Lab 6: Simon Cipher

Secure Communication

- Encryption is the process of encoding a message, *P*, with a key, *K*, to obtain an undecipherable output *C*.

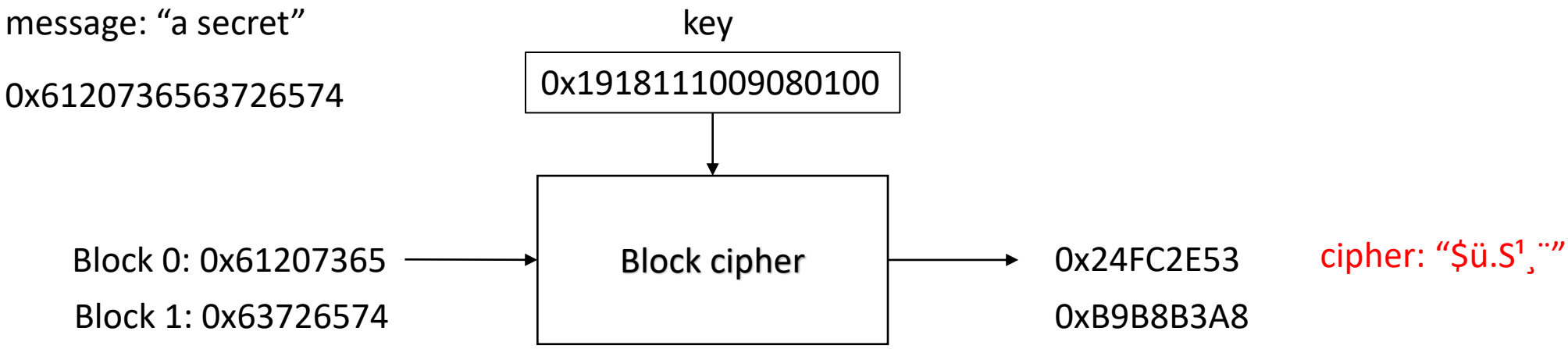
$$E(P, K) \rightarrow C$$

- The reverse of the process is decryption

$$E^{-1}(C, K) \rightarrow P$$

Block Cipher

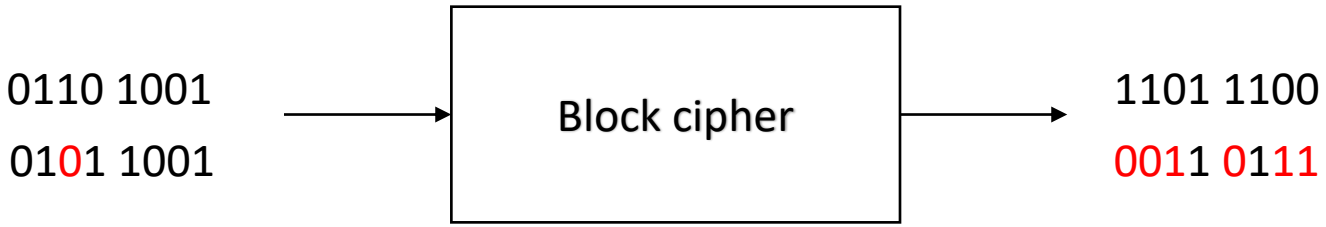
- Algorithm for encrypting blocks of data at a time



Reference: Understanding Cryptography by Christof Paar and Jan Pelzl

More Terms

- confusion – relationship between plaintext and ciphertext is obscured [1].
- diffusion – influence of change in each bit of plaintext over the change of bits in the ciphertext [1].



- round – a series of specific operations applied on a block of data.
  - rounds can be chained to introduce more confusion and diffusion which makes the ciphertext harder to break.

[1] Reference: Understanding Cryptography by Christof Paar and Jan Pelzl

Simon Cipher

- Light weight block cipher developed by National Security Agency in 2013.
- Designed to be efficiently implemented in hardware.
- Several Configurations:

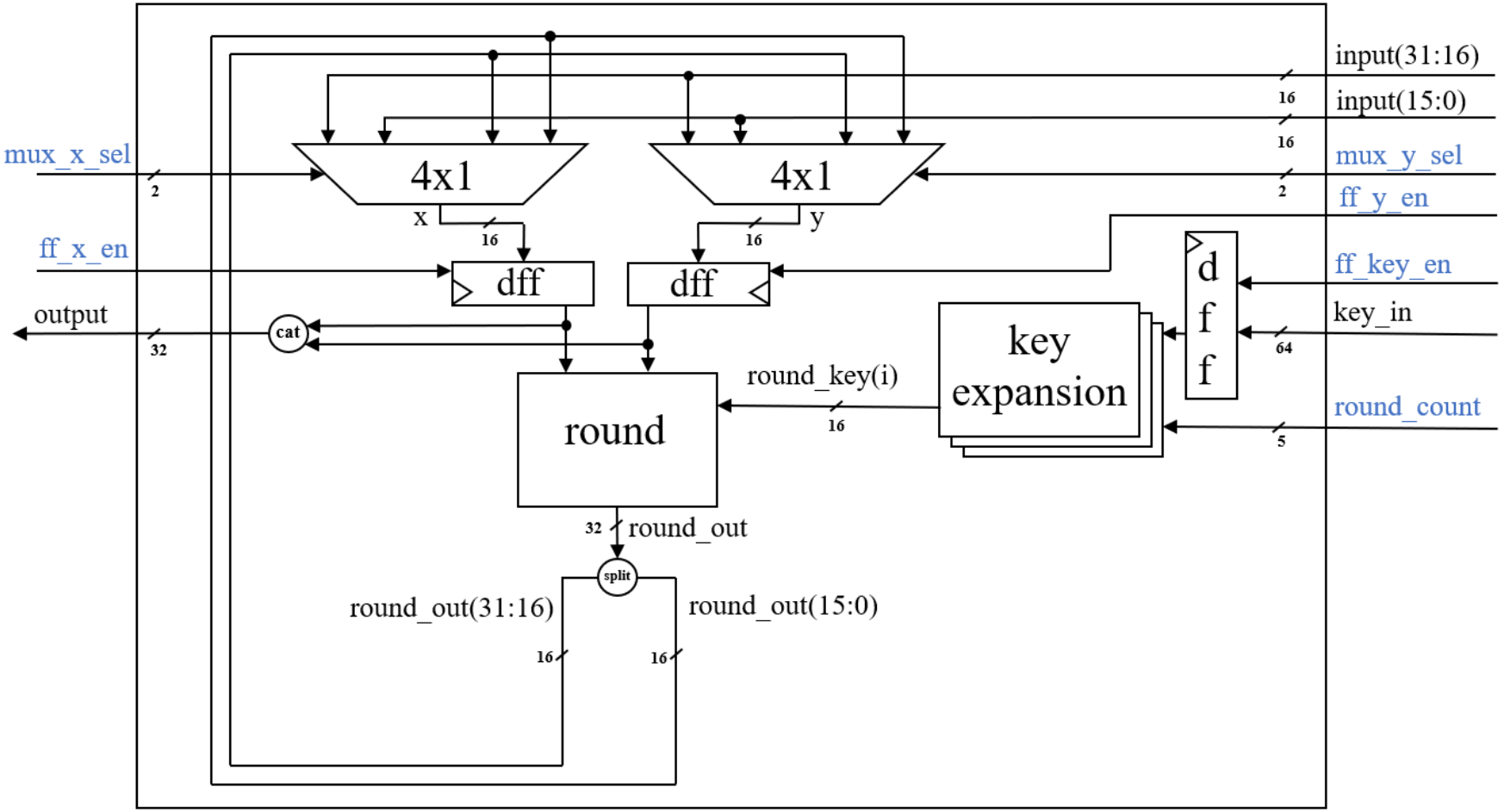
Block Size	Key Size
32	64
48	72, 96
64	96,128
96	96,144
128	128,192,256

Reference: R. Beaulieu, et al., “The SIMON and SPECK Families of Lightweight Block Ciphers.” https://eprint.iacr.org/2013/404

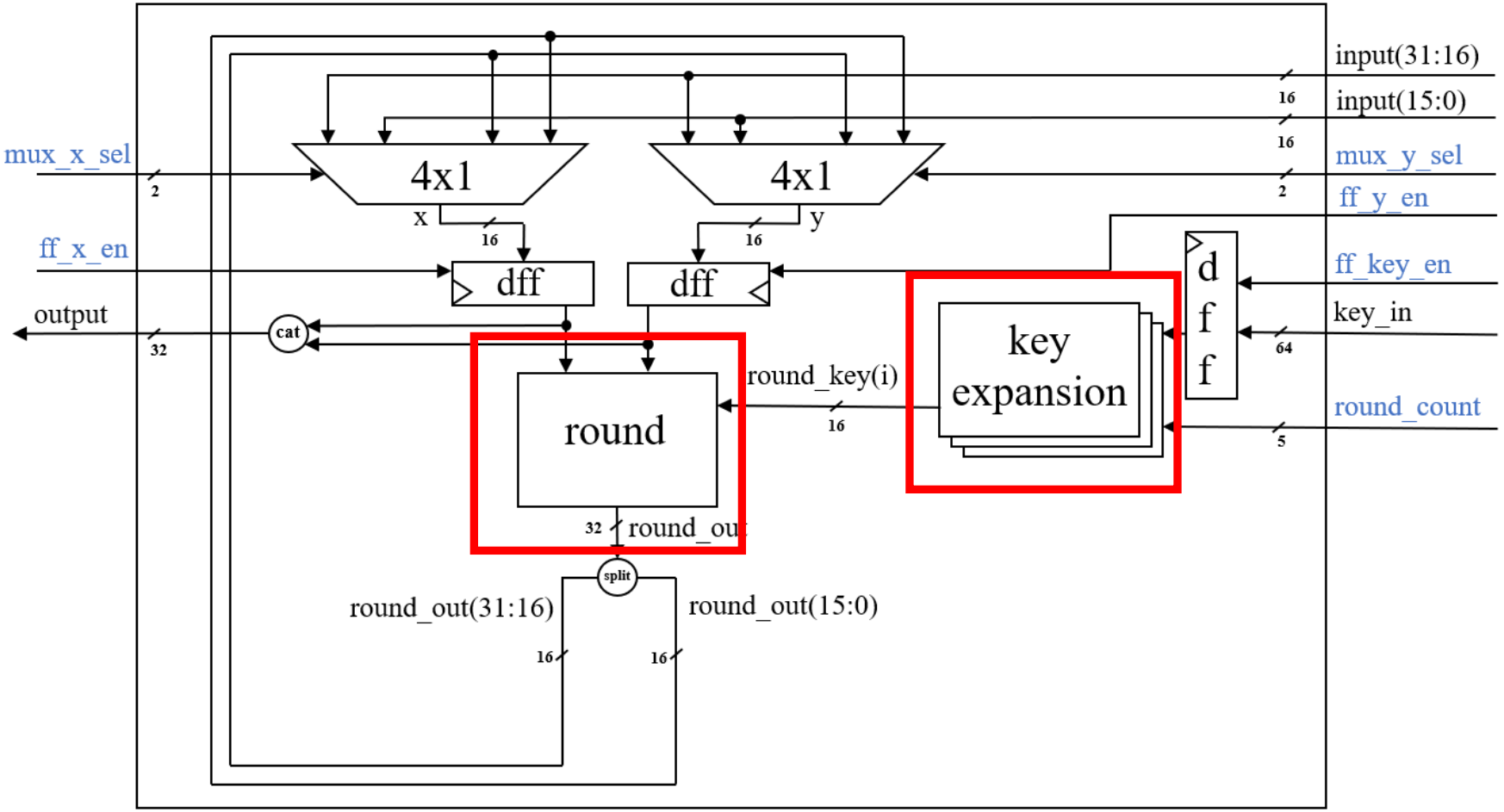
Lab 6 – Simon Cipher (lite)

- Simon32/64
  - Encryption only
  - 10 rounds instead of 32
  - Block size – 32 bits
  - Key size – 64 bits
  - Word size – 16 bits

Datapath


input(31:16)
input(15:0)
mux_y_sel
ff_y_en
ff_key_en
key_in
round_count
mux_x_sel
ff_x_en
output
4x1
4x1
x
y
dff
dff
cat
key expansion
d f f
round_key(i)
round
round_out
split
round_out(31:16)
round_out(15:0)
16
2
32
64
5

Datapath


input(31:16)
input(15:0)
mux_y_sel
ff_y_en
ff_key_en
key_in
round_count
mux_x_sel
ff_x_en
output
4x1
4x1
x
y
16
16
2
2
dff
dff
cat
32
round_key(i)
16
key expansion
d
f
f
64
5
round
32
round_out
split
round_out(31:16)
round_out(15:0)
16
16

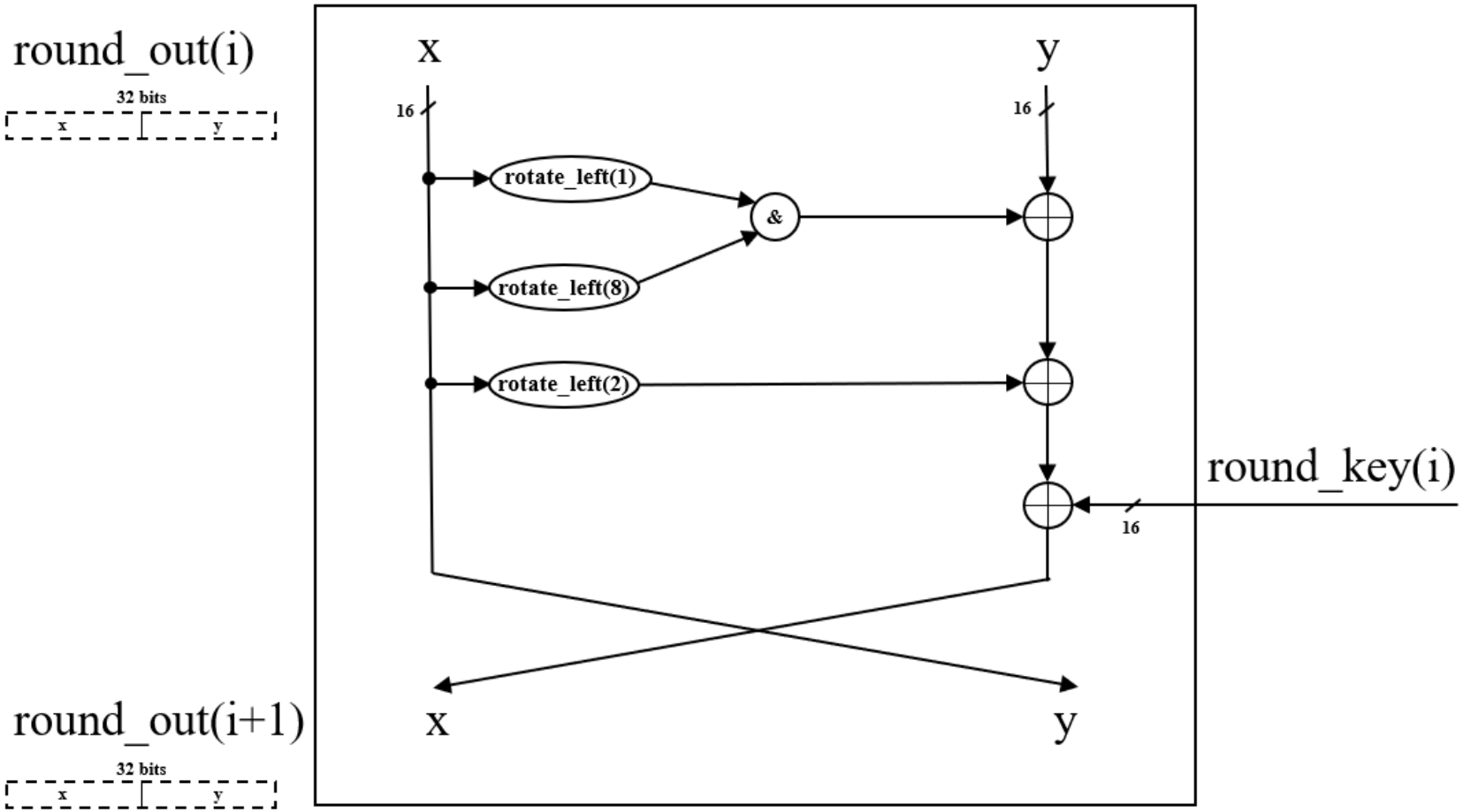
Round

```
Input: x, y
tmp = x

 x = y xor
     (circular_shift_left(x,1) and circular_shift_left(x,8)) xor
     circular_shift_left(x,2) xor
     round_key[i]

 y = tmp
```

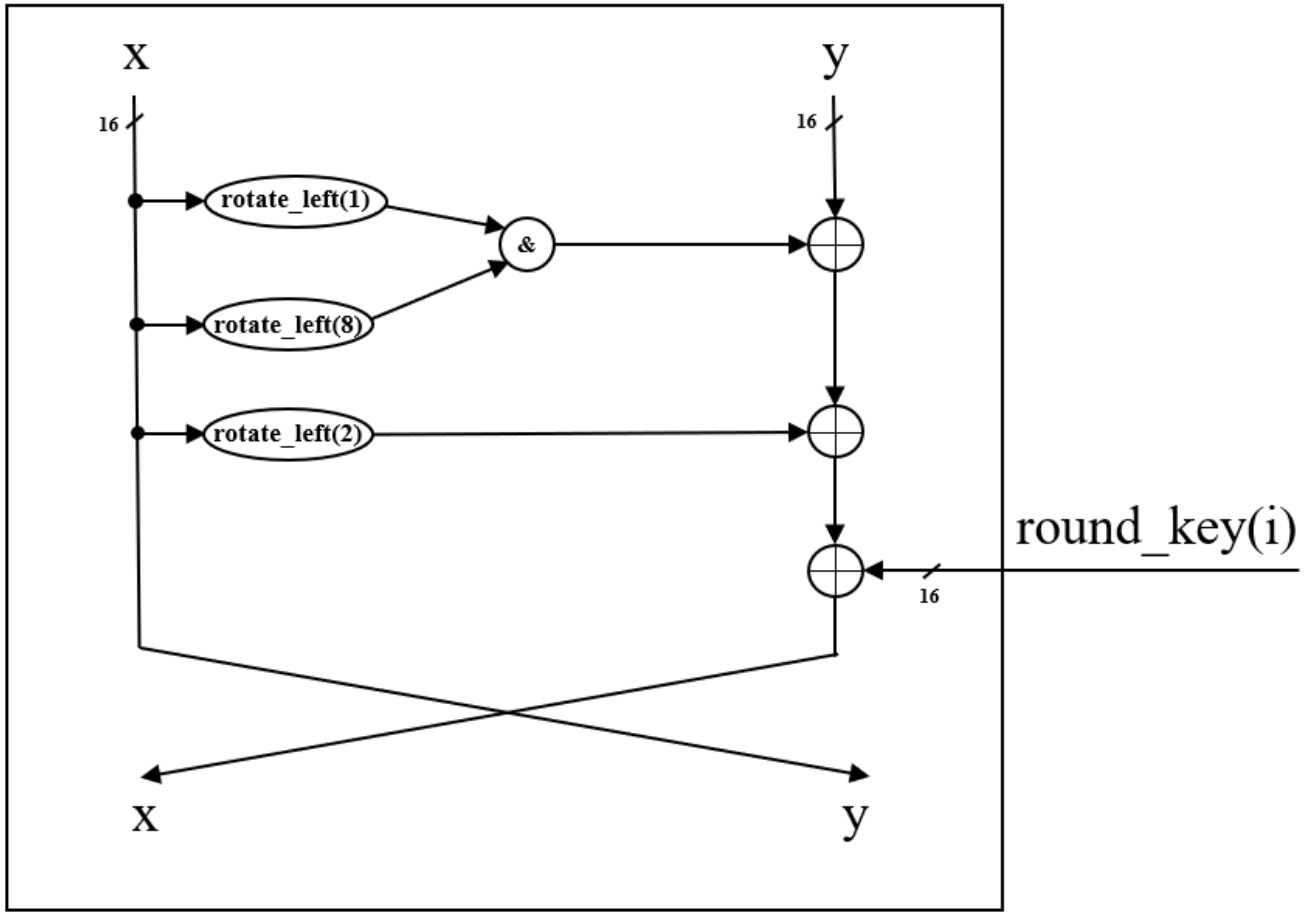
Round


round_out(i)
32 bits
x
y
x
y
16
16
rotate_left(1)
&
rotate_left(8)
rotate_left(2)
round_key(i)
16
round_out(i+1)
32 bits
x
y
x
y

Round (0)

“test”
0x74657374

key
0x1918111009080100



Round (0)

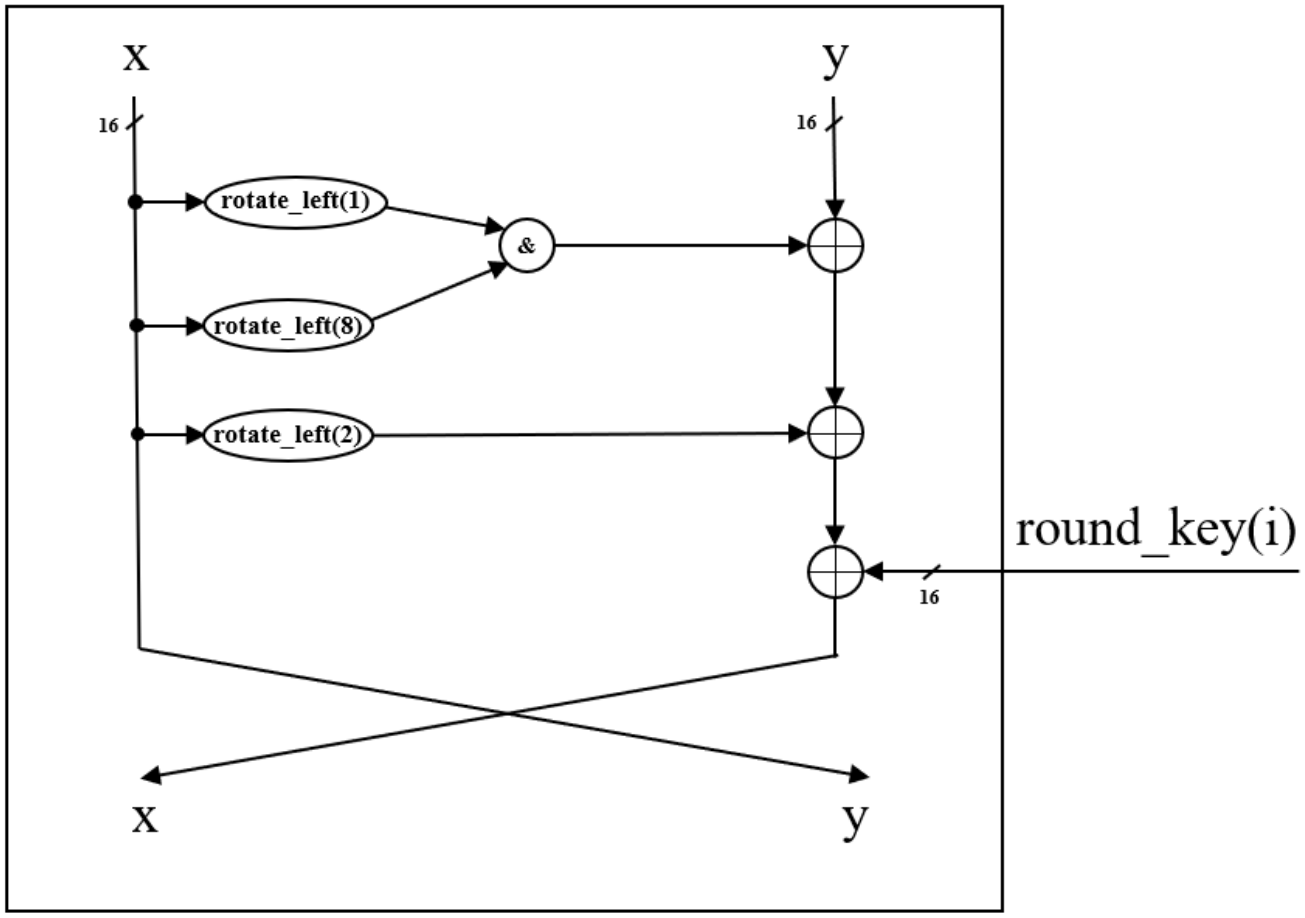
"test"

key

0x1918111009080100

0x7465

0x7374

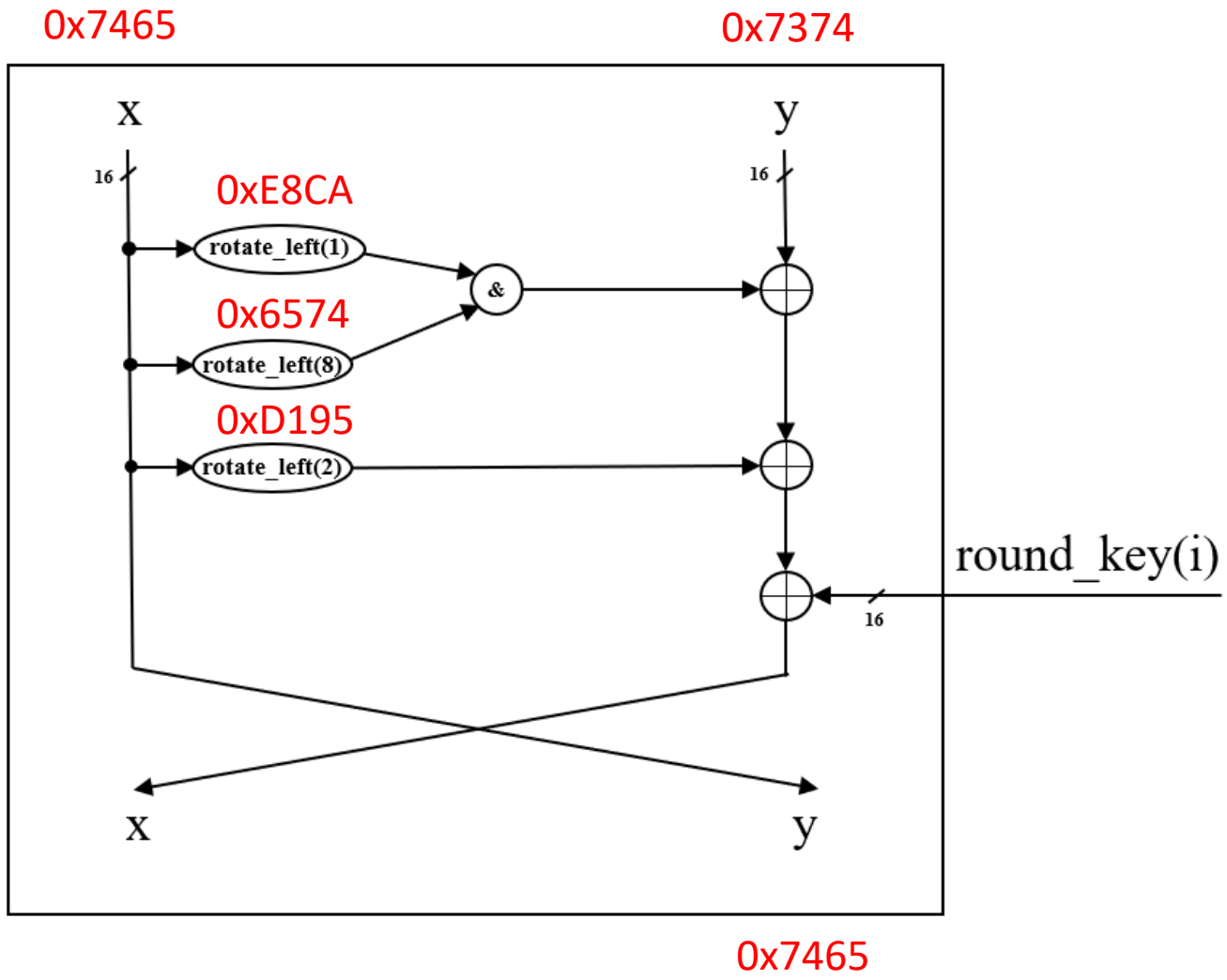


Round (0)

"test"

key

0x1918111009080100

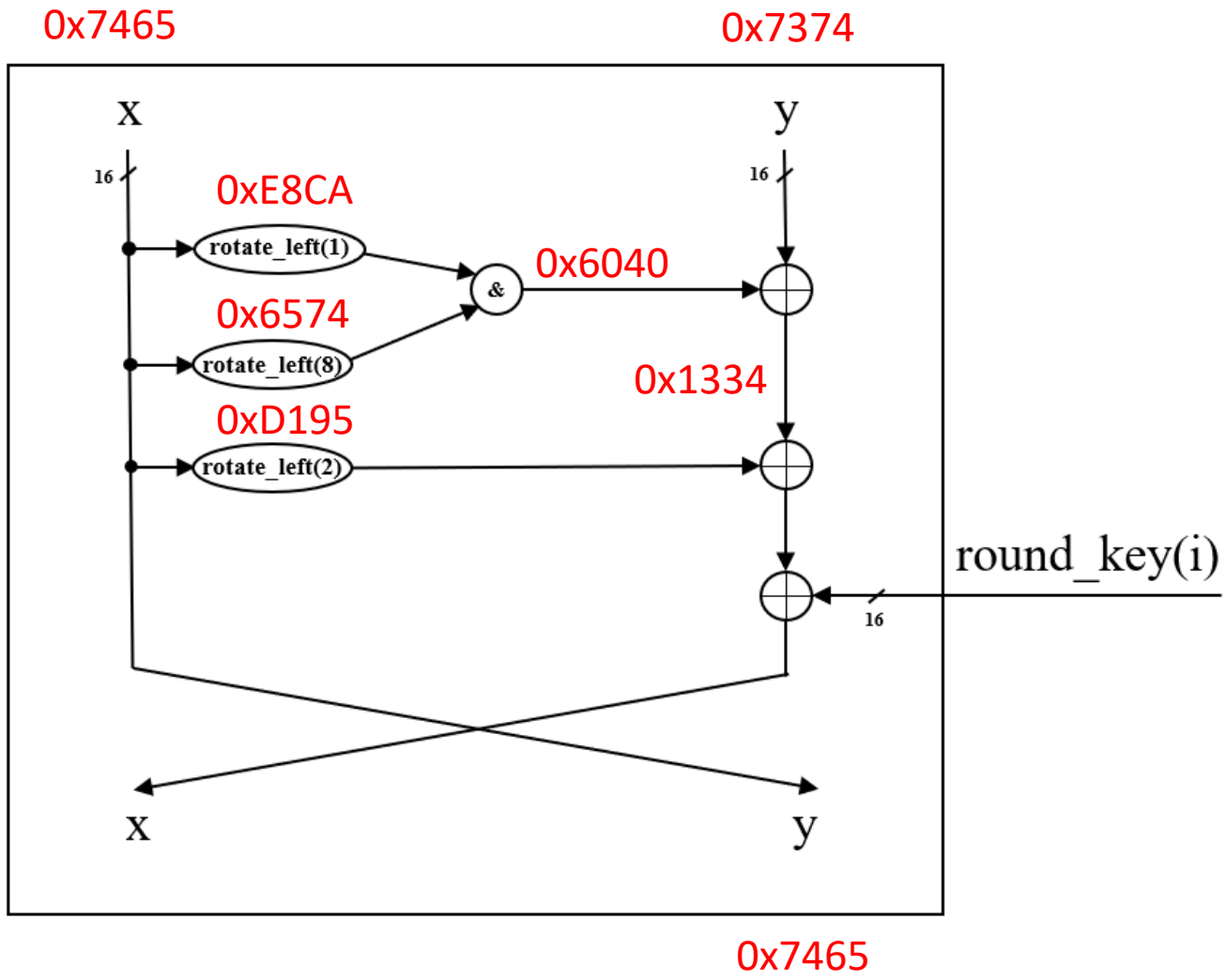


Round (0)

"test"

key

0x1918111009080100

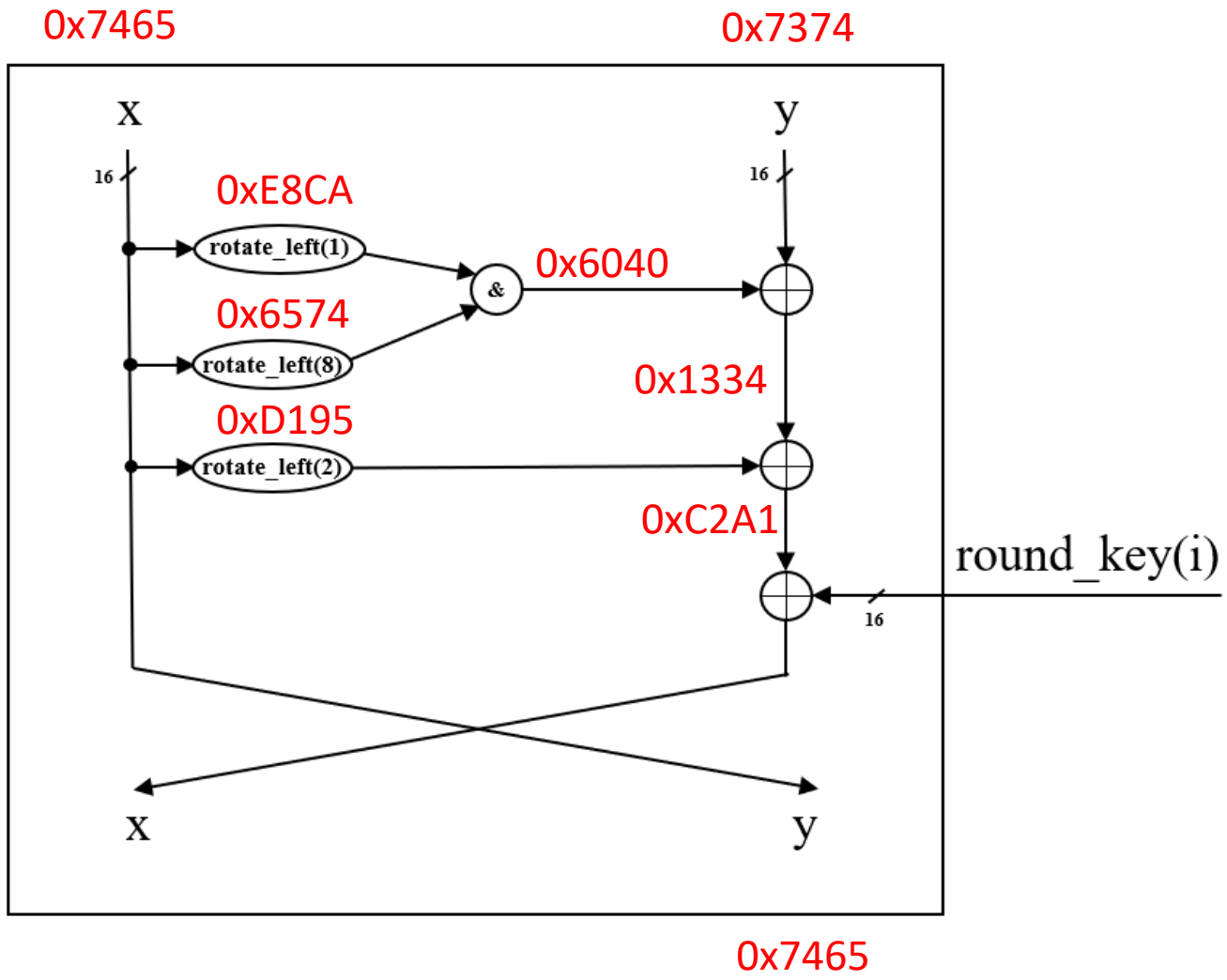


Round (0)

"test"

key

0x1918111009080100

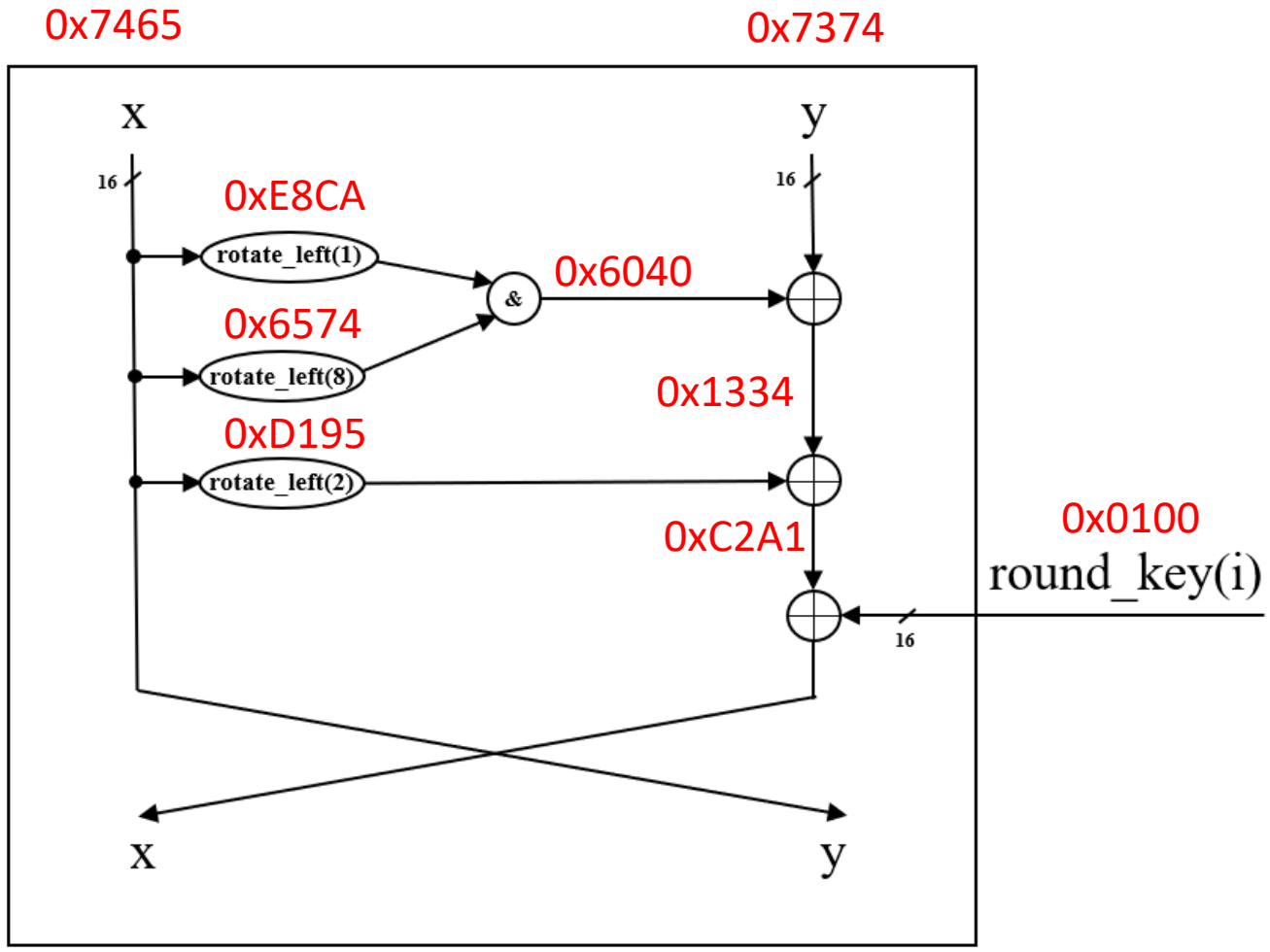


Round (0)

"test"

key

0x19181110090800100



Round_out(0)

0xC3A1

0x7465

Round (1)

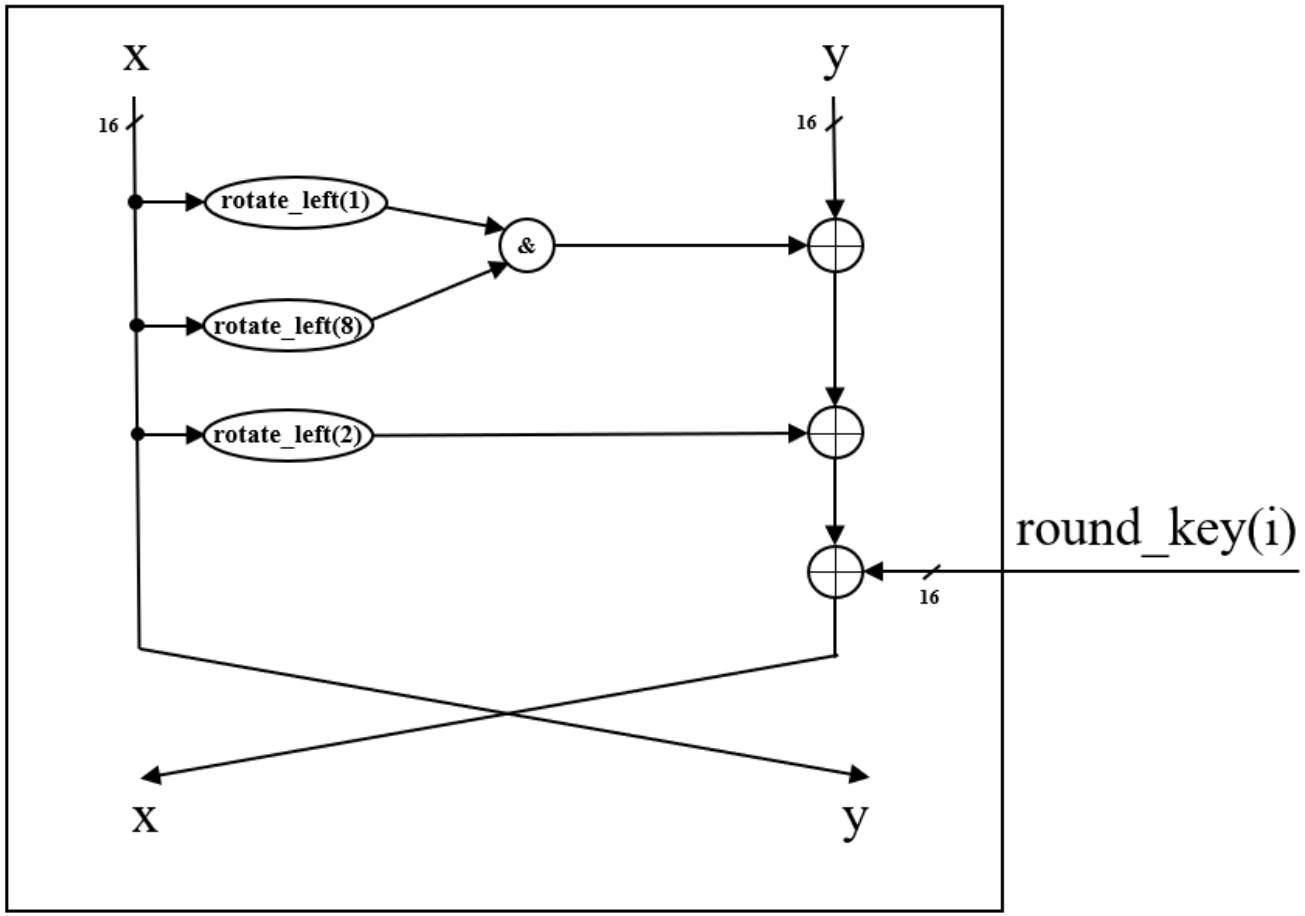
"test"

key

0x19181110090800100

0xC3A1

0x7465



Key expansion

- Each round needs a unique key.

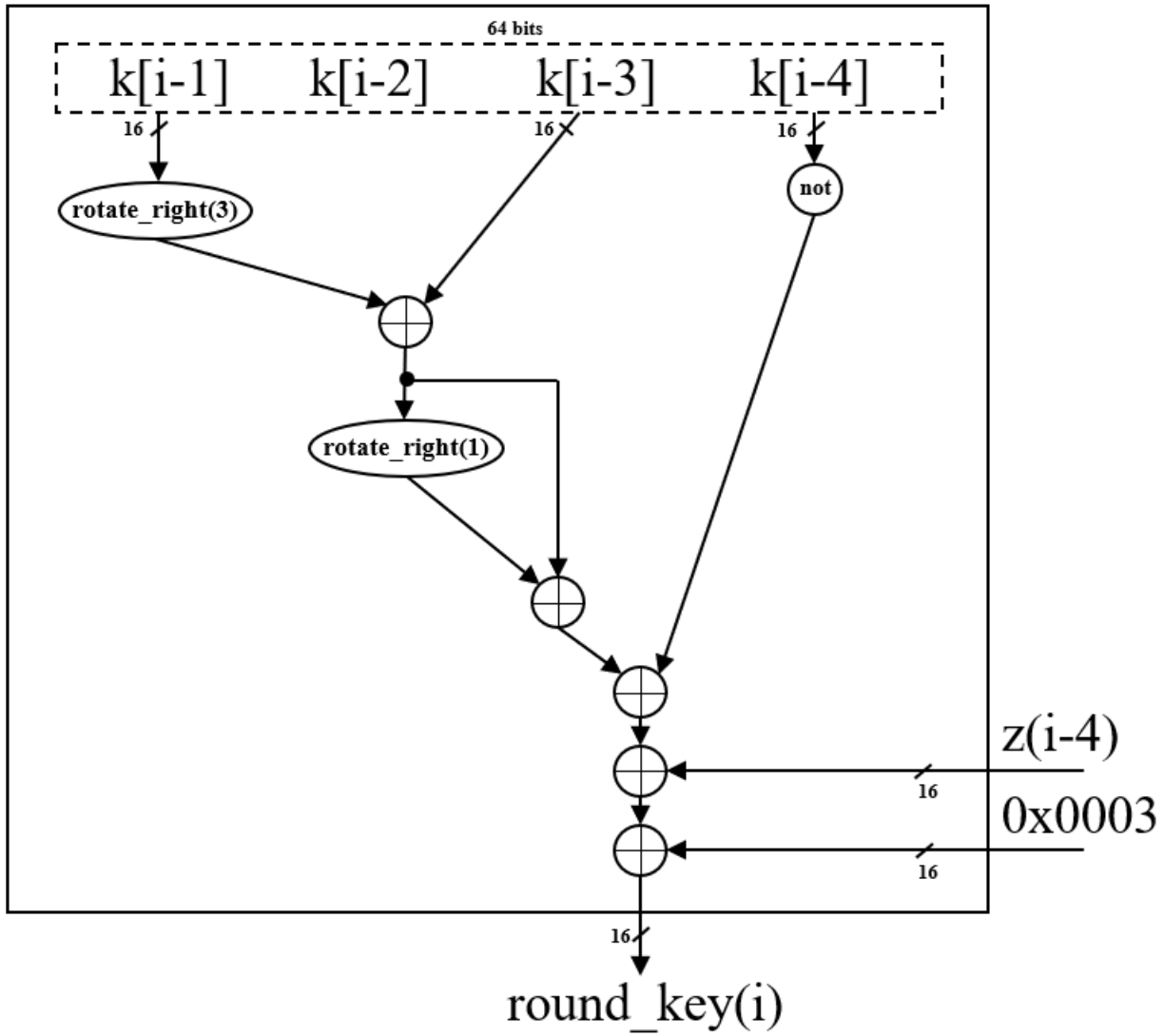
0x1918111009080100

- Round key is word size in width (16 bits)
- Round 0 – 0x0100
- Round 1 – 0x0908
- Round 2 – 0x1110
- Round 3 – 0x1918
- Round 4? Round N+4?

Key expansion

```
round_key[9]
for i = 4…9 {
    tmp = circular_shift_right(round_key[i-1], 3)
    tmp = tmp xor round_key[i-3]
    tmp = tmp xor circular_shift_right(tmp, 1)
    round_key[i] = ~(round_key[i-4]) xor tmp xor
                   z[i-4 mod 62] xor 3
}
```

Key expansion



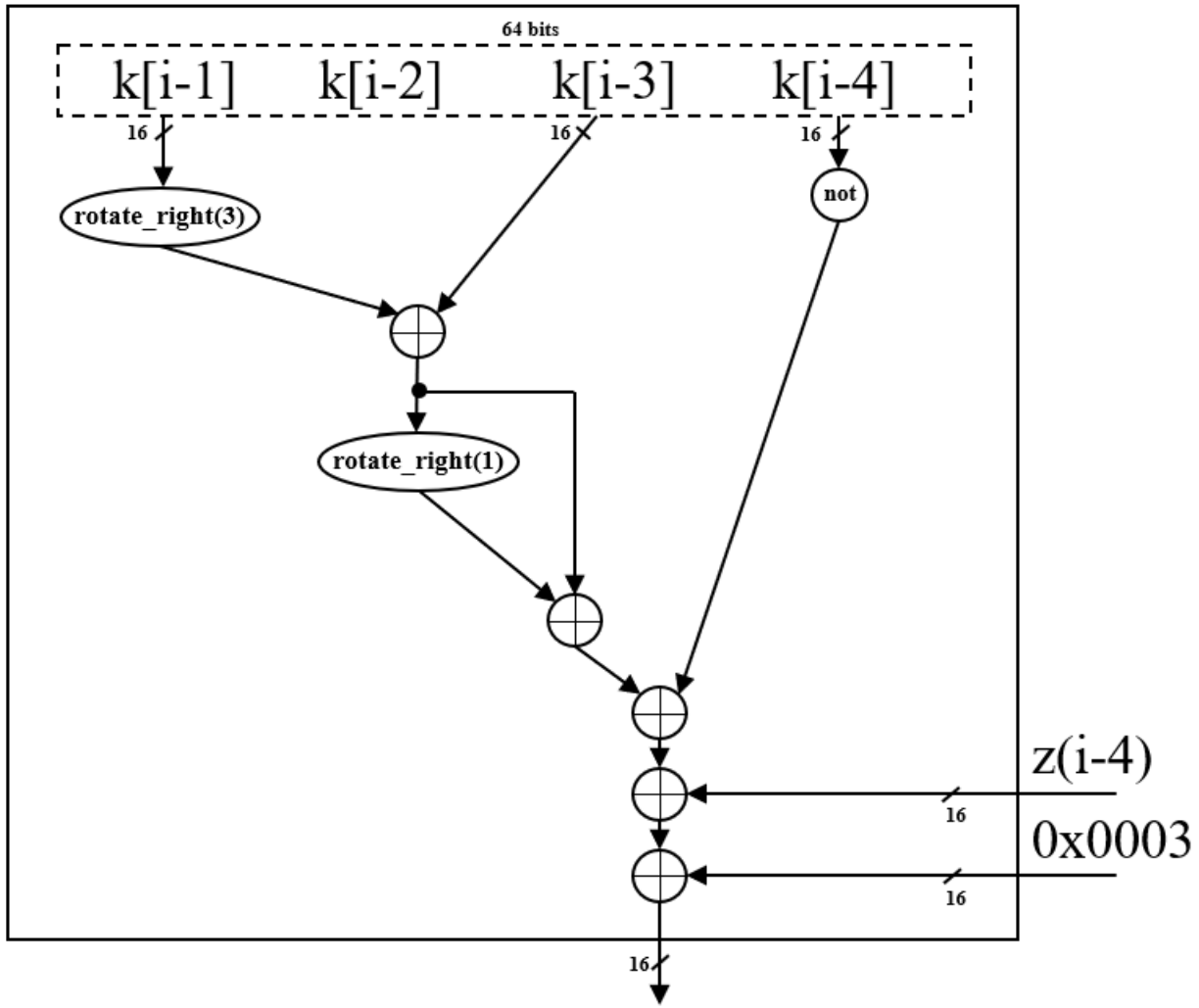
Round key (4)

key

0x1918111009080100

z

01100111000011010100100010000101111110110011100001101010010001101011111



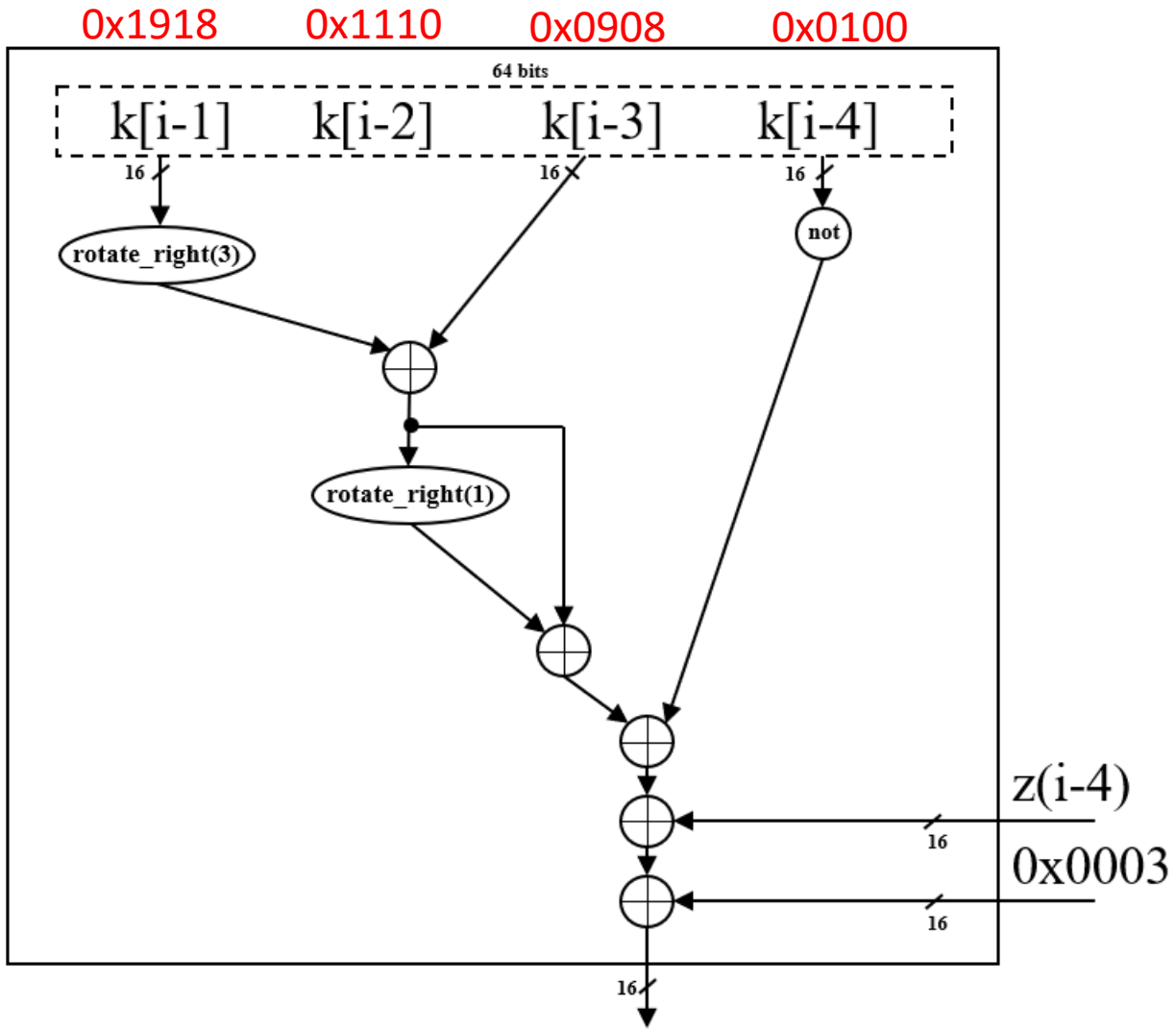
Round key (4)

key

0x1918111009080100

z

0110011100001101010010001011111011000011010100100001011111



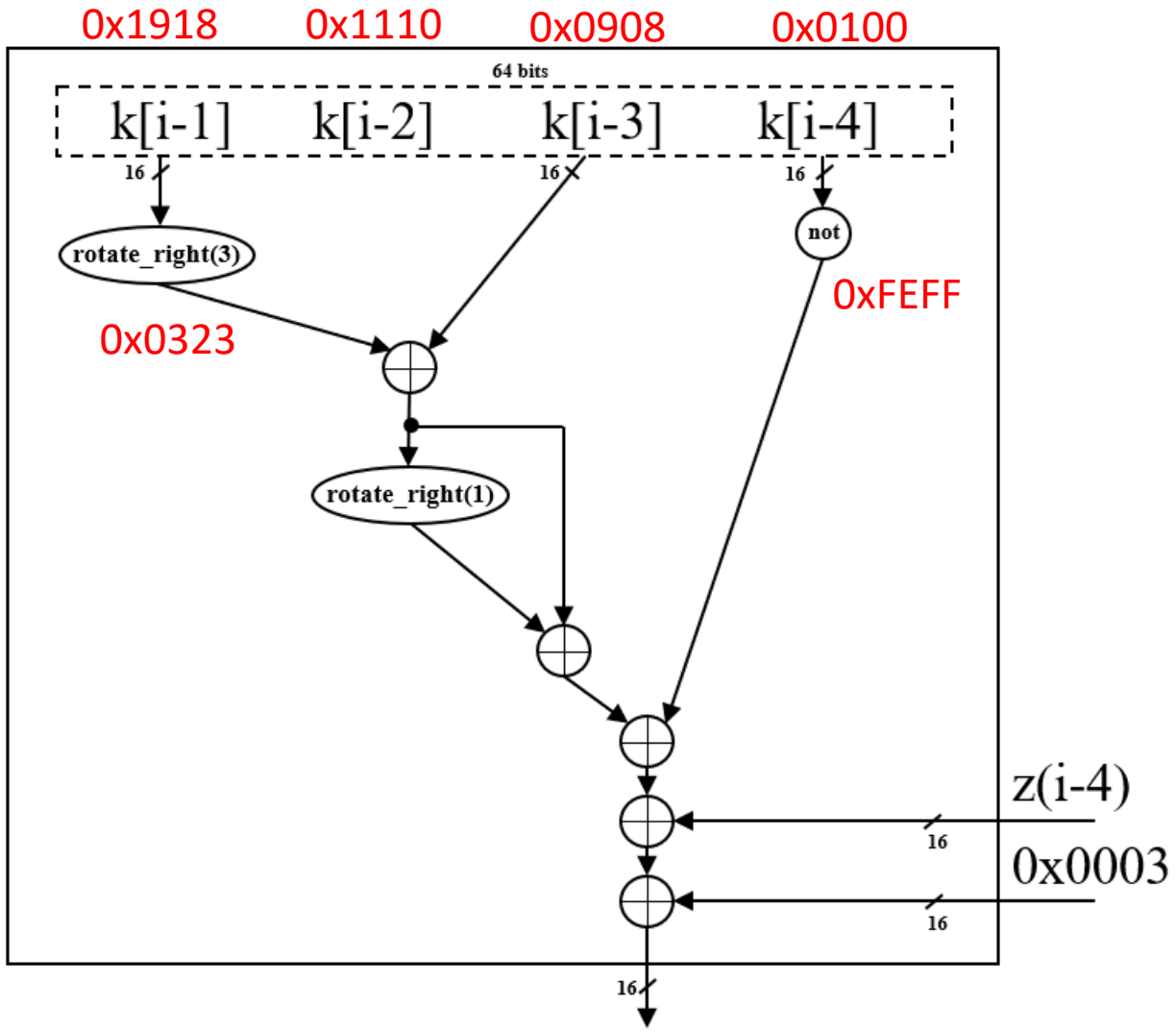
Round key (4)

key

0x1918111009080100

z

0110011100001101010010001011111011000011010100100001011111



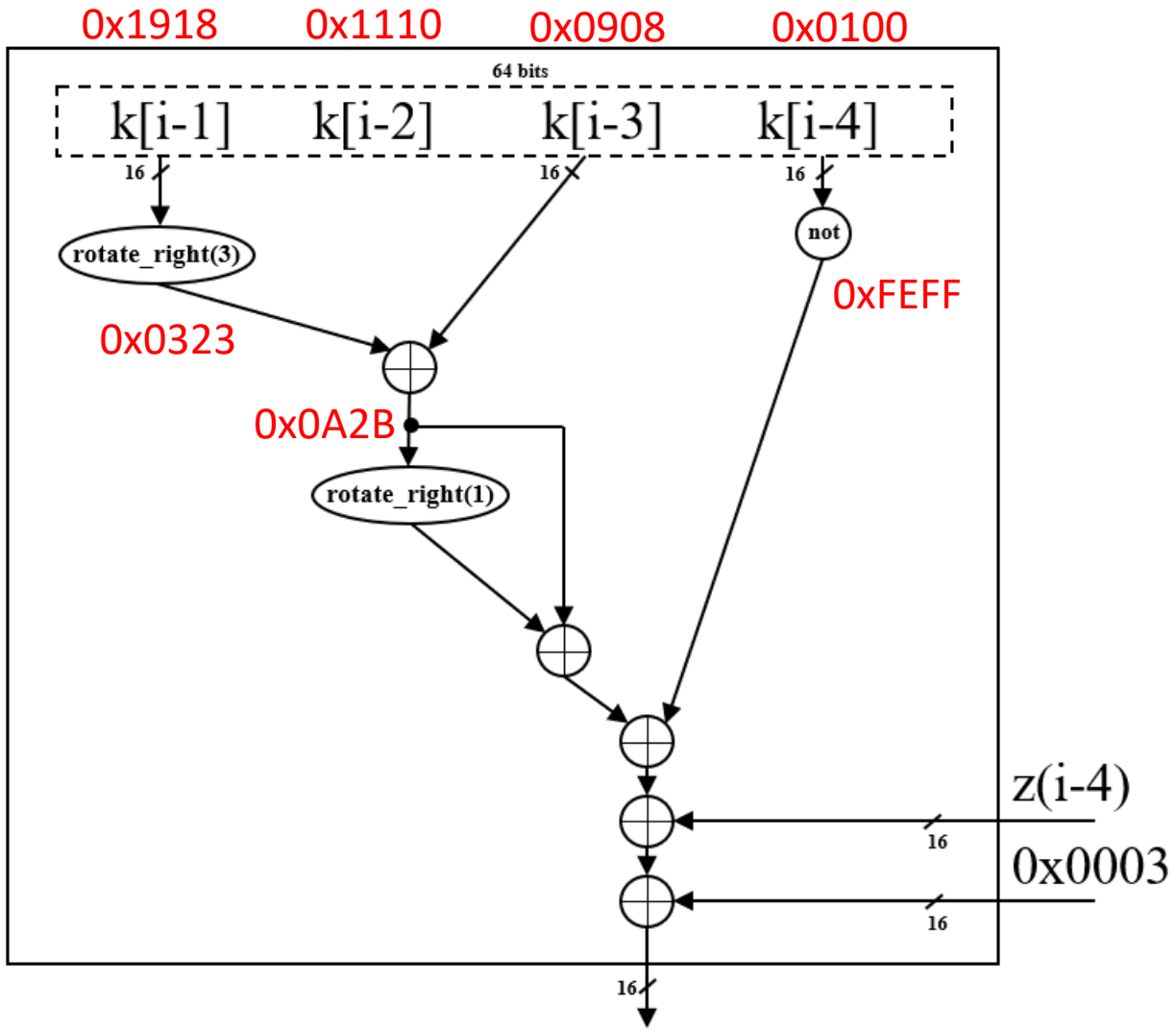
Round key (4)

key

0x1918111009080100

z

0110011100001101010010001011111011000011010100100001011111



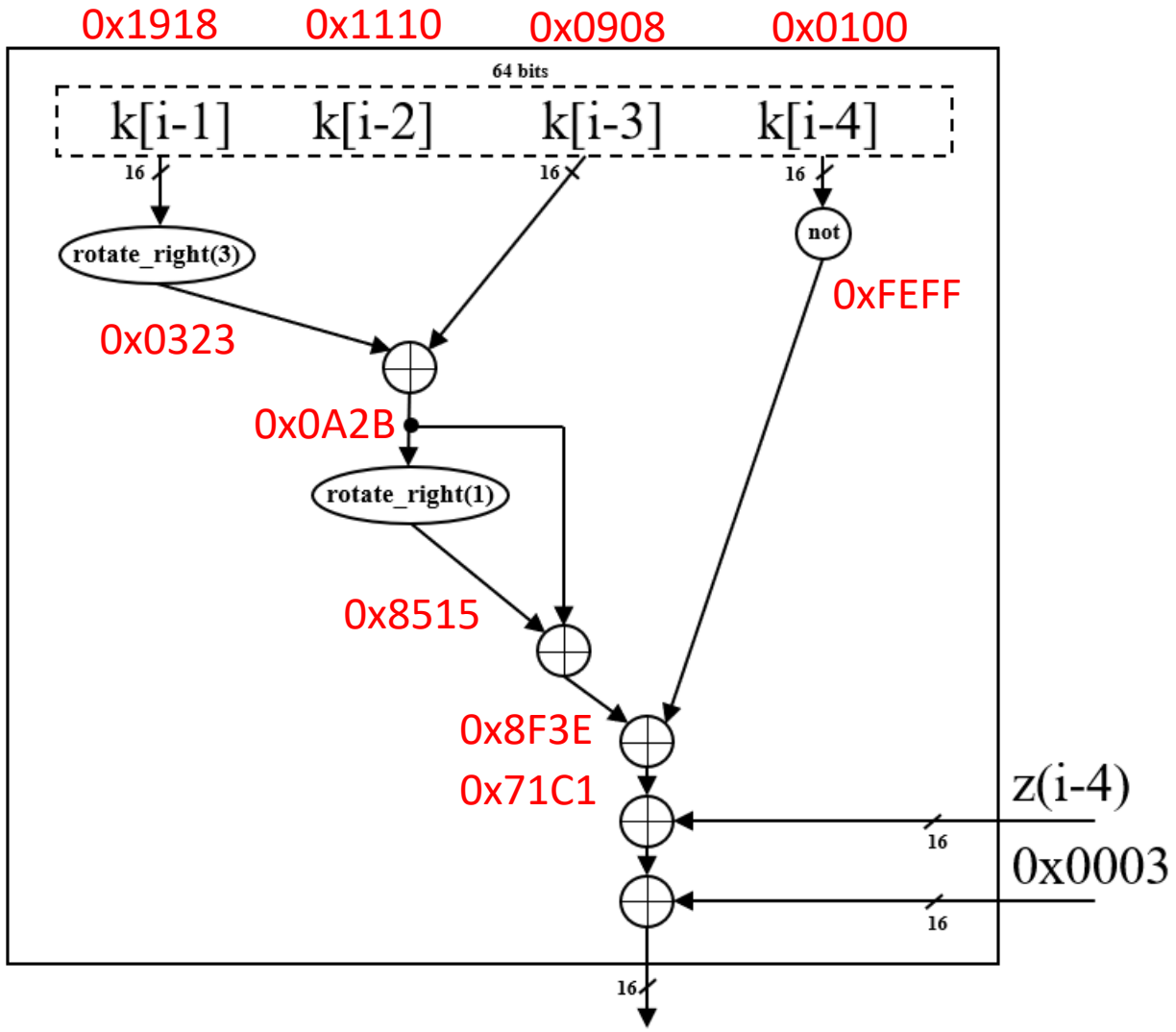
Round key (4)

key

0x1918111009080100

z

0110011100001101010010001011111011000011010100100001011111



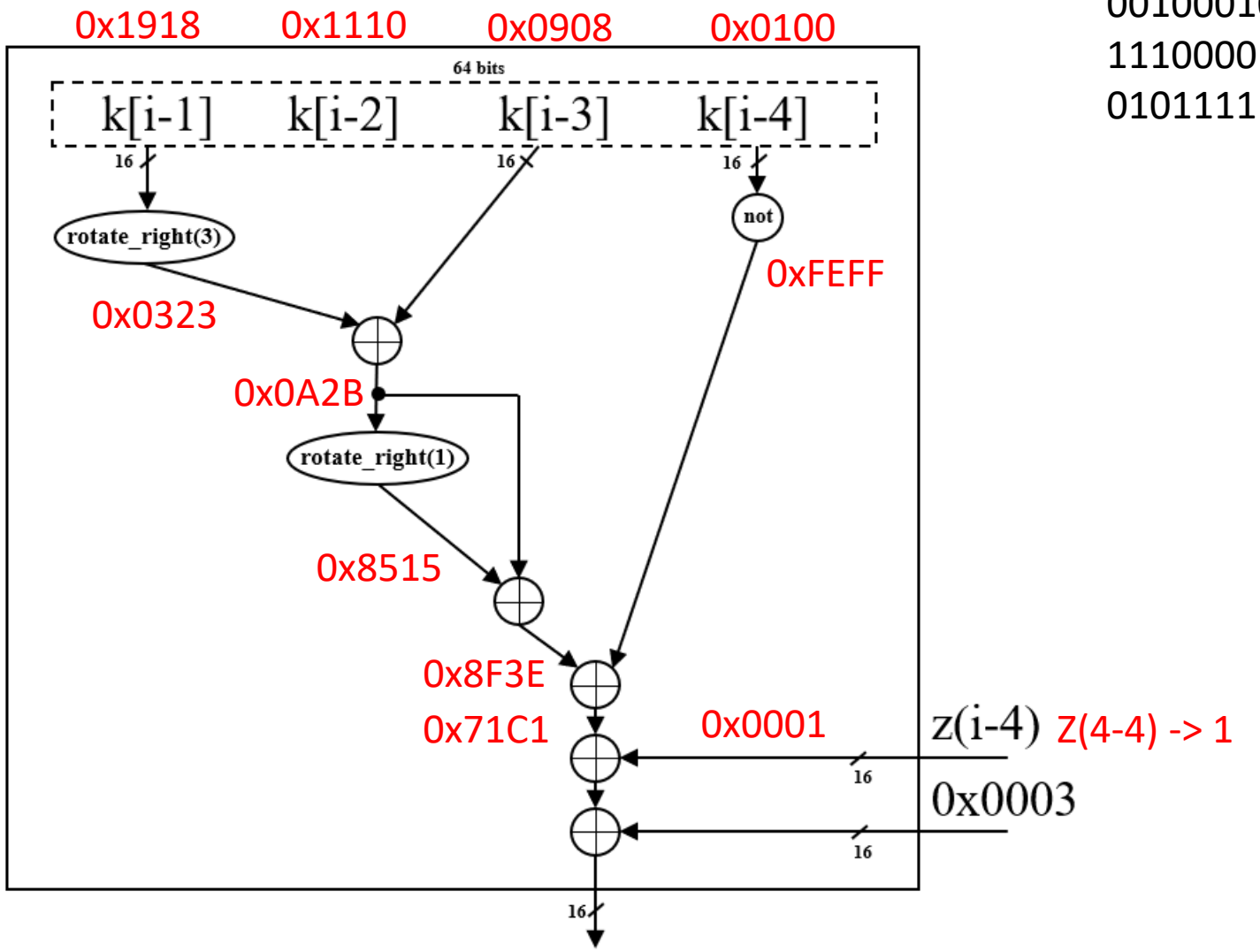
Round key (4)

key

0x1918111009080100

z

0110011100001101010010001011111011001110000110100100101011111



Round key (4)

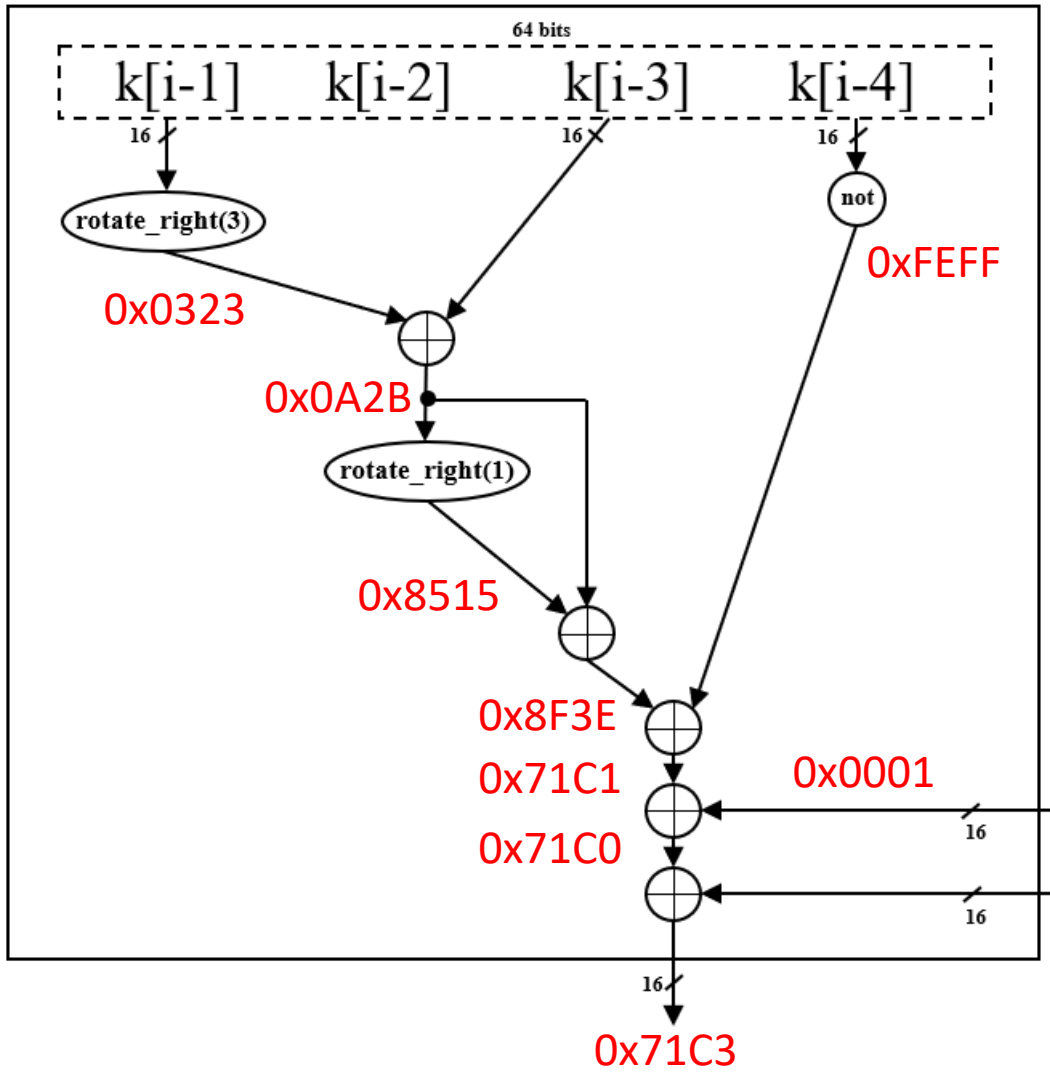
key

0x1918111009080100

z

0110011100001101010010001011111011001110000110100100101011111

0x1918 0x1110 0x0908 0x0100

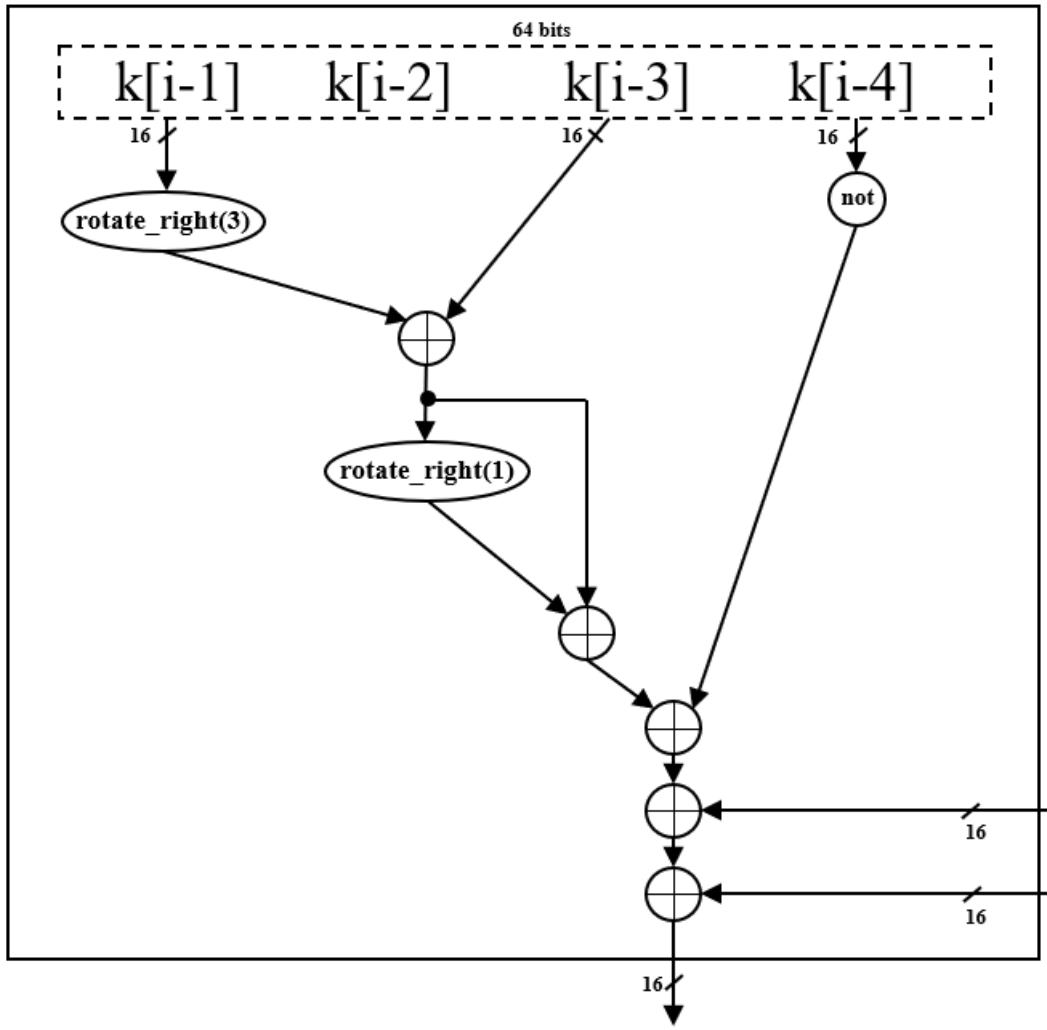


Round key (5)

key

0x1918111009080100

0x71C3 0x1918 0x1110 0x0908



Z(5-4) -> 1

z

0110011100001101010010001011111011001110000011010100100101011111

Simon Top Level