Midori

Tri-hards

Department of Electrical Engineering and Computer Science
Indian Institute of Technology Bhilai

November 28, 2020

Outline

Lightweight Cryptography

- Lightweight Cryptography is essential for providing security for systems having a tight energy constraint such as, **RFID** tags, sensor nodes etc.
- To meet the rise in the need of Lightweight Cryptography, many cipher proposals have been made recently such as **KATAN**, **CLEFIA**, **PRESENT** etc, to name a few.
- The efficiency of Lightweight cryptography depends on the size of the circuit, throughput and most importantly energy constraints.

Efficiency of Lightweight Cryptography

- To meet the energy constraints of the system, the design choices of the system should be made so as to have good energy efficiency.
- The factors that affect the energy efficiency of the system are clock frequency, architecture of the system, loop rolling and choosing proper components for the construction of the system.
- Usage of 4×4 almost MDS binary matrices are efficient in terms of area and signal delay. Due to smaller branch number of almost MDS matrices, they require more number of rounds to ensure the security of the cipher.

Midori

- To balance this, optimal cell-permutation layers are implemented which improve diffusion speeds and increases the number of active S-Boxes in each round with lower implementation overheads than the ShiftRow permutation.
- **Midori** is an AES block cipher with 128 bit key and two variants having 64 and 128 bit block size respectively.
- This cipher outperforms ciphers such as **PRINCE** and **NOEKEON** in the aforementioned aspects and is well suited for systems with tight energy constraints.

Outline

Midori

Midori is a family of variant of SPN(Substitution Permutation Network) which as the name suggests has S-layer and P-layer and uses a 4 x 4 matrix as state matrix. It has two variants:

- **Midori64**: It has a block size of $n = 64$.
- **Midori128**: It is the other variant with block size of $n = 128$.

Midori variants

Comparing the two variants

Midori64 has cell size of $m = 4$ bits, while **Midori128** has cell size of $m = 8$ bits.

Suppose we define the cell elements of the state matrix as s_i, then we can say that $s_i \in \{0, 1\}^m$. Now we just load a 64-bit or a 128-bit plaintext P accordingly with respect to the variant of **Midori** being used.

	block size(n)	key size	cell size(m)	number of rounds
Midori64	64	128	4	16
Midori128	128	128	8	20

S-box

S-box

Here we use two types of bijective S-boxes namely Sb_0 and Sb_1, where $Sb_o, Sb_1 : \{0,1\}^4 \rightarrow \{0,1\}^4$.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
Sb_0	3	5	9	a	8	e	2	f	1	b	d	7	0	6	c	4
Sb_1	3	5	9	a	8	e	2	f	1	b	d	7	0	6	c	4

S-box

Special case for Midori128

Midori128 utilizes four other $8-bit$ s-boxes namely $SSb_0, SSb_1, SSb_2, SSb_3$, where we have that $SSb_0, SSb_1, SSb_2, SSb_3 : \{0,1\}^8 \rightarrow \{0,1\}^8$.

Midori128 S-box

What does new S-boxes contain??

SSb_i mainly consists of input and output bit permutations, in order to keep the involuntary property.

If we consider p_i as the input bit permutation of each of the SSb_i.

And now let us denote $x_{[i]}$ as the *ith* bit of x with $x_{[0]}$ being the most-significant bit, then we can see that we can easily denote,

$$p_i(x) = y^{(i)}$$

$$y^{(0)}_{[0,1,2,3,4,5,6,7]} = x_{[4,1,6,3,0,5,2,7]},\ y^{(1)}_{[0,1,2,3,4,5,6,7]} = x_{[1,6,7,0,5,2,3,4]}$$

$$y^{(2)}_{[0,1,2,3,4,5,6,7]} = x_{[2,3,4,1,6,7,0,5]},\ y^{(3)}_{[0,1,2,3,4,5,6,7]} = x_{[7,4,1,2,3,0,5,6]}$$

Round functions

Round functions

The round function mainly consists of: S-layer carried out by **SubCell**, P-layer carried out through **ShuffelCell** and **MixColumn** and finally a key-addition layer which **KeyAdd** handles.

Round functions

Round functions

1.**SubCell(S)**:We simply $s_i \leftarrow Sb_o[s_i]$ in **Midori64** and $s_i \leftarrow SSb_{imod(4)}[s_i]$ in **Midori64** for i in range $[0, 15]$.

2.**ShuffelCell(S)**:It applies the permutation as:

$(s0, s1, ..., s15)$, $(s0, s10, s5, s15, s14, s4, s11, s1, s9, s3, s12, s6, s7, s13, s2$

3.**MixColumn(S)**: It updates the values as

$(s_i, s_{i+1}, s_{i+2}, s_{i+3})^t \leftarrow M(s_i, s_{i+1}, s_{i+2}, s_{i+3})^t$, for $i = 0, 4, 8, 12$.

4.**KeyAdd(S,RK_i)**:The ith n-bit round key RK_i is XORed to state S.

Key Generation

Round Key Generation

For **Midori64** we start up with the 128-bit secret key K, which is then denoted as two 64-bit keys as K_0 and K_1.

Then, we define $WK = K_0 \oplus K_1$. But for **Midori128**, we define $WK = K$ and $RK_i = K \oplus \beta_i$.Where β_i is round constants which we will define later.Round keys generations are same for both variants.

Encryption

Summary for encryption

Algorithm $\mathsf{MidoriCore}_{(R)}(X, WK, RK_0, ..., RK_{R-2})$ **:**

$S \leftarrow \mathsf{KeyAdd}(X, WK)$

for $i = 0$ to $R - 2$ do

  $S \leftarrow \mathsf{SubCell}(S)$

  $S \leftarrow \mathsf{ShuffleCell}(S)$

  $S \leftarrow \mathsf{MixColumn}(S)$

  $S \leftarrow \mathsf{KeyAdd}(S, RK_i)$

$S \leftarrow \mathsf{SubCell}(S)$

$Y \leftarrow \mathsf{KeyAdd}(S, WK)$

optimal cell-permutation

Introduction to optimal cell-permutation layer

In order to achieve full diffusion in 3-rounds we have to take care of following conditions:

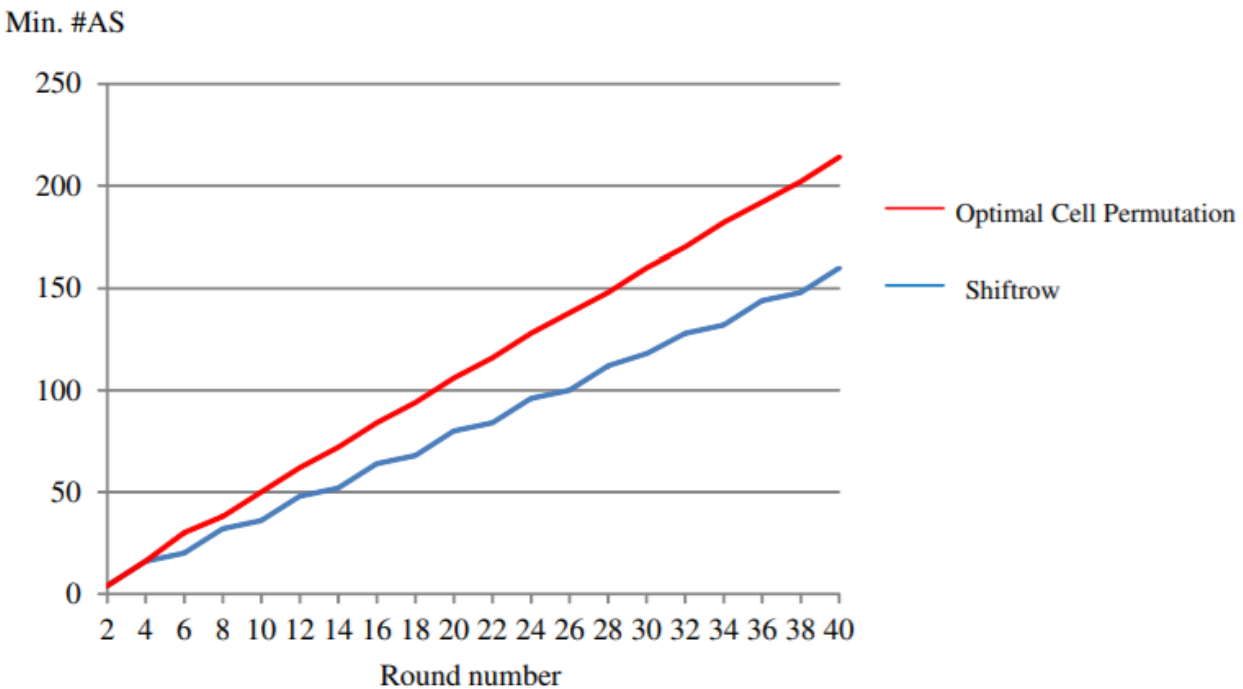
Condition-1: After cell-permutation application we have to make sure that the columns in input cell is mapped to some cell in different column all together.

Condition-2:After cell-permutation application we have to make sure that the columns in input cell is mapped to a cell in same row.

Condition-3:After applying cell-permutation once and also applying it three times inversely each cell in input must be mapped to a cell in the same row.

Comparision

optimal cell-permutation Vs Shift-Row



Key Scheduling

Key scheduling function

In order to save energy **Midori** doesn't employ any key scheduling function, the i-th round key is defined by $L{-}1(K_1) \oplus L{-}1(\beta_{18-i})$, where L^{-1} is inverse of linear layer.

L-inverse

L^{-1}

The L^{-1} has the operation defined as *InvShuffleCell*.*MixColumn*, and *InvShuffleCell* permutes the state as:

$(s0, s1, ..., s15) \leftarrow$

$(s0, s7, s14, s9, s5, s2, s11, s12, s15, s8, s1, s6, s10, s13, s4, s3)$.

Round Constants

Round Constants

We use a 4 x 4 binary matrix as round constants, which was derived from the hexadecimal encoding of fractional part of π.In order to get further round keys we just bitwise add the least significant bit of round key to previous round constants.

Outline

Differential and Linear Analysis

- All the S-Boxed used in the family of **Midori** block cipher, has a maximum differential and linear probability of 2^{-2}.
- DDT and LAT for Sb_0 is shown in the next two slides.

DDT

- DDT for Sb_0. Max differential probability is $\frac{4}{16}$.

in/out	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	-	-	-	-	-	-	-	-	-	-	-		-	-	-
1	-	2	4	-	2	2	2	-	2	-	-	-	-	-	2	-
2	-	4	-	-	4	-	-	-	-	4	-	-	4	-	-	-
3	-	-	-	-	2	-	4	2	2	2	-	-	-	2	-	2
4	-	2	4	2	2	2	-	-	2	-	-	2	-	-	-	-
5	-	2	-	-	2	-	-	4	-	2	4	-	2	-	-	-
6	-	2	-	4	-	-	-	2	2	-	-	-	2	2	-	2
7	-	-	-	2	-	4	2	-	-	-	-	2	-	4	2	-
8	-	2	-	2	2	-	2	-	-	2	-	2	2	-	2	-
9	-	-	4	2	-	2	-	-	2	2	-	2	2	-	-	-
A	-	-	-	-	-	4	-	-	-	-	4	-	-	4	-	4
B	-	-	-	-	2	-	-	2	2	2	-	4	-	2	-	2
C	-	-	4	-	-	2	2	-	2	2	-	-	2	-	2	-
D	-	-	-	2	-	-	2	4	-	-	4	2	-	-	2	-
E	-	2	-	-	-	-	-	2	2	-	-	-	2	2	4	2
F	-	-	-	2	-	-	2	-	-	-	4	2	-	-	2	4

LAT

- LAT for Sb_0. Max Linear probability is again $\frac{4}{16}$.

in/out	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	+8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	.	+2	+4	+2	-2	.	+2	.	-2	.	+2	.	+4	-2	.	-2
2	.	+4	.	.	+4	.	.	.	-4	.	.	.	.	+4	.	.
3	.	+2	.	+2	-2	.	+2	+4	+2	-4	-2	.	.	+2	.	+2
4	.	-2	+4	-2	+2	.	-2	.	-2	-4	-2	.	.	-2	.	+2
5	.	.	.	.	.	.	.	.	.	.	-4	-4	.	.	+4	-4
6	.	+2	.	+2	-2	.	+2	-4	-2	.	-2	.	-4	-2	.	+2
7	.	.	.	+4	.	.	-4	.	.	.	.	-4	.	.	-4	.
8	.	-2	-4	+2	-2	.	-2	.	-4	-2	.	+2	+2	.	+2	.
9	.	.	.	-4	-4	.	.	.	-2	+2	-2	-2	+2	+2	-2	+2
A	.	+2	.	-2	-2	-4	-2	.	.	-2	+4	-2	-2	.	+2	.
B	.	.	.	.	.	-4	.	-4	+2	-2	-2	+2	+2	+2	-2	-2
C	.	+4	.	.	.	.	-4	.	+2	+2	-2	+2	+2	-2	+2	+2
D	.	-2	+4	+2	-2	.	-2	.	.	+2	.	+2	-2	+4	+2	.
E	.	.	.	.	.	+4	.	-4	+2	-2	+2	-2	+2	+2	+2	+2
F	.	-2	.	+2	+2	-4	+2	.	.	+2	.	-2	+2	.	+2	+4

Minimum Active S-Boxes

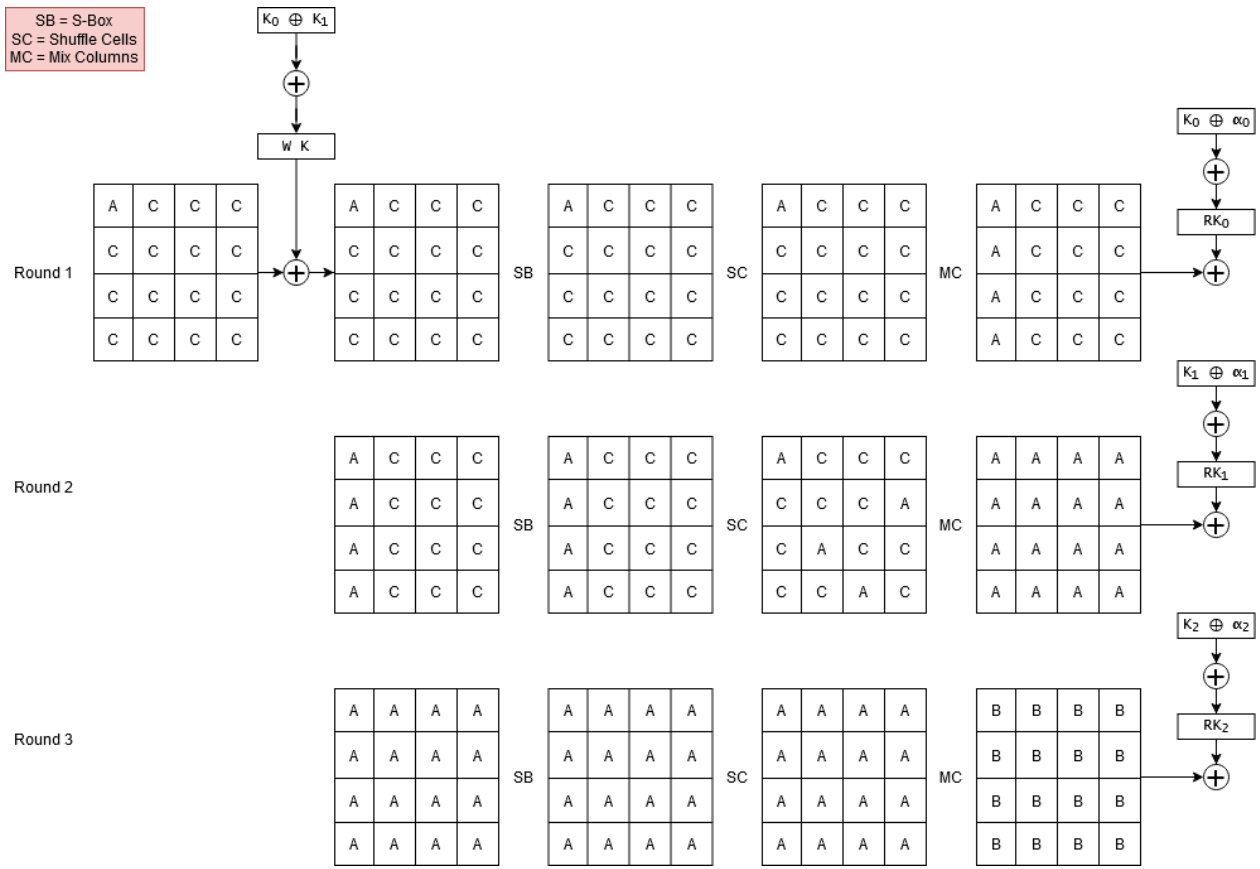
- Minimum number of deferentially/linearly active S-Box (AS) for **Midori64** and **Midori128**:

Round Number	4	5	6	7	8	9	10	11	12	13	14	15	16
Min. # of AS (Optimal Cell-Permutation)	16	23	30	**35**	38	41	50	57	62	**67**	72	72	84
Min # of AS (ShiftRow-type Permutation)	16	18	20	26	32	34	36	42	48	50	52	58	64

- For **Midori64**, the number of AS after round 7 is 32. Given the maximum differential and linear probability of 2^{-2}, the probability of differential and linear traits after round 7 is 2^{-64}.
- Similarly, after round 13 of **Midori128**, the maximum probability is 2^{-128}.

Integral Analysis

- Integral property for three round of Midori.



Integral Analysis

- After Mix-Column of Round 3 we observe that all cells has the **Balanced** property.
- After Mix-Column of Round 4 the cells will not have any of the property.
- Therefore, the Key Recovery Attack can be done up to a maximum of 5 Rounds.

Outline

Fault Analysis

- Fault analysis mixes an operational error by using laser irradiation, an abnormal voltage or an illegal clock.
- By doing so, we can compare the normal ciphertext and the faulty ciphertext and by doing so we can analyse and get some information regarding the secret key.
- The study of fault analysis is important from the viewpoint of IoT security.

Method of Fault Analysis

- Faults generated by irradiating laser are precise but they require a lot of additional equipment and is expensive, on the other hand a clock glitch can be realized with very less effort but we cannot control the location and the number of faults induced.
- To get the estimates of secret key K, we introduce a fault in the 16^{th} round and get the estimates of WK, then we do the same at 15^{th} round and get estimates of $K_\circ$ by using the estimated key WK from the first step and then finally K_1 can be calculated from the following formula:

$$K_1 = WK \bigoplus K_\circ$$

Fault Models

- The aforementioned Fault analysis included two fault models, initially we implement the first fault model and then we implement the second fault model, we utilized the clock glitch as the fault injection method in the fault analysis.
- In this glitch, we shorten the clock pulse and thus causing bit-flip with a probability of 0.5, but as it is not generated across all registers, therefore itś probability is less than 0.5 overall.

Fault Model - I

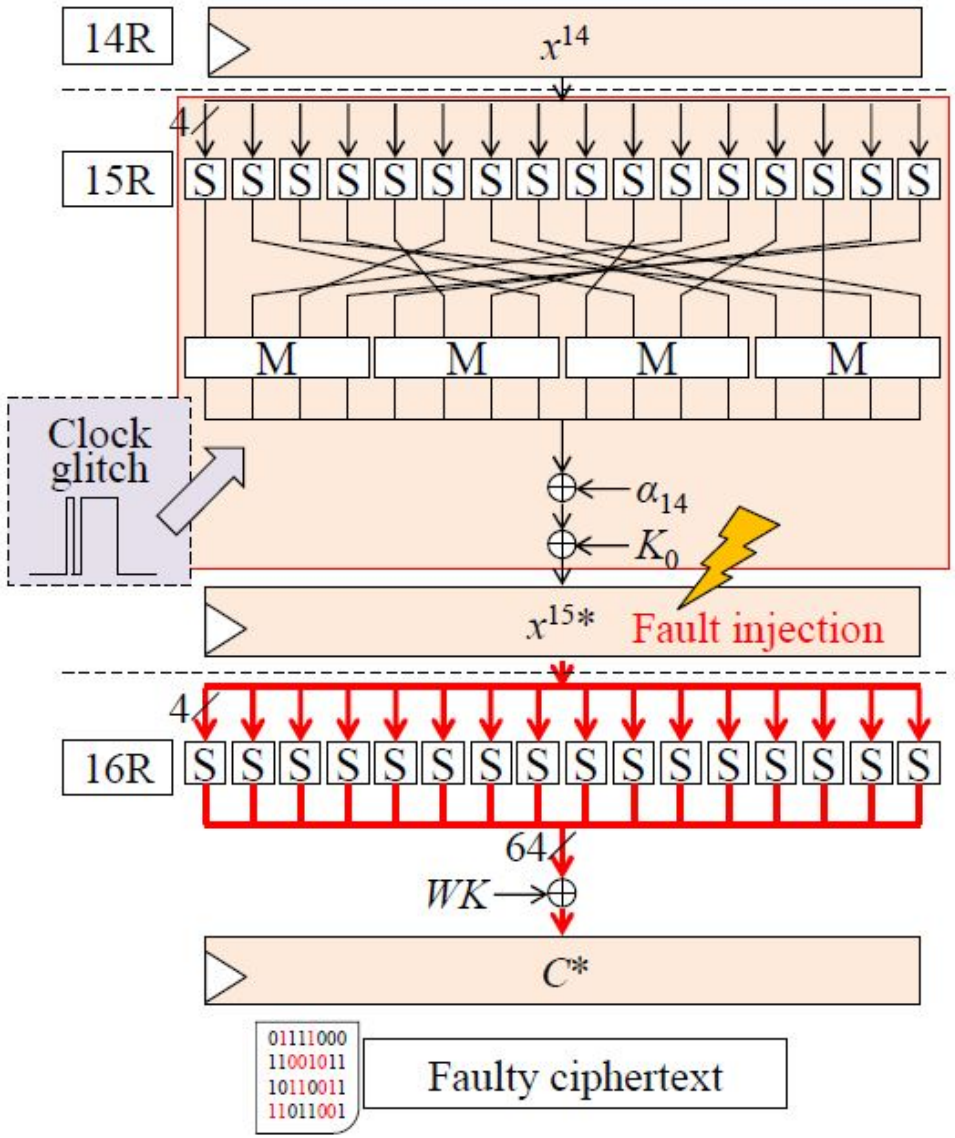
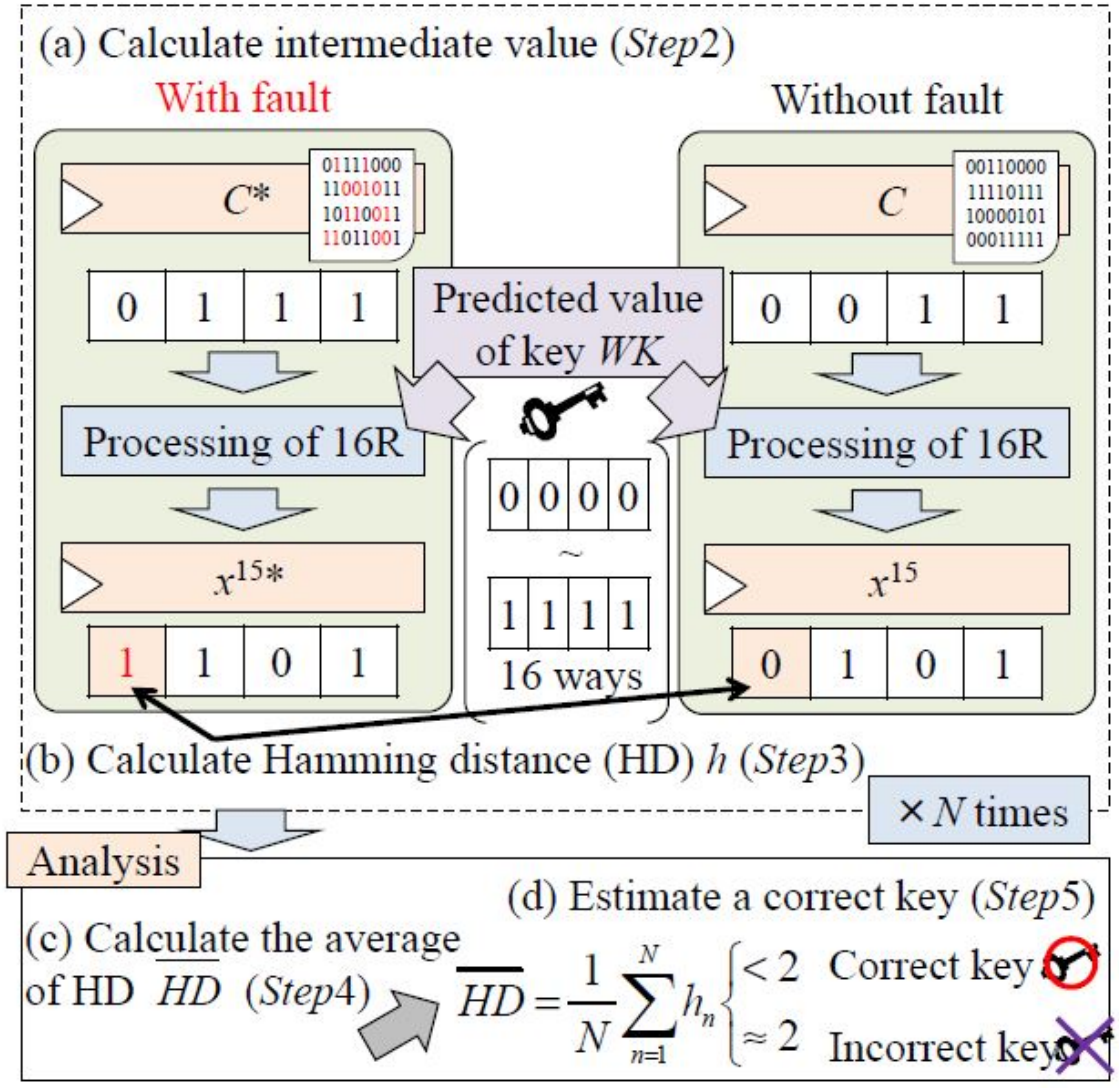


Fig. 2 Fault model I.

Fault Model - I(In Detail)



Fault Model - II

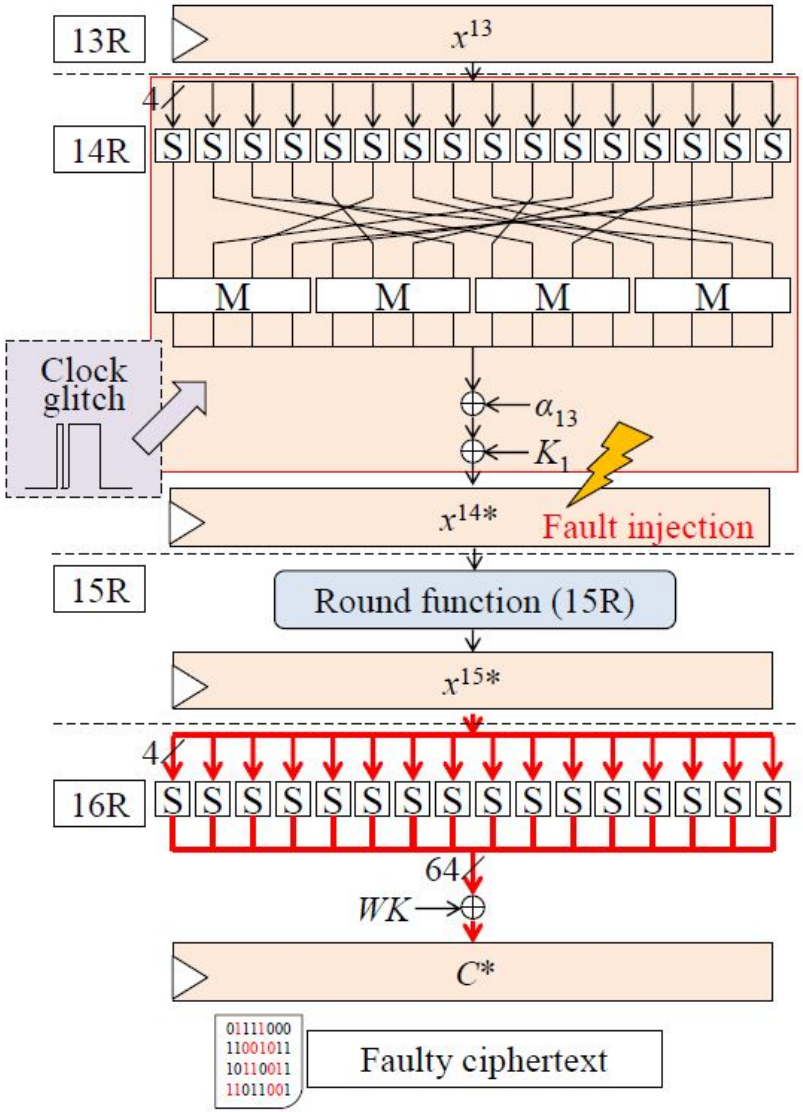
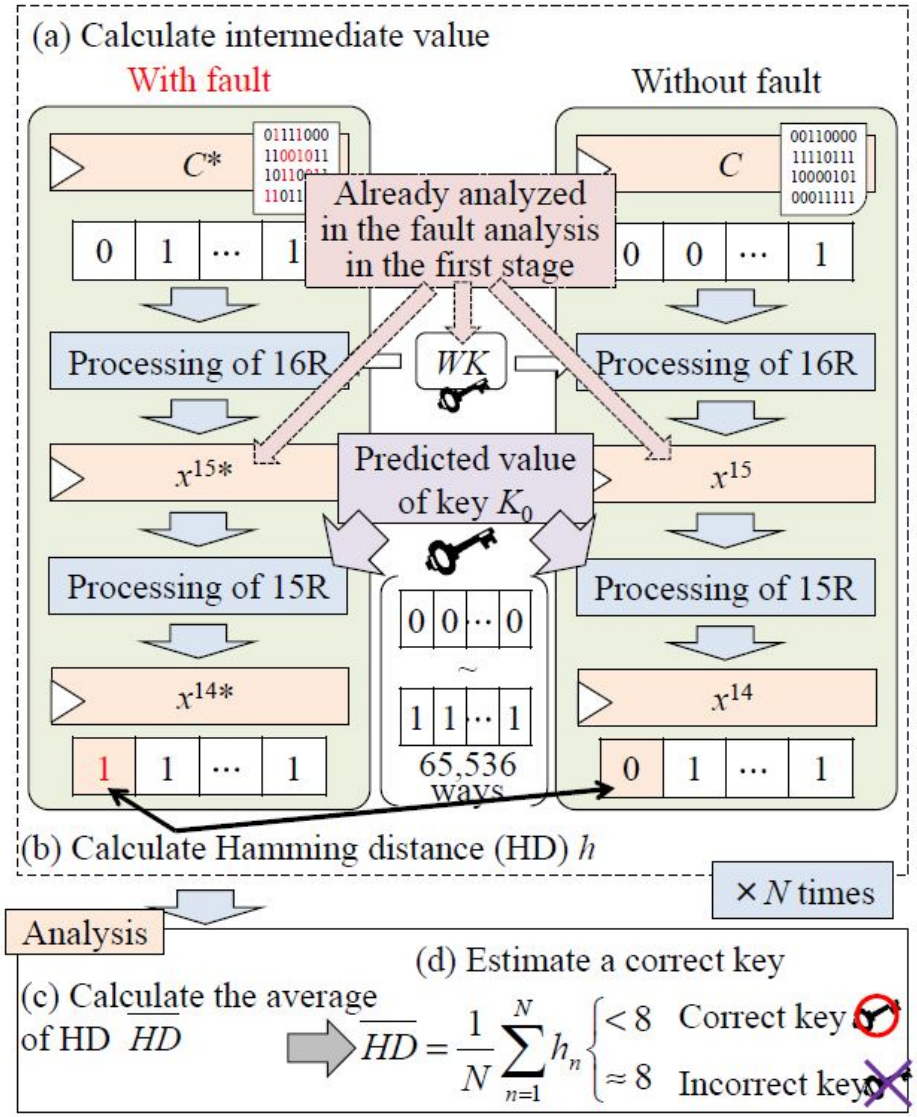


Fig. 3 Fault model II.

Fault Model - II(In Detail)



Outline

Conclusion

- We discussed the family of Midori block cipher, i.e. **Midori64** and **Midori128**.
- Midori is designed to be energy efficient and as lightweight as possible.
- We also looked at the benefits and usage of lightweight cryptography.

Conclusion

- Next, we looked at the specifications of **Midori64** and **Midori128**, such as: S-Boxes used, Round Functions, Key Generation Algorithm etc.
- We looked at the general structure of Midori and the encryption/decryption algorithms involved.
- We also compared Optimal Cell Permutation and Shift-Rows operations, where we concluded that Optimal Cell Permutation has more number of Active S-Boxes.

Conclusion

- Next, we did linear and differential cryptanalysis and concluded that Midori has enough number of rounds to prevent both linear and differential attacks.
- We also did the Integral analysis of up to 3 Rounds of Midori and saw that Key Recovery Attack can only be performed for up to 5 Rounds.
- Lastly, we briefly touched upon the topic of **Fault Analysis** and the methods involved for Fault Analysis.

Thanks

Team Members

- Shreyas K Shastry
- Anuj Verma
- Naimat Ali Khan

Implementation Info

- Github Link: Click Here