Implementation of SIMON 2n/mn

Project of

Security of Computer and Embedded systems

Lecturer: Prof. Dr. Elif Bilge Kavun

Exchange student: Farzaneh Arzaghi

Winter sem 20/21

Implementation of SIMON 128/128

According to the table below we can determine the cipher block's parameters.

block size $2n$	key size mn	word size n	key words m	const seq	rounds T
32	**64**	16	4	z_0	32
48	**72**	24	3	z_0	36
	96		4	z_1	36
64	**96**	32	3	z_2	42
	128		4	z_3	44
96	**96**	48	2	z_2	52
	144		3	z_3	54
128	**128**	64	2	z_2	68
	192		3	z_3	69
	256		4	z_4	72

Table 3.1: SIMON parameters.

Specify parameters and computing Z_2:

block size $(2n)$ = 128

Key size (mn) = 128

Word size (n) = 64

key words (m) = 2

Constant sequence (z_j) = Z_2

Rounds (T) = 68

Z_2 formed by computing the bitwise XOR of the period 2 sequence $t = t_0 t_1 t_2 \ldots = 0101\ldots$ with u

$t = t_0 t_1 t_2 \ldots = 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 01\ 0 \ldots$

$u = u_0 u_1 u_2 \ldots = 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 11\ 0000\ 111\ 00\ 11\ 0 \ldots$

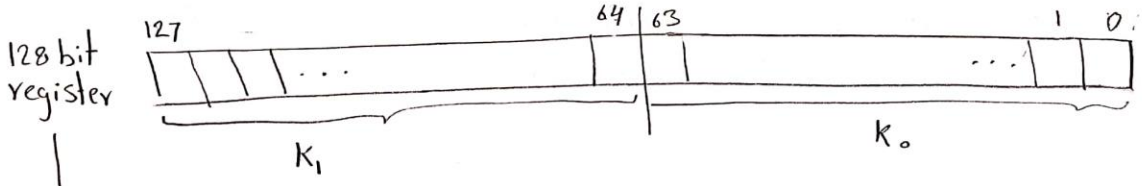
$Z_2 = (Z_2)_0 (Z_2)_1 (Z_2)_2 \ldots = 1010111101110000001101001001100 \ldots$

Key Schedules:

because $m = 2$ we have (k_0, k_1) → master key

The SIMON key schedule takes master key and generates a sequence of T key words $(k_0, k_1, k_2 \dots k_{T-1})$

Key words k_0 and k_1 are used as the first and second round keys, they are loaded into the shift registers with k_0 on the right and k_1 on the left.



this is the master key = input

other round keys generated by:

C is a constant → $C = 2^n - 4$ $\xrightarrow{\text{in this question}}$ $2^{64} - 4$

$$k_{i+m} \begin{cases} c \oplus (z_j)_i \oplus k_i \oplus (I \oplus s^{-1}) s^{-3} k_{i+1} & \text{if } m = 2 \end{cases}$$

- for $0 \leq i < T - m \xrightarrow{m=2} 0 \leq i < 66 \longrightarrow$ $i = 0 \; k_2$, $i = 1 \; k_3$, … $i = 65 \; k_{67}$ → $\{k_0, k_1, k_2, \dots k_{67}\}$

  ↓

  we need 68 different keys per round but because we already had 2 keys (k_0, k_1), we just generate 66 keys by this formula

- $(z_j)_i$ is the i^{th} bit of z_j $\xrightarrow{\text{according to SIMON 128/128}}$ $z_j = z_2$ → we have calculated

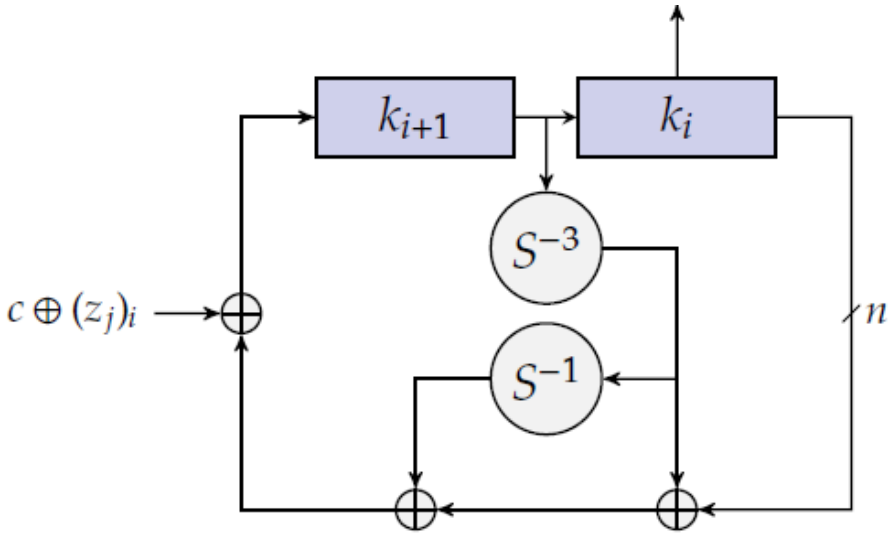
- left circular shift, s^j, by j bits
- right circular shift, s^{-j}, by j bits

For example:

For generating k_2 : $i = 0$ $\quad C \oplus (z_2)_0 \oplus k_0 \oplus (I \oplus s^{-1})\, s^{-3} k_1$

$$\bar{C} = \underbrace{C \oplus (z_2)_0}_{=1\ (0^{th}\text{ bit of } z_2)} \oplus k_0 \oplus s^{-3} k_1 \oplus s^{-1}(s^{-3} k_1)$$

k_1 is circular shifted right by three (s^{-3}), and the result is circular shifted right by one (s^{-1}) also the result XORed with the word k_0. Finally, the result is XORed with the round constant ($C \oplus (z_2)_0$).



Round Function:

SIMON 2n encryption make use of following operations on n-bit words:

- bitwise XOR ($\oplus$)
- bitwise AND ($\&$)
- left circular shift, S^j, by j bits.

SIMON 2n round function is the two stage Feistel map defined by:

$$R_k(x,y) = (y \oplus f(x) \oplus k,\ x)$$

- $f(x) = (Sx \,\&\, S^8x) \oplus S^2x$
- k is a round key
- x is the leftmost word of the cipher block
- y is the rightmost word

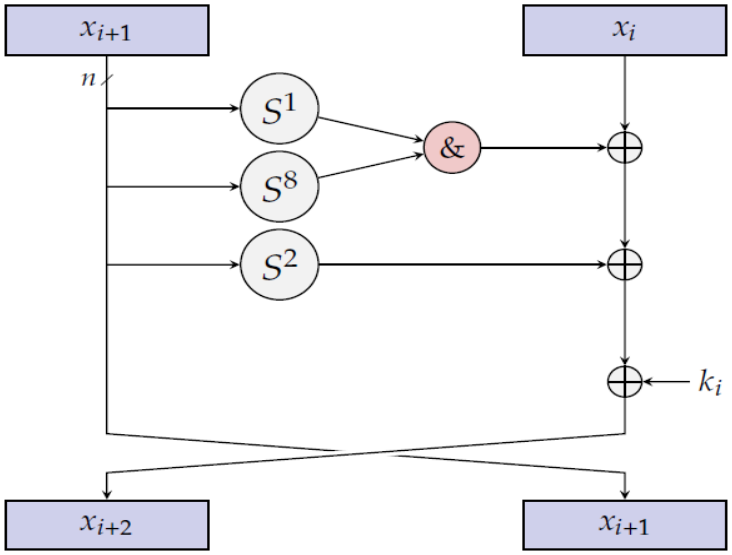
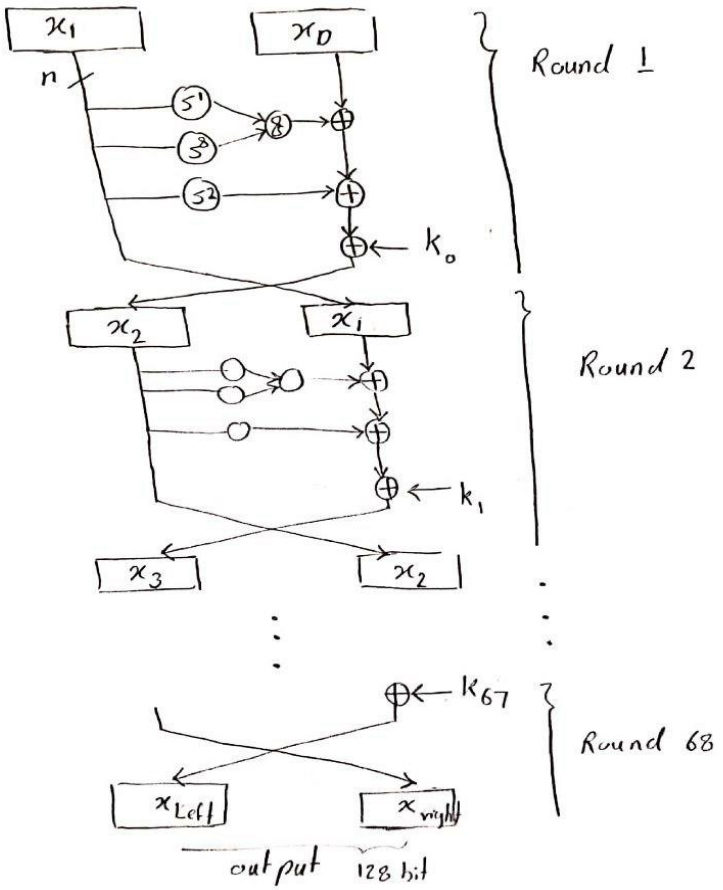


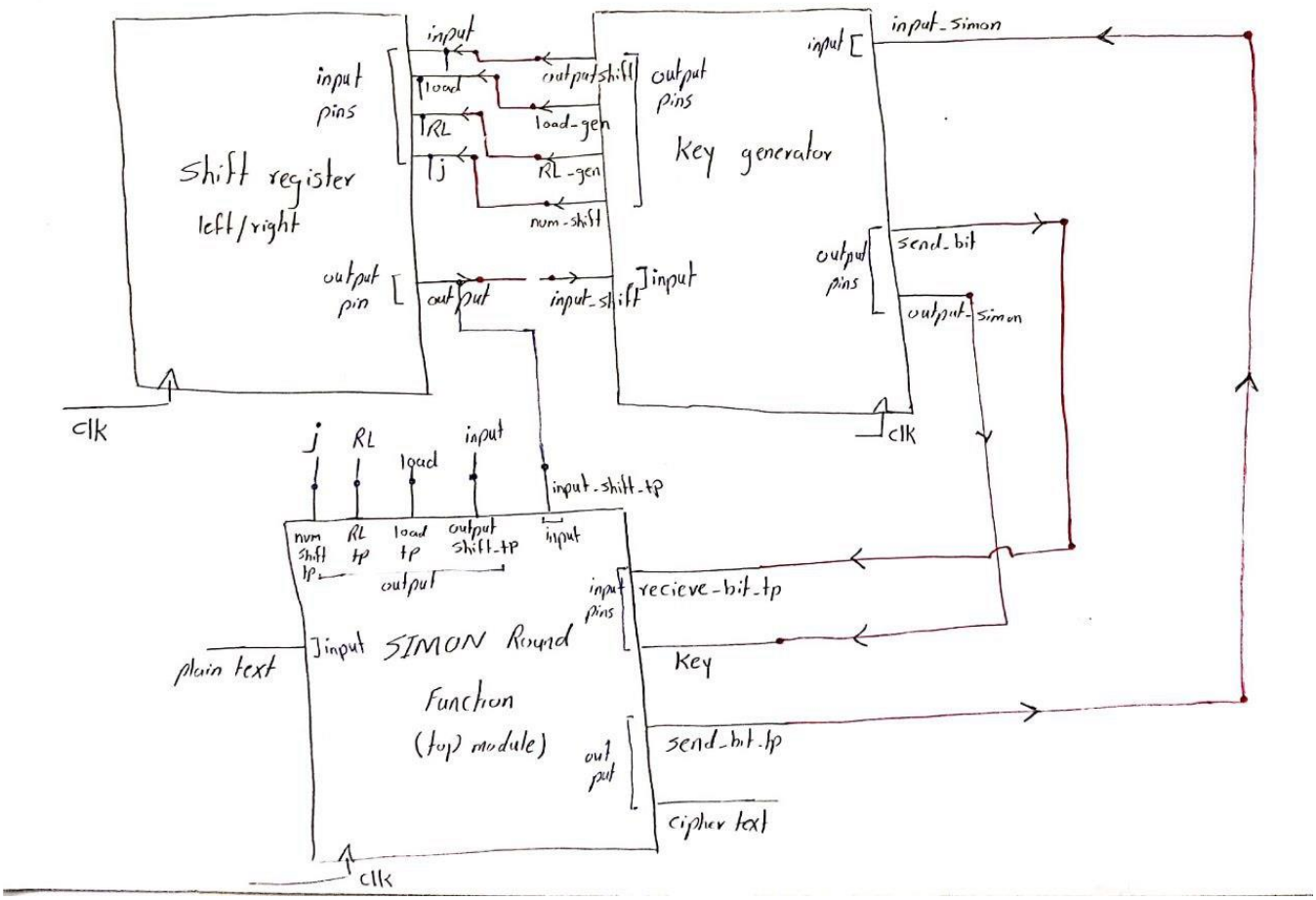
Figure 3.1: Feistel stepping of the SIMON round function.

For this question:

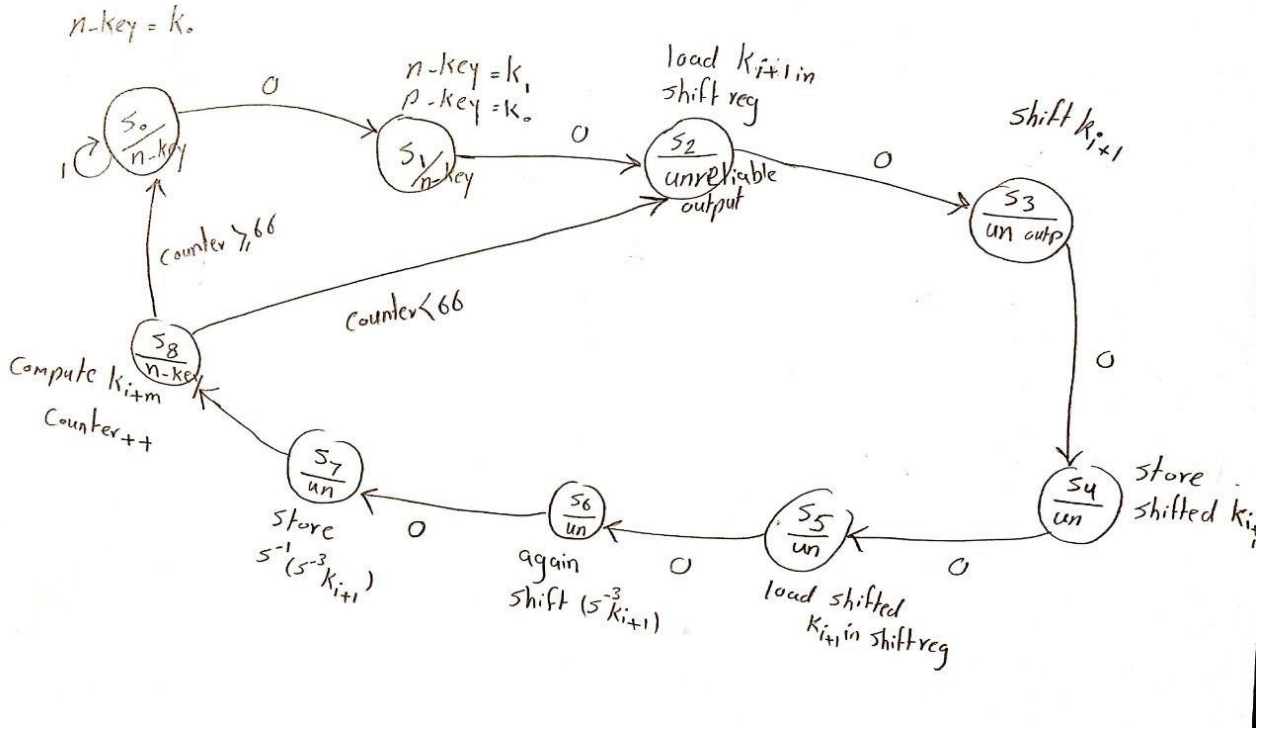


Code

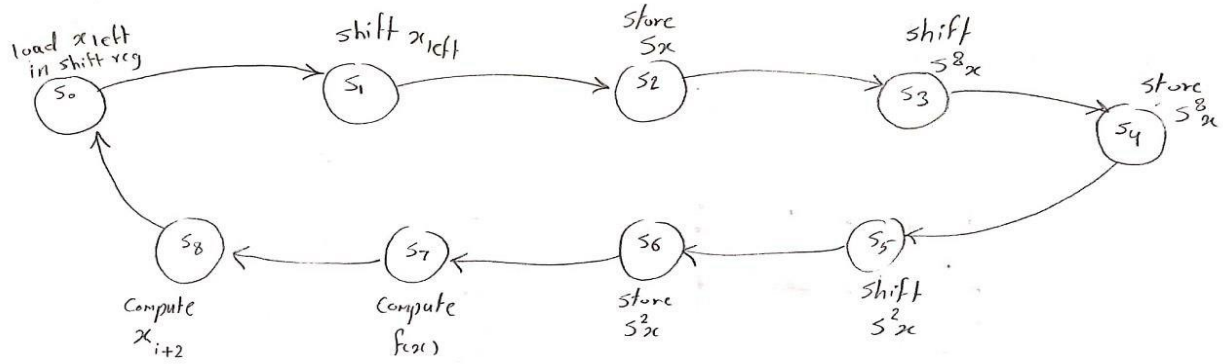
Block diagram



Key generator state machine:



SIMON function state machine:



References

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2013). The SIMON and SPECK families of lightweight block ciphers. *cryptology eprint archive*.

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2015). SIMON and SPECK: Block Ciphers for the Internet of Things. *Cryptology ePrint Archive*.

Abed, S. E., Jaffal, R., Mohd, B. J., and Alshayeji, M. (2019). FPGA modeling and optimization of a Simon lightweight block cipher. *Sensors*, *19*(4), 913.