The 23rd Annual International Conference on Information Security and Cryptology

ICISC 2020

December 2 (Wed) ~ December 4 (Fri), 2020 | Virtual Conference

Hosted by
Korea Institute of Information Security and Cryptology (KIISC)
National Security Research Institute (NSR)

Korea Institute of Information Security & Cryptology

PIPO: A Lightweight Block cipher with Efficient Higher-Order Masking Software Implementations

Hangi Kim, Yongjin Jeon, Giyoon Kim, Jongsung Kim, Bo-Yeon Sim, Dong-Guk Han, Hwajeong Seo, Seonggyeom Kim, Seokhie Hong, Jaechul Sung, and Deukjo Hong

Kookmin University, Hansung University, Korea University, University of Seoul and Chonbuk National University

Introduction (Motivations)

- Although a block cipher is secure to the classical cryptanalysis, it is necessary to apply the side-channel countermeasures.
- Increasing environments that requiring side-channel countermeasures.
- There are many lightweight block ciphers proposed, but there are very few block ciphers considering the efficiency of implementing higher-order Masking while simultaneously having excellent S/W and H/W implementation performance.

Introduction (Key Considerations of PIPO)

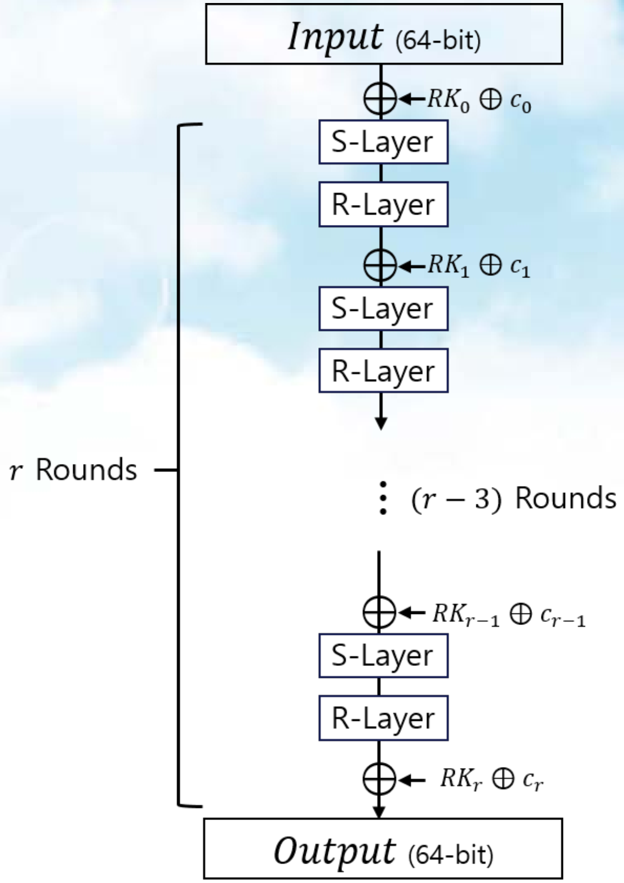
- Side-channel countermeasure applied environment (Plug-In, **PI**)
  - The less the number of nonlinear operation is used, the less reduction in efficiency when applying the side-channel countermeasure technique.
    - Linear operations: $O(d)$, nonlinear operations: $O(d^2)$ (d is the number of operations)
- General S/W, H/W implementation environment (Plug-Out, **PO**)
  - Design for application in ultra-light environments
    - Execution time, RAM, Area, etc.

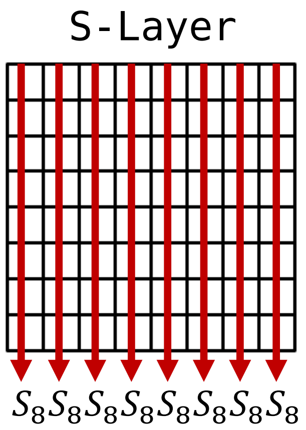
Introduction (Contributions)

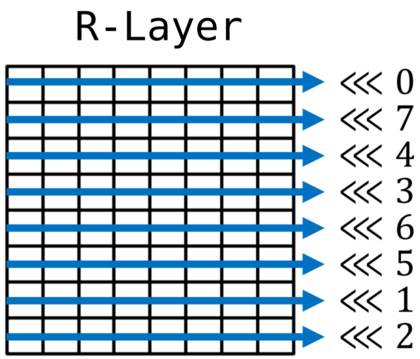
- New lightweight 8-bit S-box
  - It offers an efficient bitsliced implementation including only 11 nonlinear bitwise operations.
  - Both DBN and LBN are 3.
- PIPO can be implemented using fewer nonlinear operations than other block ciphers.
- PIPO has excellent performance on S/W and H/W implementations

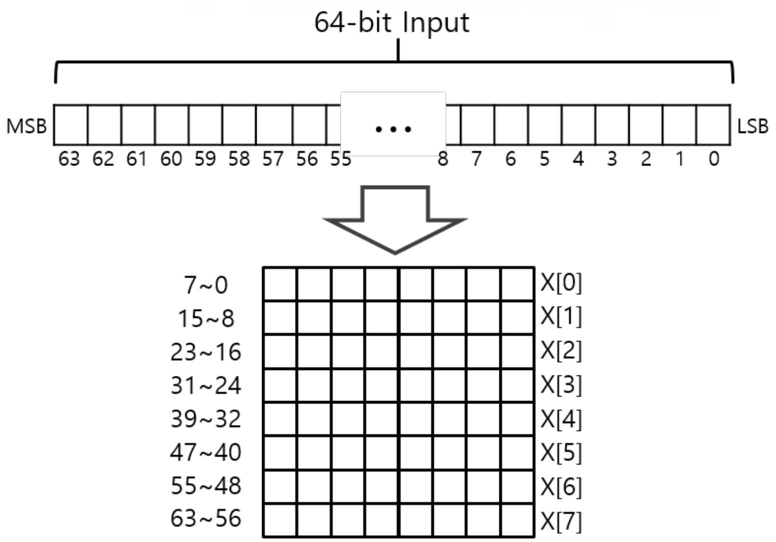
Specification of PIPO

- SbPN (S-box bit-Permutation Network) structure
  - Using 8-bit S-box, 8-bit rotations



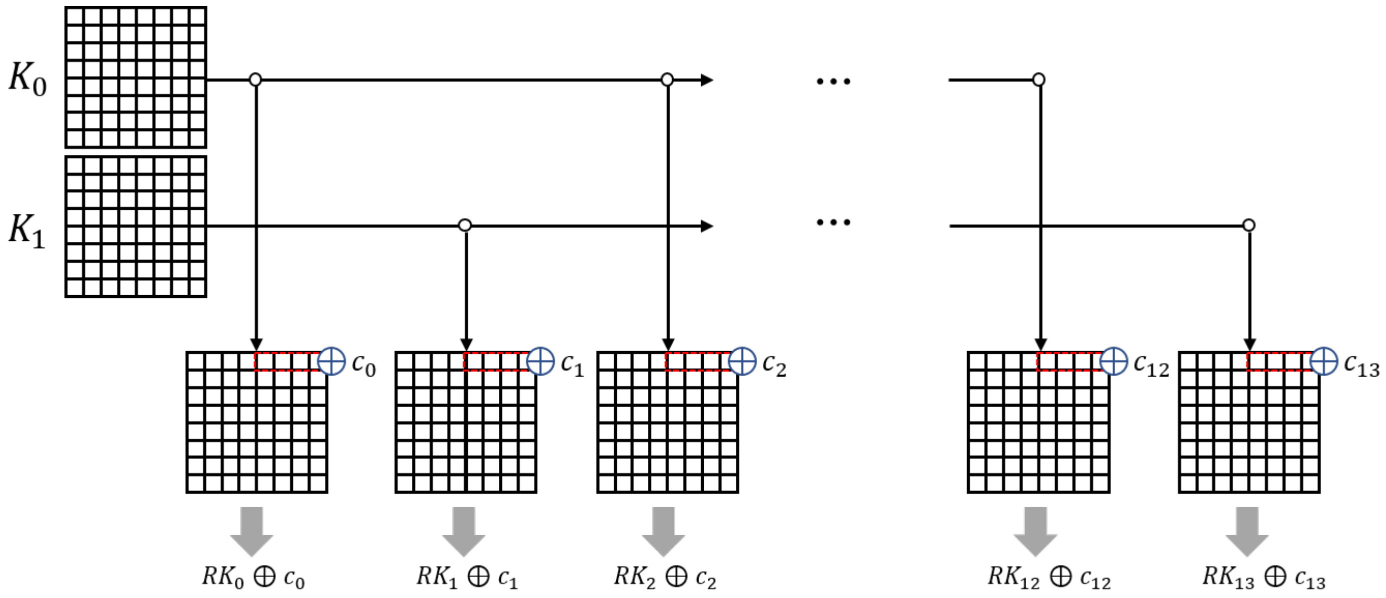






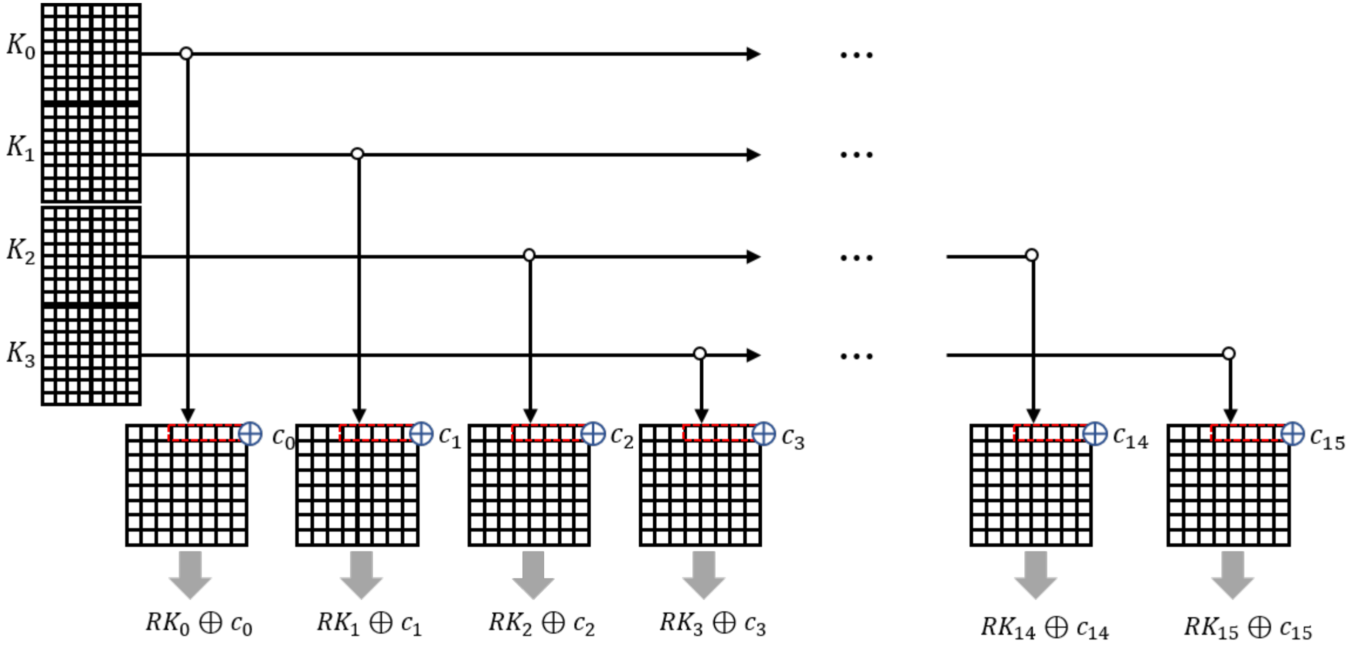
Specification of PIPO

- Key schedules (128-bit key)
  - $K = (K_1 || K_0)$, K_0, K_1 are 64-bit respectively, K_0 is the lower 64-bit.
  - Use $RK_0 \sim RK_{13}$ $(RK_i = K_{i\ (mod\ 2)})$
  - $c_i = i$



Specification of PIPO

- Key schedules (256-bit key)
  - $K = (K_3||K_2||K_1||K_0)$, K_0, K_1, K_2, K_3 are 64-bit respectively, K_0 is the lower 64-bit.
  - Use $RK_0 \sim RK_{15}$ ($RK_i = K_{i\ (mod\ 4)}$)
  - $c_i = i$



Specification of PIPO

- New lightweight S-box, S_8

Right (low-order) 4-bit

Left (high-order) 4-bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x5E	0xF9	0xFC	0x00	0x3F	0x85	0xBA	0x5B	0x18	0x37	0xB2	0xC6	0x71	0xC3	0x74	0x9D
1	0xA7	0x94	0x0D	0xE1	0xCA	0x68	0x53	0x2E	0x49	0x62	0xEB	0x97	0xA4	0x0E	0x2D	0xD0
2	0x16	0x25	0xAC	0x48	0x63	0xD1	0xEA	0x8F	0xF7	0x40	0x45	0xB1	0x9E	0x34	0x1B	0xF2
3	0xB9	0x86	0x03	0x7F	0xD8	0x7A	0xDD	0x3C	0xE0	0xCB	0x52	0x26	0x15	0xAF	0x8C	0x69
4	0xC2	0x75	0x70	0x1C	0x33	0x99	0xB6	0xC7	0x04	0x3B	0xBE	0x5A	0xFD	0x5F	0xF8	0x81
5	0x93	0xA0	0x29	0x4D	0x66	0xD4	0xEF	0x0A	0xE5	0xCE	0x57	0xA3	0x90	0x2A	0x09	0x6C
6	0x22	0x11	0x88	0xE4	0xCF	0x6D	0x56	0xAB	0x7B	0xDC	0xD9	0xBD	0x82	0x38	0x07	0x7E
7	0xB5	0x9A	0x1F	0xF3	0x44	0xF6	0x41	0x30	0x4C	0x67	0xEE	0x12	0x21	0x8B	0xA8	0xD5
8	0x55	0x6E	0xE7	0x0B	0x28	0x92	0xA1	0xCC	0x2B	0x08	0x91	0xED	0xD6	0x64	0x4F	0xA2
9	0xBC	0x83	0x06	0xFA	0x5D	0xFF	0x58	0x39	0x72	0xC5	0xC0	0xB4	0x9B	0x31	0x1E	0x77
A	0x01	0x3E	0xBB	0xDF	0x78	0xDA	0x7D	0x84	0x50	0x6B	0xE2	0x8E	0xAD	0x17	0x24	0xC9
B	0xAE	0x8D	0x14	0xE8	0xD3	0x61	0x4A	0x27	0x47	0xF0	0xF5	0x19	0x36	0x9C	0xB3	0x42
C	0x1D	0x32	0xB7	0x43	0xF4	0x46	0xF1	0x98	0xEC	0xD7	0x4E	0xAA	0x89	0x23	0x10	0x65
D	0x8A	0xA9	0x20	0x54	0x6F	0xCD	0xE6	0x13	0xDB	0x7C	0x79	0x05	0x3A	0x80	0xBF	0xDE
E	0xE9	0xD2	0x4B	0x2F	0x0C	0xA6	0x95	0x60	0x0F	0x2C	0xA5	0x51	0x6A	0xC8	0xE3	0x96
F	0xB0	0x9F	0x1A	0x76	0xC1	0x73	0xC4	0x35	0xFE	0x59	0x5C	0xB8	0x87	0x3D	0x02	0xFB

8-bit S-box Table

```c
//(MSb: x[7], LSb: x[0]) :"b" represents bit
// Input: x[7], x[6], x[5], x[4], x[3], x[2], x[1], x[0]
// S5_1
x[5] ^= (x[7] & x[6]);
x[4] ^= (x[3] & x[5]);
x[7] ^= x[4];
x[6] ^= x[3];
x[3] ^= (x[4] | x[5]);
x[5] ^= x[7];
x[4] ^= (x[5] & x[6]);
// S3
x[2] ^= x[1] & x[0];
x[0] ^= x[2] | x[1];
x[1] ^= x[2] | x[0];
x[2] = ~x[2];
// Extend XOR
x[7] ^= x[1]; x[3] ^= x[2]; x[4] ^= x[0];
//S5_2
t[0] = x[7]; t[1] = x[3]; t[2] = x[4];
x[6] ^= (t[0] & x[5]);
t[0] ^= x[6];
x[6] ^= (t[2] | t[1]);
t[1] ^= x[5];
x[5] ^= (x[6] | t[2]);
t[2] ^= (t[1] & t[0]);
// truncate XOR and swap
x[2] ^= t[0]; t[0] = x[1] ^ t[2]; x[1] = x[0]^t[1];
x[0] = x[7]; x[7] = t[0];
t[1] = x[3]; x[3] = x[6]; x[6] = t[1];
t[2] = x[4]; x[4] = x[5]; x[5] = t[2];
// Output: x[7], x[6], x[5], x[4], x[3], x[2], x[1], x[0]
```

The bitsliced implementation of the S_8 (in C code)

Specification of PIPO

- Advantages of S_8
  - Bitslice implementation
  - Small number of nonlinear operations
    - Efficient high-order Masking
  - Both DBN and LBN are 3
    - Secure cryptographic security

Comparison of bitslice 8-bit S-boxes

Blockcipher	PIPO	FLY	Fantomas	Robin	LILLIPUT
Differential uniformity	**16**	16	16	16	8
DBN	**3**	3	2	2	2
Non-linearity	**96**	96	96	96	96
LBN	**3**	3	2	2	2
Algebraic degree	**5**	5	5	6	6
#(Fixed points)	**0**	2	0	16	1
#(Nonlinear operations)	**11**	12	11	12	12
#(Linear operations)	**23**	24	27	24	27
Construction method	***U-bridge**	Lai-Massey	*U-MISTY	MISTY	Feistel
Reference	**This paper**	[41]	[35]	[35]	[1]

*'U-' represents 'Unbalanced-'.
**Nonlinear (resp. linear) operations represent AND, OR (resp. XOR, NOT).

Design Rationales of S-box (three criteria of S_8)

1. It should lower an efficient bitsliced implementation including 11 or fewer nonlinear operations.
2. Its differential and linear branch numbers (DBN and LBN) should both be greater than 2.
3. Its differential uniformity should be 16 or less, and its non-linearity should be 96 or more.

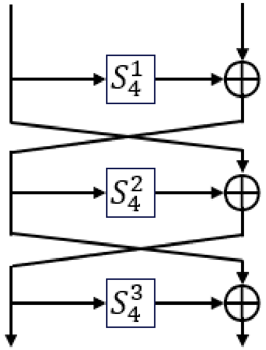
- Additional conditions: No fixed point, less linear operations

Design Rationales of S-box (searching method)

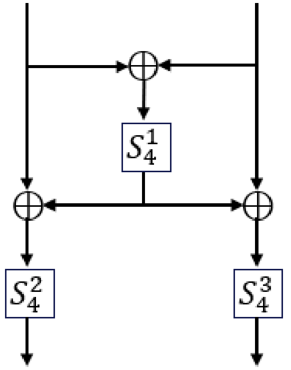
- Algebraic method
  - Cryptographic Security guaranteed
  - Difficult to find efficient bitsliced implementation
- Derive 8-bit S-box from small S-boxes using Structure
  - Bitslice implementation of 8-bit S-box can be derived from bitslice implementation of small S-box!
  - Secure cryptographic security by using 3 or more small S-boxes.

Design Rationales of S-box (searching method)

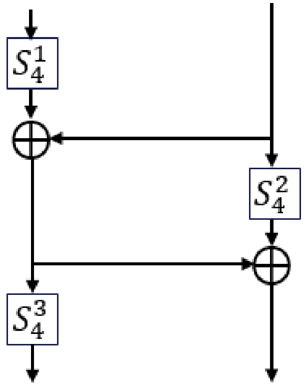
- Limitations of Feistel, Lai-Massey, and MISTY Structures
  - 4-bit S-box must use 4 or more nonlinear operations
    - Below 4 nonlinear operations, differential uniformity 4 and non-linearity 4 cannot be satisfied.
  - If three 4-bit S-boxes are used, Criterion 1 of of S_8 is not satisfied
    - In order to satisfy criterion 3 (DC/LC security), criterion 1 (number of nonlinear operations) will be violated.



Feistel



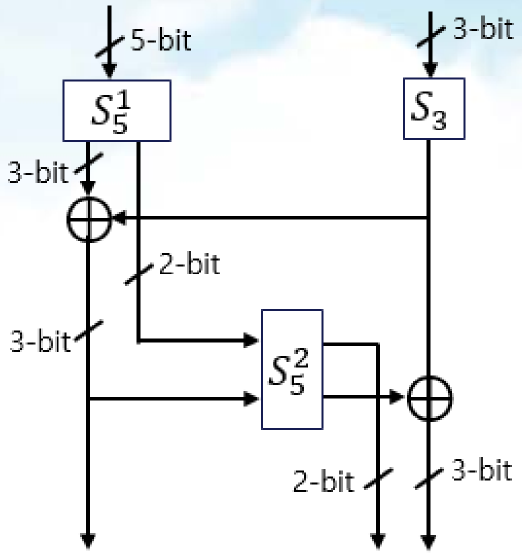
Lai-Massey



MISTY

Design Rationales of S-box (searching method)

- Unbalanced-Bridge structure
  - Using 3-bit, 5-bit S-boxes
    - 3-bit S-box: 3 nonlinear operations
    - 5-bit S-box: 4 nonlinear operations
    - ➢ 3+4+4=11 nonlinear operations
  - Other advantages
    - S_5^2 can be nonbijective S-box
    - The number of bit-XORs used for the structure is 6, which is relatively small



Unbalanced-Bridge structure

Design Rationales of S-box (searching method)

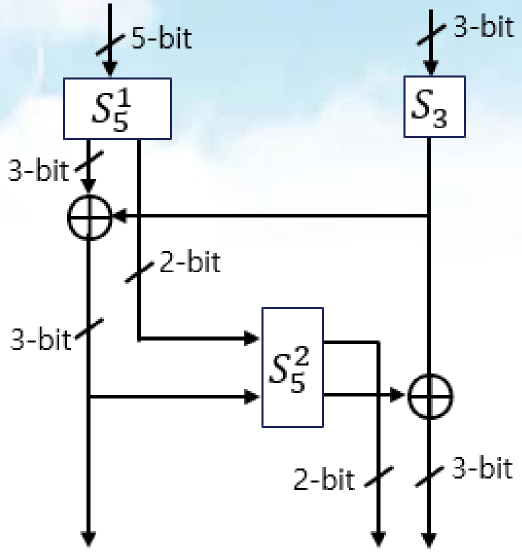
- Theorem for DBN 3

Theorem 1. *The DBN of bijective 8-bit S-boxes constructed using the unbalanced-Bridge is greater than 2 if and only if conditions i), ii), and iii) are all satisfied ($\Delta\alpha$ and $\Delta\beta$ below represent arbitrary differences where $wt(\Delta\alpha) = wt(\Delta\beta) = 1$):*

i) For each $\Delta\alpha, \Delta\beta \in \mathbb{F}_2^3$, at least one of the entry $(\Delta\alpha, \Delta\beta)$ in DDT of S_3 and the entry $(\Delta\beta||0^{(2)}, \Delta\beta||0^{(2)})$ in DDT of S_5^2 is 0,

ii) For each $\Delta\alpha, \Delta\beta \in \mathbb{F}_2^5$, for each $A, B(\neq A) \in \mathbb{F}_2^2$, at least one of $\mathfrak{F}_A^1(X) \oplus \mathfrak{F}_B^1(X) = \Delta\alpha$ and $\mathfrak{F}_A^2(X) \oplus \mathfrak{F}_B^2(X) = \Delta\beta$ has no solution X, where $X \in \mathbb{F}_2^3$,

iii) For each $\Delta\alpha \in \mathbb{F}_2^3$ and $\Delta\beta \in \mathbb{F}_2^5$, for each $A, B \in \mathbb{F}_2^2$, at least one of $\mathfrak{F}_A^1(X) \oplus \mathfrak{F}_B^1(X \oplus \Delta\alpha) = \Delta\beta$ and $\mathfrak{F}_A^2(X) \oplus \mathfrak{F}_B^2(X \oplus \Delta\alpha) = \Delta 0$ has no solution X, where $X \in \mathbb{F}_2^3$.



Unbalanced-Bridge structure

$$\mathfrak{F}_A^1 : \mathbb{F}_2^3 \to \mathbb{F}_2^5,\ \ \mathfrak{F}_A^1(X) = (S_5^1)^{-1}(X||A)\ \text{ for } A \in \mathbb{F}_2^2,$$
$$\mathfrak{F}_A^2 : \mathbb{F}_2^3 \to \mathbb{F}_2^5,\ \ \mathfrak{F}_A^2(X) = S_5^2(X||A)\ \text{ for } A \in \mathbb{F}_2^2.$$

Design Rationales of S-box (searching method)

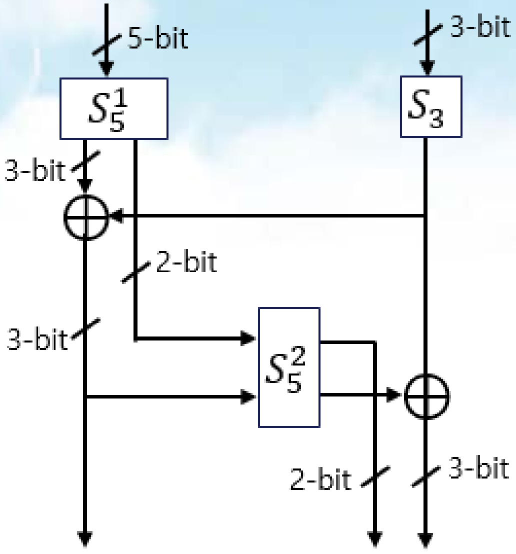
- Theorem for LBN 3

Theorem 2. *The LBN of bijective 8-bit S-boxes constructed using the unbalanced-Bridge is greater than 2 if and only if conditions i), ii), and iii) are all satisfied (λ_α and λ_β below represent arbitrary masks where $wt(\lambda_\alpha) = wt(\lambda_\beta) = 1$):*

i) For each $\lambda_\alpha, \lambda_\beta \in \mathbb{F}_2^3$, at least one of the entry $(\lambda_\alpha, \lambda_\beta)$ in LAT of S_3 and the entry $(0, \lambda_\beta||0^{(2)})$ in LAT of S_5^2 is 0,

ii) For each $\lambda_\alpha \in \mathbb{F}_2^5$ and $\lambda_\beta \in \mathbb{F}_2^3$, $\sum_{A\in\mathbb{F}_2^2} X \cdot Y = 0$ where X is the entry $(\lambda_\beta, \lambda_\alpha)$ in LAT of $\mathfrak{F}_A^1$ and Y is the entry $(\lambda_\beta, \lambda_\beta||0^{(2)})$ in LAT of $\mathfrak{F}_A^2$,

iii) For each $\lambda_\alpha, \lambda_\beta \in \mathbb{F}_2^5$ satisfying $\tau_3(\lambda_\beta) = 0$, $\sum_{A\in\mathbb{F}_2^2} X \cdot Y = 0$ where X is the entry $(0, \lambda_\alpha)$ in LAT of $\mathfrak{F}_A^1$ and Y is the entry $(0, \lambda_\beta)$ in LAT of $\mathfrak{F}_A^2$.



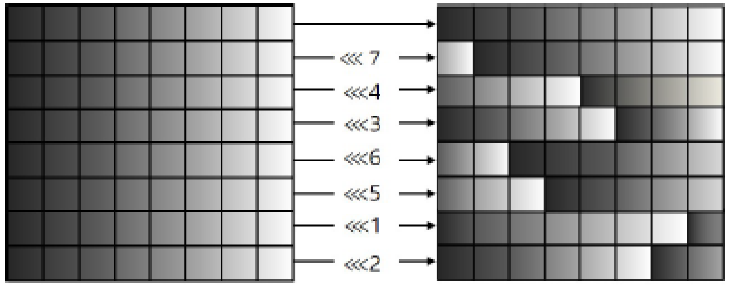
Unbalanced-Bridge structure

$$\tau_n : \mathbb{F}_2^5 \to \mathbb{F}_2^n,\ \ \tau_n(x||y) = x,\ \ \text{for } x \in \mathbb{F}_2^n,\ y \in \mathbb{F}_2^{5-n},\ n \in \{1,2,3,4\},$$
$$\mathfrak{F}_A^1 : \mathbb{F}_2^3 \to \mathbb{F}_2^5,\ \ \mathfrak{F}_A^1(X) = (S_5^1)^{-1}(X||A)\ \ \text{for } A \in \mathbb{F}_2^2,$$
$$\mathfrak{F}_A^2 : \mathbb{F}_2^3 \to \mathbb{F}_2^5,\ \ \mathfrak{F}_A^2(X) = S_5^2(X||A)\ \ \text{for } A \in \mathbb{F}_2^2.$$

Design Rationales of R-Layer

- Satisfying full diffusion in 2 rounds.
- Can be implemented using only 8-bit rotation operations.
- Combining the R-layer with the S-layer should enable the cipher to have the best resistance to DC and LC

```
//Input: (MSB) X[7], X[6], X[5], X[4], X[3], X[2], X[1], X[0] (LSB)
X[1] = ((X[1] << 7)) | ((X[1] >> 1));
X[2] = ((X[2] << 4)) | ((X[2] >> 4));
X[3] = ((X[3] << 3)) | ((X[3] >> 5));
X[4] = ((X[4] << 6)) | ((X[4] >> 2));
X[5] = ((X[5] << 5)) | ((X[5] >> 3));
X[6] = ((X[6] << 1)) | ((X[6] >> 7));
X[7] = ((X[7] << 2)) | ((X[7] >> 6));
//Output: (MSB) X[7], X[6], X[5], X[4], X[3], X[2], X[1], X[0] (LSB)
```



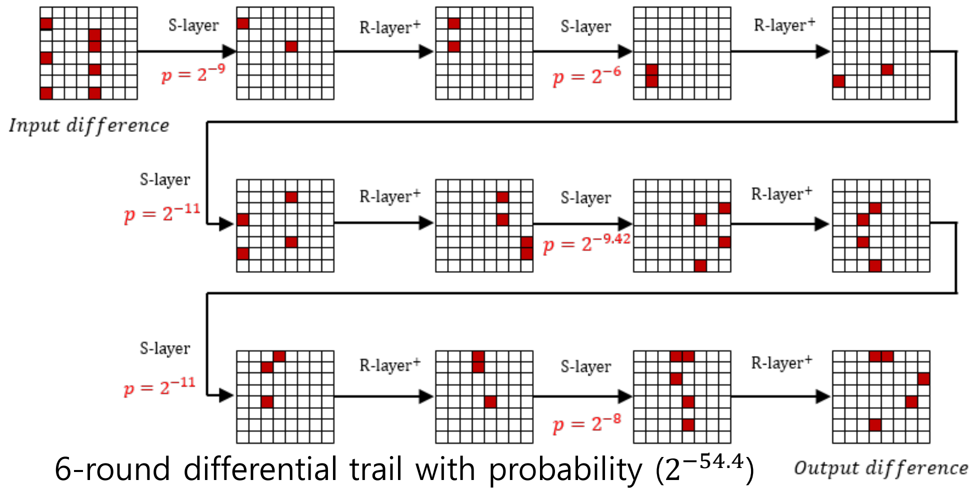
Full diffusion: any input bit can affect the entire output bits

Cryptographic Security (Differential cryptanalsysis)

- The best differential probability for 7-round PIPO is less then 2^{-64}.
  - Difference characteristic of 7 rounds or more cannot be used for differential attacks.
- The best of differential trails reaches 6 rounds with a probability of $2^{-54.4}$.
  - Up to 9-round key recovery attacks are possible using 6-round characteristics.

	Rounds						
	1	2	3	4	5	6	7
#(Active S-box)	1	2	4	6	9	11	13
Prob. of best trail	2^{-4}	2^{-8}	2^{-16}	$2^{-26.8}$	$2^{-40.4}$	$2^{-54.4}$	2^{-65}

Minimum numbers of differential active S-boxes and probabilities of best differential trails



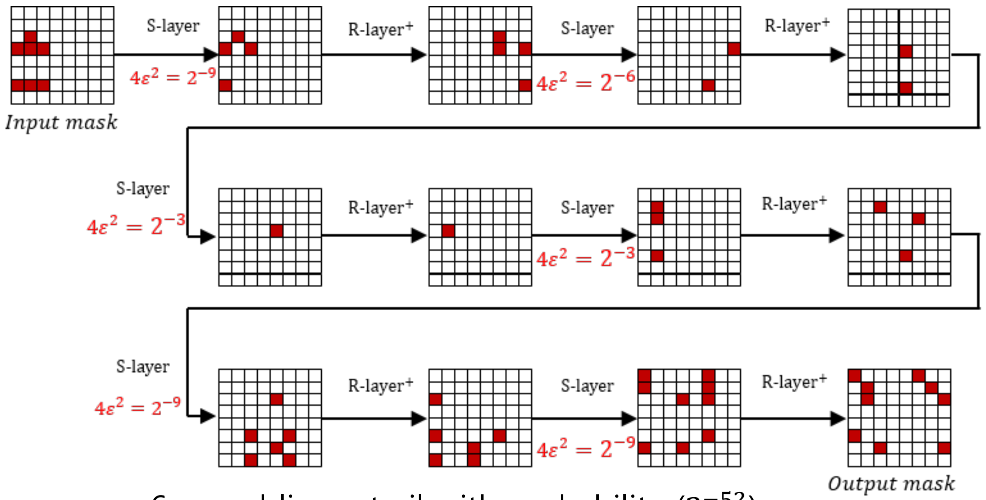
6-round differential trail with probability ($2^{-54.4}$)

Cryptographic Security (Linear cryptanalsysis)

- The best average correlation potentials of 7-round PIPO is less then 2^{-64}.
  - Linear characteristic of 7 rounds or more cannot be used for linear attacks.
- The best of linear trails reaches 6 rounds with a correlation potential of 2^{-52}.
  - Up to 9-round key recovery attacks are possible using 6-round Characteristics.

	Rounds						
	1	2	3	4	5	6	7
#(Active S-box)	1	2	4	6	9	11	13
Best correlation potential	2^{-4}	2^{-8}	2^{-16}	2^{-24}	2^{-38}	2^{-52}	2^{-66}

Minimum numbers of linear active S-boxes and correlation potentials of best linear trails



6-round linear trail with probability (2^{-52})

Cryptographic Security (Other cryptanalsysis)

- Boomerang/Rectangle Attack
- Impossible Differential Attack
- Algebraic Attack
- Integral Attack
- Statistical Saturation Attack
- Meet-In-The-Middle Attack
- Invariant Subspace Attack
- Nonlinear Invariant Attack
- Slide Attack
- Etc..

Table 2. The numbers of rounds of the best characteristics for each cryptanalysis

Key length	Cryptanalysis	Best characteristic	Key recovery attack
128-bit	Differential	6-round	9-round
	Linear	6-round	9-round
	Impossible differential	4-round	6-round
	Boomerang/Rectangle	6-round	8-round
	Meet-in-the-Middle	6-round	6-round
256-bit	Differential	6-round	11-round
	Linear	6-round	11-round
	Impossible differential	4-round	8-round
	Boomerang/Rectangle	6-round	10-round
	Meet-in-the-Middle	10-round	10-round

S/W Implementations

- $RANK = (10^6/CPB)/(ROM + 2 \times RAM)$
  - The metric to measure overall performance on low-end devices
  - Implementation environment: 8-bit AVR (ATmega128 running at 8MHz)

Block cipher	Code size (bytes)	RAM (bytes)	Execution time (cycles per byte)	RANK
PIPO-64/128	320	31	197	13.31
SIMON-64/128	290	24	253	11.69
RoadRunneR-64/128	196	24	477	8.59
RECTANGLE-64/128	466	204	403	2.84
PRIDE-64/128	650	47	969	1.39
SKINNY-64/128	502	187	877	1.30
PRESENT-64/128	660	280	1,349	0.61
CRAFT-64/128	894	243	1,504	0.48
PIPO-64/256	320	47	224	10.77

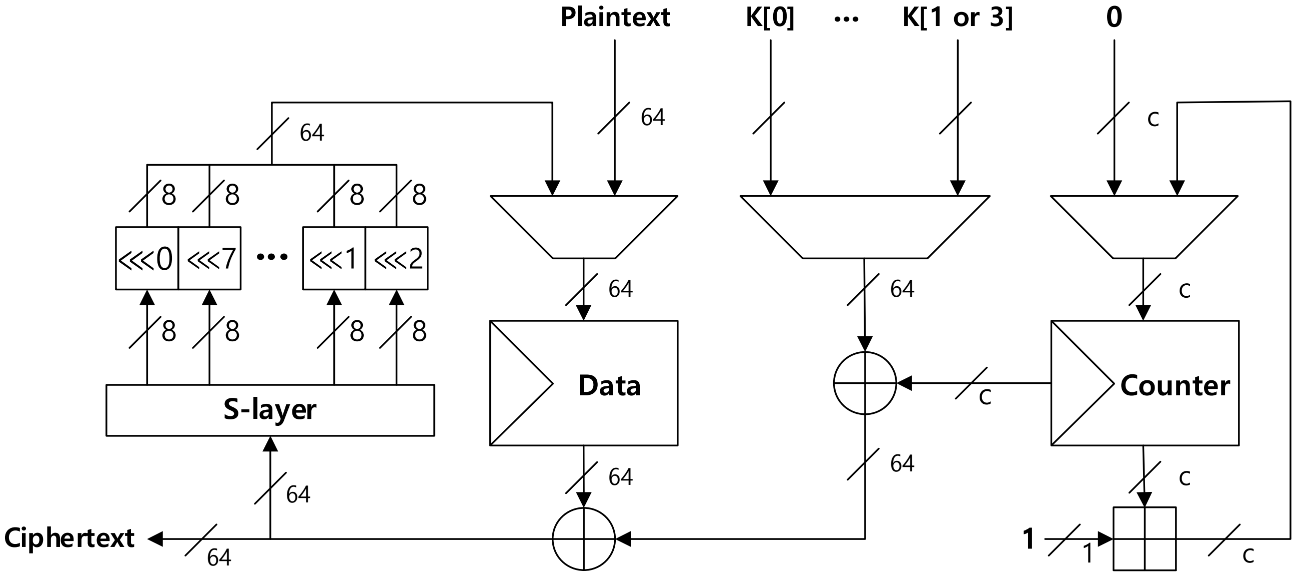
Comparison of software implementation performances
with block ciphers optimized for Bitslice implementation

H/W Implementations

- $FOM = (bits \times 10^9)/(clk + GE^2)$
  - nano bits per clock cycle per GE squared
  - Implementation environment: 130nm ASIC library

Block cipher	Area [GE]	Throughput (Kbps@100KHz)	cycles /block	FOM $[\frac{bits \times 10^9}{clk \times GE^2}]$
PIPO-64/128	1,446	492	13	2,355
CRAFT-64/128	949	200	32	2,221
Piccolo-64/128	1,197	194	33	1,354
SIMON-64/128	1,417	133	48	664
RECTANGLE-64/128	2,064	246	26	578
PIPO-64/256	1,583	427	15	1,703

Comparison of hardware implementation performances



Area optimized hardware implementation

Higher-Order Masking Implementations

- PIPO implementation does not require table or constant storage
- PIPO can be implemented with the smallest nonlinear operations among block ciphers that can be implemented in bitslice

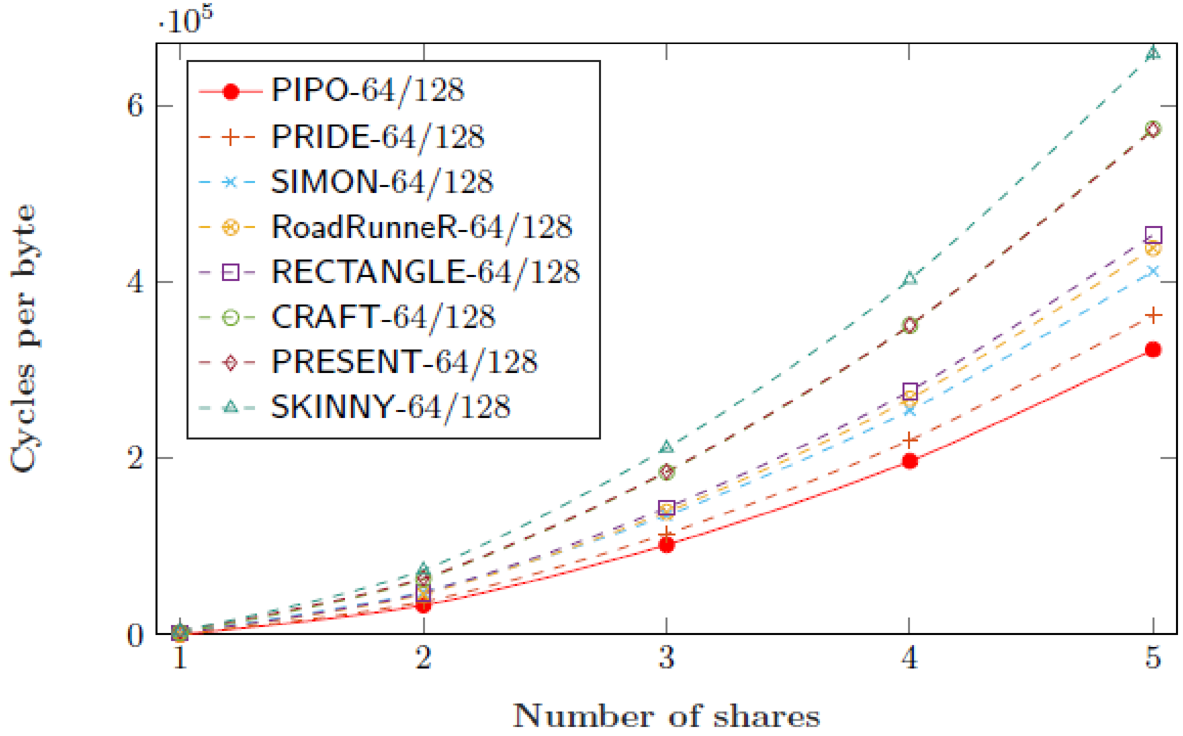
Block cipher	Table size	#(nonlinear bitwise operations)	Permutation
PIPO-64/128	0	1,144	7 bit-rotations in bytes
PRIDE-64/128	80	1,280	MixColumns*
SIMON-64/128	62	1,408	3 bit-rotations in 32-bit words
RoadRunneR-64/128	0	1,536	24 bit-rotations in bytes
RECTANGLE-64/128	25	1,600	3 bit-rotations in 16-bit words
CRAFT-64/128	64	1,984	MixColumns*, PermuteNibbles
PRESENT-64/128	0	1,984	Bit permutation
SKINNY-64/128	62	2,304	ShiftRows, MixColumns*

* : multiply with binary matrix

Comparison of the number of nonlinear operations with block ciphers optimized for bitsliced implementation

Higher-Order Masking Implementations

- As the number of shares increases, the gap of cycles per bytes according to the number of nonlinear operations becomes prominent.



Conclusion

- New lightweight block cipher PIPO.
- Optimized for 8-bit microcontrollers and hardware implementations.
- Excellent performance in both side-channel protected (Plug-In) and unprotected environments (Plug-Out)

Test vectors, reference codes can be found in github (https://github.com/PIPO-Blockcipher)