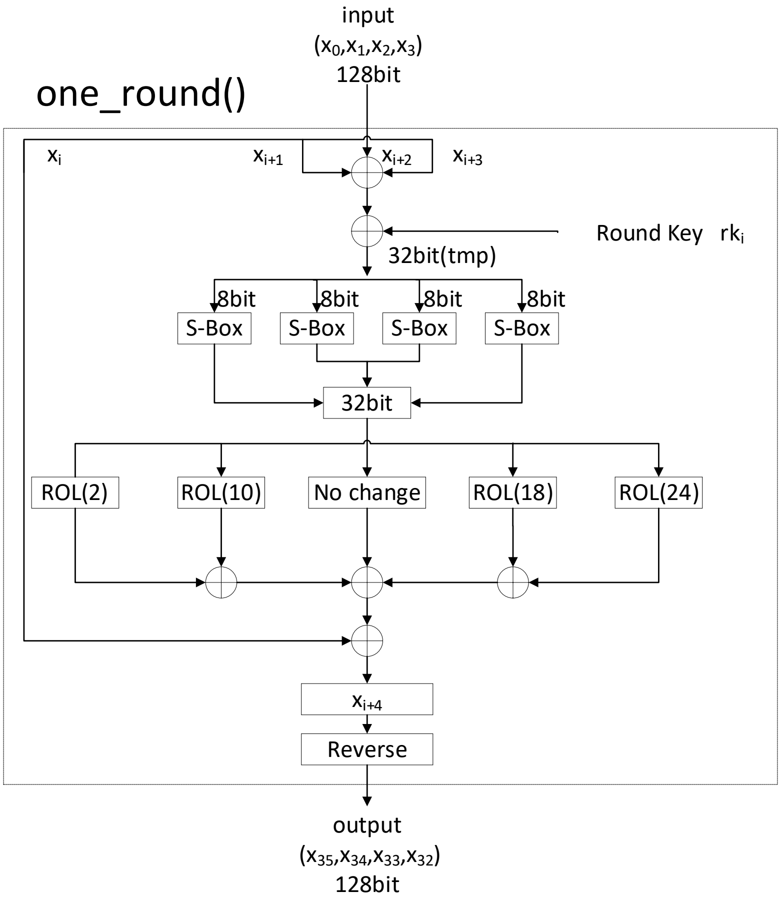
A brief description of SM4 algorithm

SM4 is the block cipher algorithm in China's Standards of Encryption Algorithms.

For more information, see ISO/IEC 18033-3:2010/Amd 1:2021(https://www.iso.org/standard/81564.html).
http://www.gmbz.org.cn/main/viewfile/20180108015408199368.html (In Chinese).

SM4 is a block symmetric cryptographic algorithm, with plaintext, ciphertext, and key lengths of 128 bits. The SM4 algorithm mainly includes encryption and decryption algorithms and key expansion algorithms, adopting a mathematical structure of 32 rounds of nonlinear iteration, where each iteration operation in the algorithm is a round of nonlinear transformation. The main operations include XOR, synthetic permutation, nonlinear iteration, inverse transformation, cyclic shift, and S-box transformation. The mathematical architecture, operation rules, and operations of encryption and decryption algorithms are completely identical, and decryption operations only require the reverse order use of the round keys generated in the encryption algorithm. The main operation of SM4 is a non-equilibrium Feistel network.

Encryption: fn one_round(rk []u32, input []u8, mut output []u8)



Eg. the round function ***one_round*** act as:

(1) ***tmp*** = x_0 ^ x_1 ^ x_2 ^ rk_0;

(2) Divide the 32bit ***tmp*** into 4 parts, and use S-Box transfer them, then combine the results into another 32bit ***result***;

(3) Rotate Left Shift(ROL) ***result***, 2, 10, 0, 18, 24 bits, respectively, and XOR them;

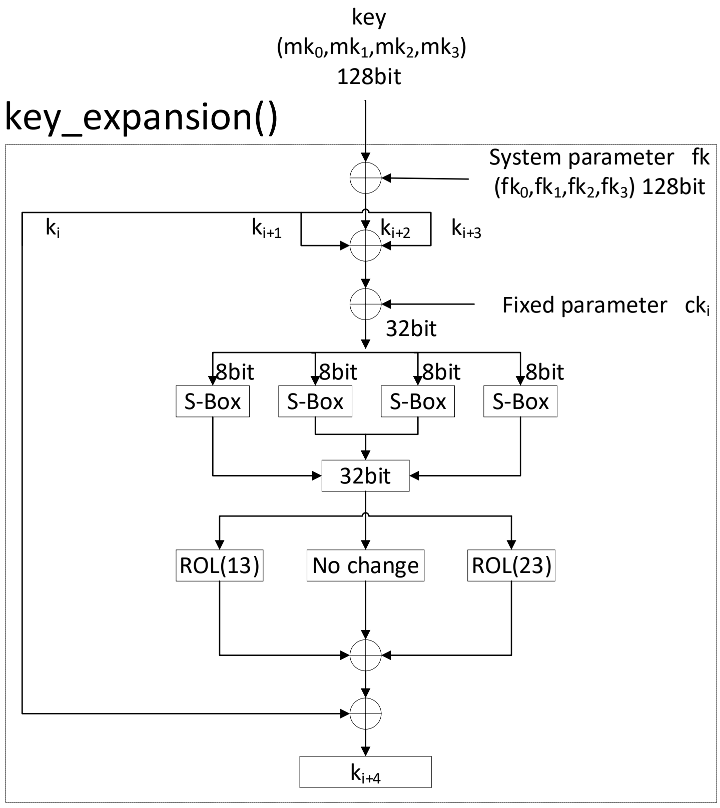
(4) XOR with x_0, we got x_4;

(5) Loop (1)~(4) 32 times, we got x_0~x_{35};

(6) (x_{35},x_{34},x_{33},x_{32}) is the output.

NOTE: (2)~(3) can be optimized, see following.

Key expansion: fn key_expansion(mut rk []u32, key []u8)



Decryption:

For decryption, is simply the same step as encryption, but with reverse round keys. For example, the encryption round keys are (rk_0,rk_1,rk_2,···,rk_{31}), then the decryption round keys should be (rk_{31},rk_{30},···,rk_1,rk_0). This is done in ***new_cipher().***

Optimization:

(1) In ***one_round()***, replace x_0~x_{35} with x_0~x_3, this can slightly increase performance;

(2) In ***one_round()***, replace s-box, ROL operations with pre-calculated ***table0~table3***, this can greatly increase performance.