# PRINT Cipher

**Berkay Akçören**

MIDDLE EAST TECHNICAL UNIVERSITY

Informatics Institute, Department of Cyber Security

*Last Modified:* December 18, 2024
*Created: December 18, 2024*

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.
- **Block Size:** 48 or 96 bits

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.
- **Block Size:** 48 or 96 bits
- **Effective Key Length:** $\frac{5}{3} \times \text{Block Size} = SK_1 || SK_2$

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.
- **Block Size:** 48 or 96 bits
- **Effective Key Length:** $\frac{5}{3} \times$ Block Size $= SK_1 || SK_2$
- **Rounds:** 48 or 96

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.
- **Block Size:** 48 or 96 bits
- **Effective Key Length:** $\frac{5}{3} \times$ Block Size $= SK_1||SK_2$
- **Rounds:** 48 or 96
- There is no key schedule, constant round key

# PRINT Cipher

## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.
- **Block Size:** 48 or 96 bits
- **Effective Key Length:** $\frac{5}{3} \times \text{Block Size} = SK_1 || SK_2$
- **Rounds:** 48 or 96
- There is no key schedule, constant round key

## Security Goals

- Side-channel attacks are not considered

# PRINT Cipher

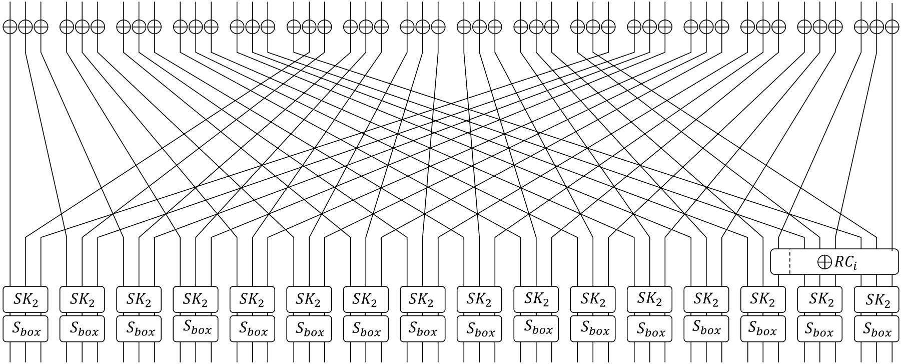
## General Information

- Designed by Lars Knudsen, Gregor Leander, Axel Poschmann and Matthew J.B. Robshaw published in 2010
- A lightweight design specifically for IC Printing.
- **Block Size:** 48 or 96 bits
- **Effective Key Length:** $\frac{5}{3} \times \text{Block Size} = SK_1 || SK_2$
- **Rounds:** 48 or 96
- There is no key schedule, constant round key

## Security Goals

- Side-channel attacks are not considered
- Related-key attacks are not considered

# PRINT Cipher



# PRINT Cipher: Round Function

## General Information

- **Key Addition:** Cipher state `xor` with round key $SK_1$

# PRINT Cipher: Round Function

## General Information

- **Key Addition:** Cipher state `xor` with round key $SK_1$
- **Linear Diffusion:** Cipher state shuffled with fixed permutation layer

# PRINT Cipher: Round Function

## General Information

- **Key Addition:** Cipher state `xor` with round key $SK_1$
- **Linear Diffusion:** Cipher state shuffled with fixed permutation layer
- **Round Counter addition $RC_i$:** Round constant addition with bitwise `xor`

# PRINT Cipher: Round Function

## General Information

- **Key Addition:** Cipher state `xor` with round key $SK_1$
- **Linear Diffusion:** Cipher state shuffled with fixed permutation layer
- **Round Counter addition $RC_i$:** Round constant addition with bitwise `xor`
- **Keyed Permutation:** Permutation operation on each 3 bit dependent on $SK_2$

# PRINT Cipher: Round Function

## General Information

- **Key Addition:** Cipher state `xor` with round key $SK_1$
- **Linear Diffusion:** Cipher state shuffled with fixed permutation layer
- **Round Counter addition $RC_i$:** Round constant addition with bitwise `xor`
- **Keyed Permutation:** Permutation operation on each 3 bit dependent on $SK_2$
- **$S_{box}$ Layer:** Non-linear $S_{box}$

# PRINT Cipher: Round Function

## General Information

- **Key Addition:** Cipher state `xor` with round key $SK_1$
- **Linear Diffusion:** Cipher state shuffled with fixed permutation layer
- **Round Counter addition $RC_i$:** Round constant addition with bitwise `xor`
- **Keyed Permutation:** Permutation operation on each 3 bit dependent on $SK_2$
- $S_{box}$ **Layer:** Non-linear $S_{box}$

# PRINT Cipher: Round Function

## Linear Diffusion Layer

Simple permutation defined as

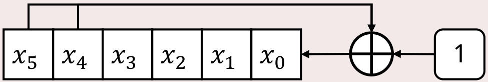
$$P(i) = \begin{cases} 3 \times i \bmod b-1 & \text{for} \quad 0 \geq i \geq b-2, \\ b-1 & \text{for} \quad i = b-1. \end{cases}$$

# PRINT Cipher: Round Function

## Round Counter Addition

Round counters are generated using $n$-bit LFSR in the following way

$$t = 1 + x_{n-1} + x_{n-2}$$
$$x_i = x_{i-1} \qquad \text{for } n-1 \geq i \geq 1$$
$$x_0 = t$$



# PRINT Cipher: $S_{box}$ and Keyed Permutation

## $S_{box}$

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $S[x]$ | 0 | 1 | 3 | 6 | 7 | 4 | 5 | 2 |

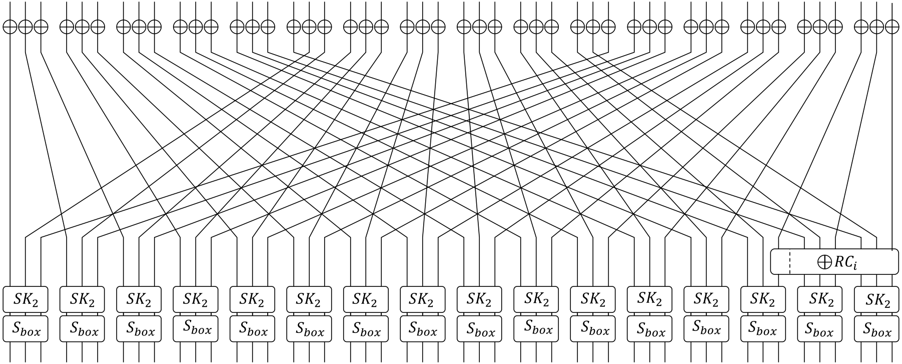
## Permuted $S_{box}$

- $K = SK_1 || SK_2$
- $SK_1$ is $b$-bit long
- $SK_2$ is $\frac{2}{3}b$-bit long, which is $\frac{b}{3}$ pair bits.
- Change the order of the 3 input bits for different values of $a_1||a_2$ in $SK_2$

| $a_1 \| a_0$ | |
|---|---|
| 00 | $c_2 \| c_1 \| c_0$ |
| 01 | $c_1 \| c_2 \| c_0$ |
| 10 | $c_2 \| c_0 \| c_1$ |
| 11 | $c_0 \| c_1 \| c_2$ |

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $V_0[x]$ | 0 | 1 | 3 | 6 | 7 | 4 | 5 | 2 |
| $V_1[x]$ | 0 | 1 | 7 | 4 | 3 | 5 | 6 | 2 |
| $V_2[x]$ | 0 | 3 | 1 | 6 | 7 | 5 | 4 | 2 |
| $V_3[x]$ | 0 | 7 | 3 | 5 | 1 | 4 | 6 | 2 |

# PRINT Cipher



# PRINT Cipher: Round Function

**Example single round encryption for $b = 48$-bit block size**

1. **Pick key:** $SK_1||SK_2$ which is $48 + 32$ bits long.

# PRINT Cipher: Round Function

**Example single round encryption for $b = 48$-bit block size**

1. **Pick key:** $SK_1 || SK_2$ which is $48 + 32$ bits long.
2. **Key `xor`:** $SK_1 \oplus$ STATE

# PRINT Cipher: Round Function

**Example single round encryption for $b = 48$-bit block size**

1. **Pick key:** $SK_1||SK_2$ which is $48 + 32$ bits long.
2. **Key** `xor`**:** $SK_1 \oplus \text{STATE}$
3. **Linear diffusion:** Move bits around

# PRINT Cipher: Round Function

**Example single round encryption for $b = 48$-bit block size**

1. **Pick key:** $SK_1||SK_2$ which is $48 + 32$ bits long.
2. **Key** `xor`**:** $SK_1 \oplus$ STATE
3. **Linear diffusion:** Move bits around
4. **Round counter:** $RC_i \oplus$ STATE

# PRINT Cipher: Round Function

## Example single round encryption for $b = 48$-bit block size

1. **Pick key:** $SK_1||SK_2$ which is $48 + 32$ bits long.
2. **Key `xor`:** $SK_1 \oplus$ STATE
3. **Linear diffusion:** Move bits around
4. **Round counter:** $RC_i \oplus$ STATE
5. **Keyed permutation:** $SK_2 = 32$ bit key. Change the order of the bits OR determine $S_{box}$ for each pair bit

# PRINT Cipher: Round Function

## Example single round encryption for $b = 48$-bit block size

1. **Pick key:** $SK_1 || SK_2$ which is $48 + 32$ bits long.
2. **Key** `xor`**:** $SK_1 \oplus$ STATE
3. **Linear diffusion:** Move bits around
4. **Round counter:** $RC_i \oplus$ STATE
5. **Keyed permutation:** $SK_2 = 32$ bit key. Change the order of the bits OR determine $S_{box}$ for each pair bit
6. $S_{box}$ **layer:** Regular $S_{box}$ operation.

# Example Case of Linear Cryptanalysis

## 4-bit Block Example

- 4-bit input plaintext is encrypted into 4-bit output ciphertext

$$p_3p_2p_1p_0 \xrightarrow{ENC} c_3c_2c_1c_0$$

# Example Case of Linear Cryptanalysis

## 4-bit Block Example

- 4-bit input plaintext is encrypted into 4-bit output ciphertext

$$p_3p_2p_1p_0 \xrightarrow{ENC} c_3c_2c_1c_0$$

- We try to find a relation between arbitrary input and output bits

$$P(p_2 \oplus p_1 \oplus c_0 = 1) = \frac{1}{2} \pm \epsilon$$

- For a random permutation $\epsilon = 0$

# Observation on $S_{box}$

| Perm. Key | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| $(c_2, c_1, c_0)$ | $S(c_2, c_1, c_0)$ | $S(c_1, c_2, c_0)$ | $S(c_2, c_0, c_1)$ | $S(c_0, c_1, c_2)$ |
| $(0,0,0)$ | $(0,0,0)$ | $(0,0,0)$ | $(0,0,0)$ | $(0,0,0)$ |
| $(0,0,1)$ | $(0,0,1)$ | $(0,0,1)$ | $(0,1,1)$ | $(1,1,1)$ |
| $(0,1,0)$ | $(0,1,1)$ | $(1,1,1)$ | $(0,0,1)$ | $(0,1,1)$ |
| $(0,1,1)$ | $(1,1,0)$ | $(1,0,0)$ | $(1,1,0)$ | $(1,0,1)$ |
| $(1,0,0)$ | $(1,1,1)$ | $(0,1,1)$ | $(1,1,1)$ | $(0,0,1)$ |
| $(1,0,1)$ | $(1,0,0)$ | $(1,1,0)$ | $(1,0,1)$ | $(1,0,0)$ |
| $(1,1,0)$ | $(1,0,1)$ | $(1,0,1)$ | $(1,0,0)$ | $(1,1,0)$ |
| $(1,1,1)$ | $(0,1,1)$ | $(0,1,1)$ | $(0,1,1)$ | $(0,1,1)$ |

# Possible keys for bit rotations

| Bit Move | Possible Keys |
|---|---|
| $c_0 \to c_0$ | $(0,0),(0,1)$ |
| $c_1 \to c_0$ | $(1,0)$ |
| $c_2 \to c_0$ | $(1,1)$ |
| $c_0 \to c_1$ | $(1,0)$ |
| $c_1 \to c_1$ | $(0,0),(1,1)$ |
| $c_2 \to c_1$ | $(0,1)$ |
| $c_0 \to c_2$ | $(1,1)$ |
| $c_1 \to c_2$ | $(0,1)$ |
| $c_2 \to c_2$ | $(0,0),(1,0)$ |

# Attack Idea

## Attack Idea

- Assume permutation bit at left-most $S_{box}$ as $SK_2^{(31,30)} = (*0)$
- 2 out of 4 keys this happens
- Probability of left-most bit remains unaltered is $\frac{3}{4}$
- After 1 round of encryption

$$P(c_{47} = p_{47} \oplus SK_1^{47}) = \frac{1}{2} + 2^{-2}$$

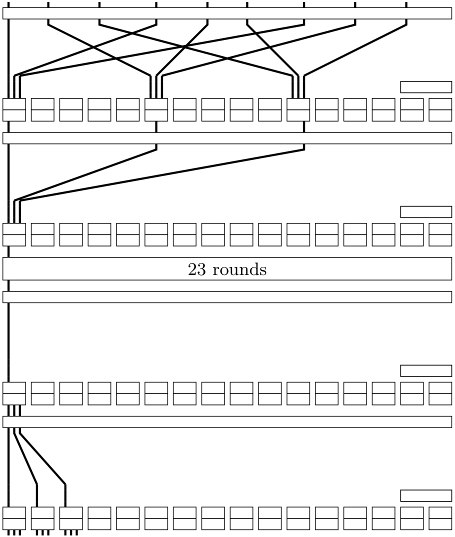
- After 2 rounds of encryption

$$P(c_{47} = p_{47}) = \frac{1}{2} + 2^{-3}$$

- After $r$ rounds of encryption for even $r$

$$P(c_{47} = p_{47}) = \frac{1}{2} + 2^{-r-1}$$

# Extended 25-Round Attack



# Extended 25-Round Attack

## Attack Idea

- Assume $SK_2^{30} = 0$
- Guess $SK_1^{(47,42,37,31,26,21,15,10,5)}$ and $SK_2^{(21,20,19,3)}$ for encryption
- Guess $SK_1^{(47,46,45)}$ and $SK_2^{(18,16)}$ for decryption
- Total of $2^{13} \times 3^3 \approx 2^{17.8}$
- 2 Rounds of encryption and decryption
- If $c_{47}^{enc} = p_{47}^{enc} \oplus SK_1^{47}$ increase the counter of the guess
- Highest counter assumed to be correct guess.

# Results

## Other Attacks with Different Trials

All attacks require collection of the whole codebook, $2^{48}$ plaintext-ciphertext pairs.

| Rounds | # of enc/dec |
|---|---|
| 27 | $2^{36}$ |
| 28 | $2^{67}$ |
| 29 | $2^{76}$ |

# Differentials in Print Cipher

## Difficulties

- Main technical problem is differentials are Key-dependent
- Without knowing the key, one cannot find the best differential

# Differential Distribution Table

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $S[x]$ | 0 | 1 | 3 | 6 | 7 | 4 | 5 | 2 |

| | $0_x$ | $1_x$ | $2_x$ | $3_x$ | $4_x$ | $5_x$ | $6_x$ | $7_x$ |
|---|---|---|---|---|---|---|---|---|
| $0_x$ | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $1_x$ | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 |
| $2_x$ | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 2 |
| $3_x$ | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 0 |
| $4_x$ | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| $5_x$ | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |
| $6_x$ | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 |
| $7_x$ | 0 | 2 | 2 | 0 | 2 | 0 | 0 | 2 |

# Two Differential Attacks

## Optimal Characteristic

- There exist a 1-bit to 1-bit difference in every bit location with probability $\frac{1}{4}$
- For $r$ many rounds, there is at least one differential with of one with the probability $(1/4)^r$
- For $r = 22$ rounds, one can successfully construct a distinguisher with probability $2^{-44}$

## Obtaining the roots of PRINT Cipher's permutation layer

- Constructing a 22 round distinguisher requires full codebook i.e. $2^{48}$ plaintext-ciphertext pairs.
- Attacker can form $2^{47}$ plaintext pairs for every 1-bit difference
- Therefore, attacker can learn the permutation $PK^r$ for $r = 22$ rounds.
- If one can somehow find the roots of permutations i.e. $PK$ by looking at the $RK^r$, get the permutation key $SK_2$ and then get the $SK_1$

# Roots of Permutations

## Example Case

- $(1, 2, 3, 4, 5)$ is mapped on to $(4, 5, 2, 3, 1)$

# Roots of Permutations

## Example Case

- $(1, 2, 3, 4, 5)$ is mapped on to $(4, 5, 2, 3, 1)$
- $(1, 2, 3, 4, 5)$ is mapped on to $(2, 4, 1, 5, 3)$

# Roots of Permutations

## Example Case

- $(1, 2, 3, 4, 5)$ is mapped on to $(4, 5, 2, 3, 1)$
- $(1, 2, 3, 4, 5)$ is mapped on to $(2, 4, 1, 5, 3)$
- $(2, 4, 1, 5, 3)$ is a square root of $(4, 5, 2, 3, 1)$

# Roots of Permutations

## Example Case

- $(1,2,3,4,5)$ is mapped on to $(4,5,2,3,1)$
- $(1,2,3,4,5)$ is mapped on to $(2,4,1,5,3)$
- $(2,4,1,5,3)$ is a square root of $(4,5,2,3,1)$
- $(1,2,3,4,5) \Rightarrow (2,4,1,5,3) \Rightarrow (4,5,2,3,1)$

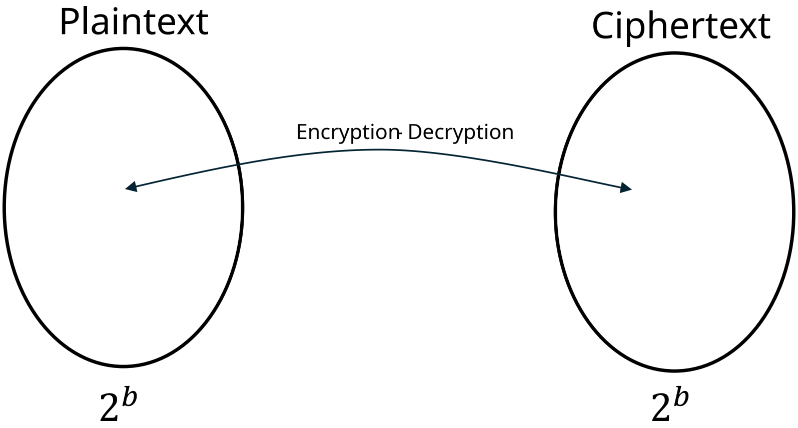
# Results

### Two differential attacks

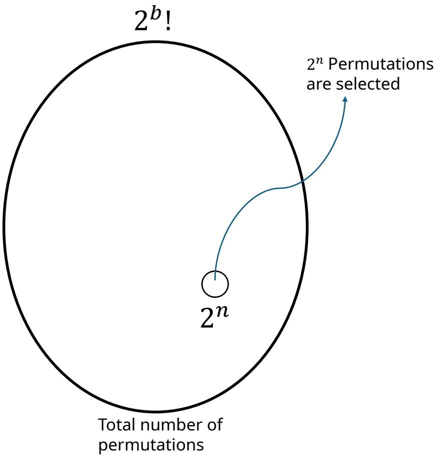
- Attacks require collection of the whole codebook, $2^{48}$ plaintext-ciphertext pairs
- Only able to break 22 rounds of the cipher

# Block Cipher

- b-bit block

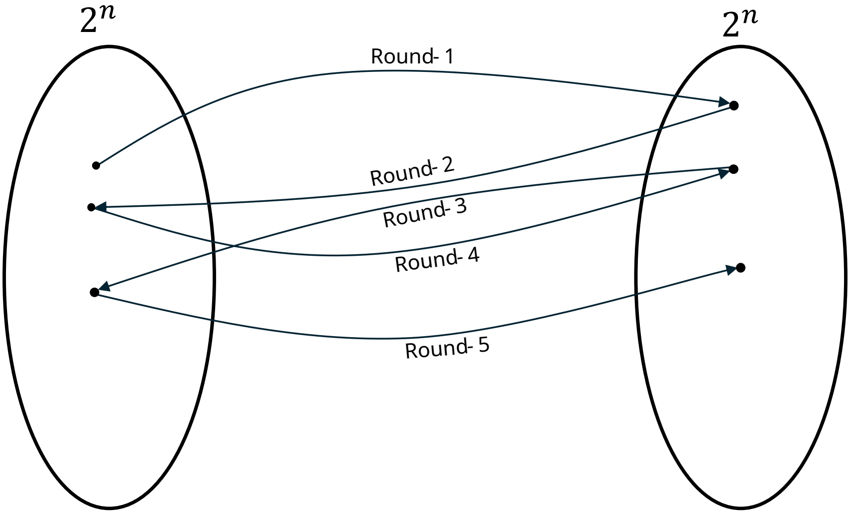


# Block Cipher

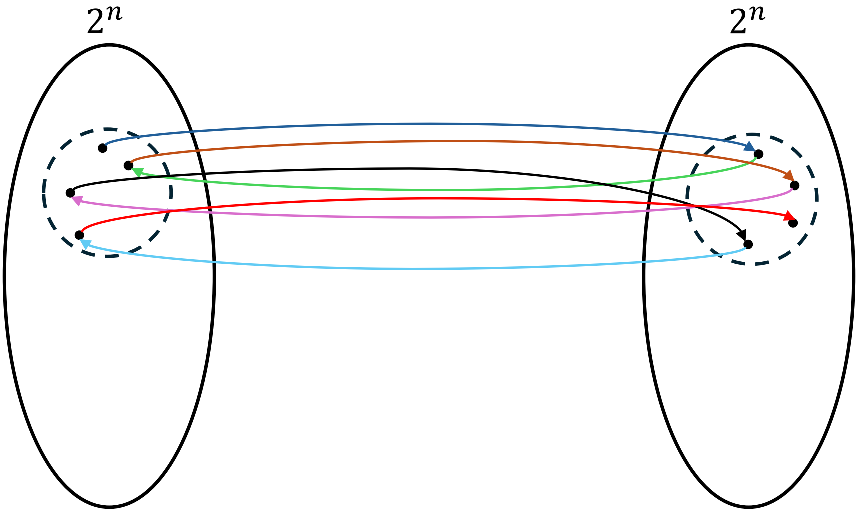



- b-bit block
- n-bit key

# Between Round Functions Inside a Block Cipher



# Between Round Functions Inside a Invariant Subspace



# Round Function

## Round Function Depends on

- Key `xor`

# Round Function

## Round Function Depends on

- Key `xor`
- Linear Diffusion

# Round Function

**Round Function Depends on**

- Key `xor`
- Linear Diffusion
- Round Counter addition

# Round Function

## Round Function Depends on

- Key `xor`
- Linear Diffusion
- Round Counter addition
- Keyed Permutation

# Round Function

## Round Function Depends on

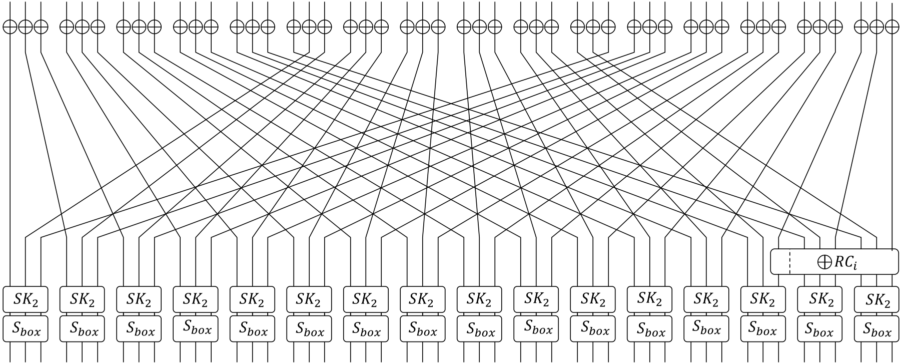
- Key `xor`
- Linear Diffusion
- Round Counter addition
- Keyed Permutation
- $(S_{box})$ Layer

# Round Function

## Round Function Depends on

- Key `xor`
- Linear Diffusion
- Round Counter addition
- Keyed Permutation
- $(S_{box})$ Layer

# Round Function



# Round Function

## Round Function Depends on

- Key $\texttt{xor}$ $(SK_1) \leftarrow$ KEY DEPENDENT
- Linear Diffusion $(P)$
- Round Counter addition $(RC)$
- Keyed Permutation $(SK_2) \leftarrow$ KEY DEPENDENT
- $(S_{box})$ Layer

# Round Function

## Round Function Depends on

- Key `xor` $(SK_1) \leftarrow$ KEY DEPENDENT
- Linear Diffusion $(P)$
- Round Counter addition $(RC)$
- Keyed Permutation $(SK_2) \leftarrow$ KEY DEPENDENT
- $(S_{box})$ Layer

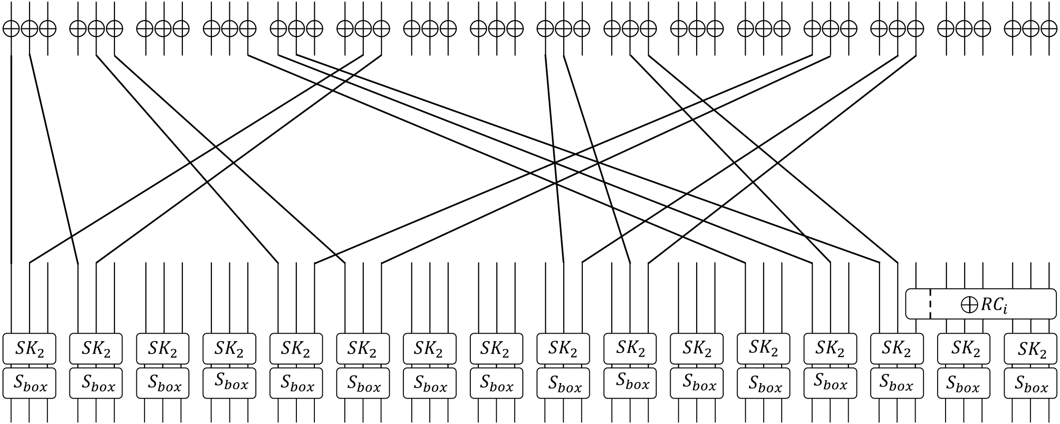
$$R = \hat{R}(SK_1, P, RC, SK_2, S_{box})$$

# S-Box

## Undisturbed Bits

$$
\begin{array}{rcl}
S(000) = 000 & & \\
S(001) = 001 & \Leftrightarrow & S(00{*}) = 00{*} \\
\hline
S(100) = 111 & & \\
S(110) = 101 & \Leftrightarrow & S(1{*}0) = 1{*}0 \\
\hline
S(011) = 110 & & \\
S(111) = 010 & \Leftrightarrow & S({*}11) = {*}10 \\
\hline
\end{array}
$$

# S-Boxes mapping to themselves



# Example Iterative Round

- Xor key = 01* *11 *** *** 01* *11 *** *** 01* *11 *** *** 01* *11 *** ***
- Perm. key = 0* 11 ** ** 10 01 ** ** 11 *0 ** ** *0 11 ** **

| | |
|---|---|
| Start | 00* *10 *** *** 00* *10 *** *** 00* *10 *** *** 00* *10 *** *** |
| Key xoring | 01* *01 *** *** 01* *01 *** *** 01* *01 *** *** 01* *01 *** *** |
| Lin. layer | 00* 11* *** *** 0*0 1*1 *** *** *00 *11 *** *** 00* 11* *** *** |
| RC | 00* 11* *** *** 0*0 1*1 *** *** *00 *11 *** *** 00* 11* *** *** |
| Perm. layer | 00* *11 *** *** 00* *11 *** *** 00* *11 *** *** 00* *11 *** *** |
| S-box layer | 00* *10 *** *** 00* *10 *** *** 00* *10 *** *** 00* *10 *** *** |

# Weak Keys

## Weak Keys

- $2^{-16}$ XOR keys
- $2^{-13}$ permutations keys
- $2^{51}$ weak keys out of $2^{80}$ total keys

# Weak Keys

## Weak Keys

- $2^{-16}$ XOR keys
- $2^{-13}$ permutations keys
- $2^{51}$ weak keys out of $2^{80}$ total keys
- This attack is independent of number of rounds!
- Distinguisher for any number of rounds

## PRINT Cipher - 96 bit block

- $2^{101}$ weak keys out of $2^{160}$ total keys

# Protection Against the Attack

## Remedies

- Spread the round function $RC_i$ to last 3 $S_{box}$
- 2-bits in each $S_{box}$ without any extra hardware cost