack for 4.5 Rounds:

The sequence of truncated differentials has the form:

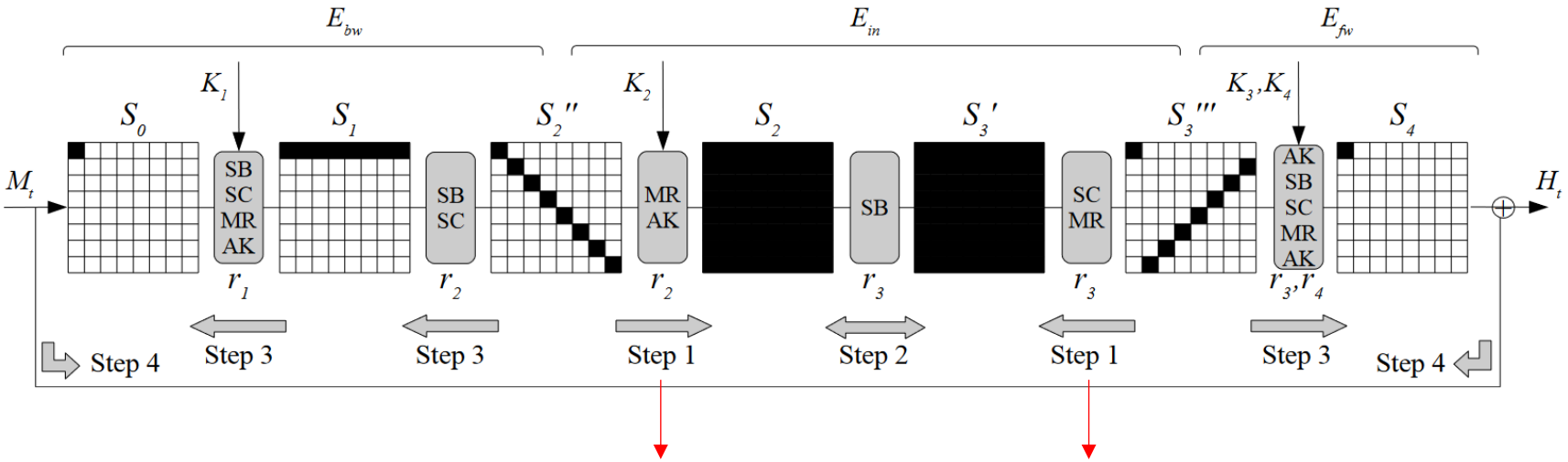
$$1 \xrightarrow{r_1} 8 \xrightarrow{r_2} 64 \xrightarrow{r_3} 8 \xrightarrow{r_4} 1 \xrightarrow{r_{4.5}} 1$$

we will analyze the 4 step of the attack in detail.

Precomputation:

Compute 256 x 256 lookup table for each S-box differential

Step 1 :

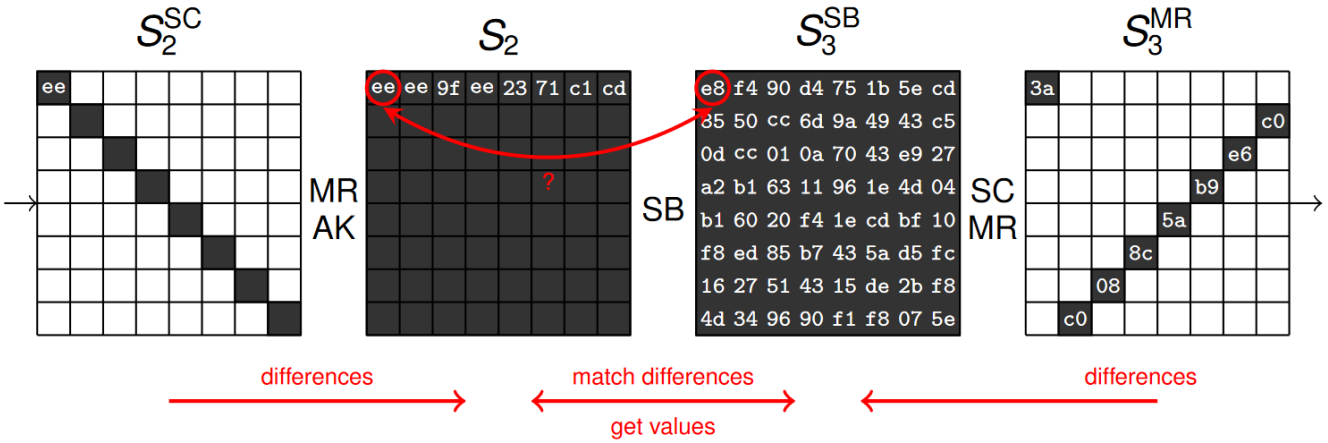


Choose random difference with 8 active bytes of state S_2''

Choose another difference and 8 active bytes in state S_3'' and propagate backwards. Again, the diagonal shape ensures that we get a full active state at the output of SubBytes of round r_3

Note that all active bytes have to be in the diagonal of state S_2'' Then, the differences propagate forward to a full active state at the input of the next SubBytes layer (state S_2) with a probability of 1.

Step 2 :



For each byte i of S_2 and j of S_3^{SB},

Check whether S-box(i) = j

Step 2 :

We can find a match with probability $\frac{1}{2}$ for each byte

Input Difference	Output Difference 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Step 2 :

We can find a match with probability $\frac{1}{2}$ for each byte

$\Downarrow$

We can find math for whole state with probabilty about 2^{-64}

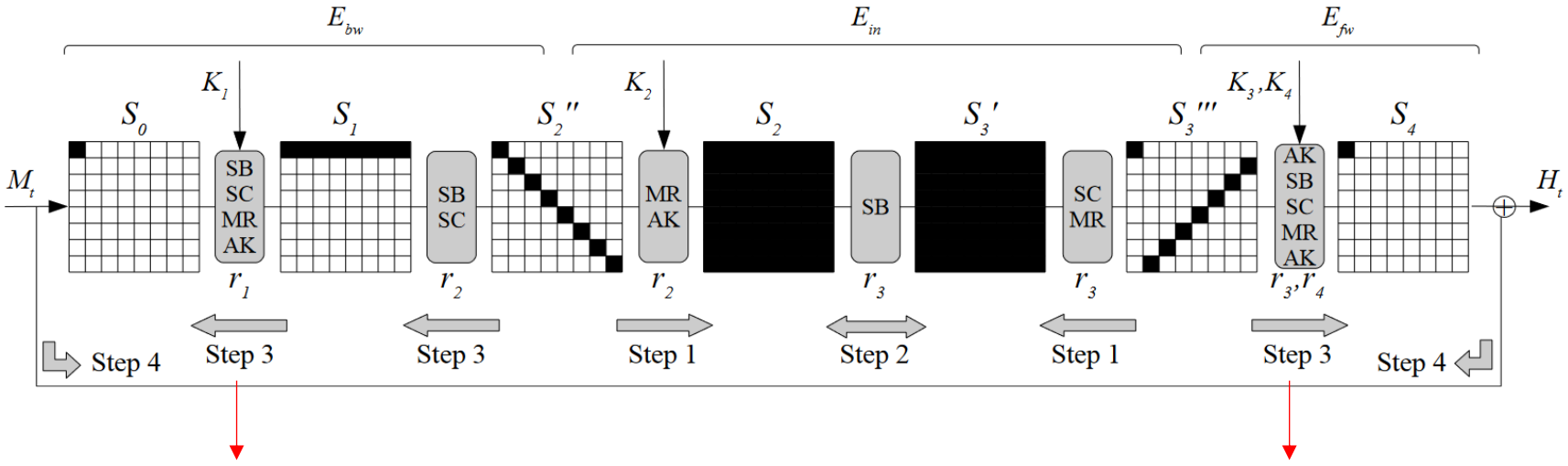
So,

After repeating Step 1 2^{64} times, we expect to have matching in the subbyte layer.

Since we get at least two state values for each S-box match, we get about 2^{64} starting points for the outbound phase

Step 3 :

Extend the differential path backward and forward.



Probability = 2^{-56}

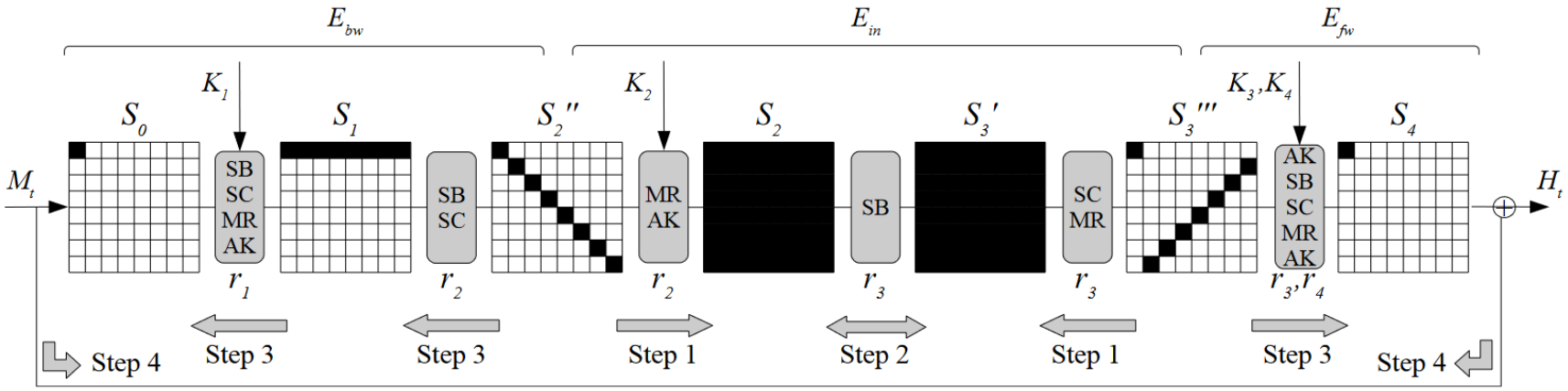
Probability = 2^{-56}

Probability of the outbound phase $= 2^{-2*56} = 2^{-112}$

Step 4 :

To construct a collision at the output of this 4 round compression function, the exact value of the input and output difference has to match.

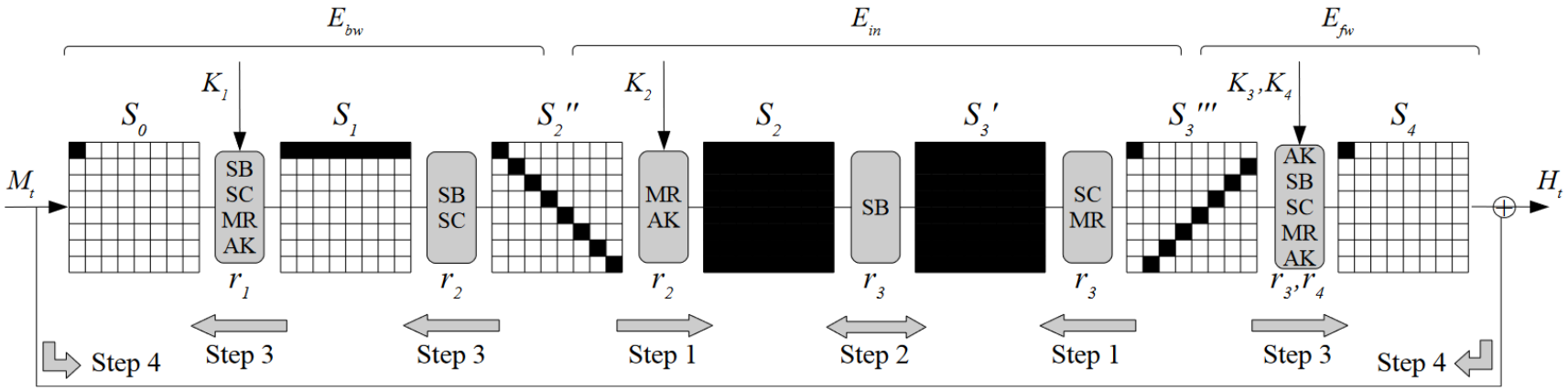
Since only one byte is active, this can be fulfilled with a probability of 2^{-8}.



Step 4 :

Hence, the complexity to find a collision for 4 rounds of Whirlpool is $2^{112+8} = 2^{120}$.

Note that we can add half of a round (SB,SC) at the end for free, since we are only interested in the number of active bytes



References

1. Courtois, N., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Cryptology ePrint Archive, Report 2002/044 (2002)

2. Cid C., Leurent G. (2005) An Analysis of the XSL Algorithm. In: Roy B. (eds) Advances in Cryptology - ASIACRYPT 2005. ASIACRYPT 2005. Lecture Notes in Computer Science, vol 3788. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11593447_18

3. Cui, Jie & Zhong, Hong & Wang, Jiankai & Shi, Run-hua. (2014). Generation and Optimization of Rijndael S-box Equation System. Information Technology Journal. 13. 2482-2488. 10.3923/itj.2014.2482.2488.

4. Lim, C., & Khoo, K. (2007). An Analysis of XSL Applied to BES. FSE.