/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FileName:

SM3.h Version:

SM3\_V1.1 Date:

Sep 18,2016 Description:

This headfile provide macro defination, parameter definition

and function declaration needed in SM3 algorithm implement Function List:

1.SM3\_256 //calls SM3\_init, SM3\_process and SM3\_done to calculate hash value

2.SM3\_init //init the SM3 state

3.SM3\_process //compress the the first len/64 blocks of the message

4.SM3\_done //compress the rest message and output the hash value

5.SM3\_compress //called by SM3\_process and SM3\_done, compress a single block of message

6.BiToW //called by SM3\_compress,to calculate W from Bi

7.WToW1 //called by SM3\_compress, calculate W' from W

8.CF //called by SM3\_compress, to calculate CF function.

9.BigEndian //called by SM3\_compress and SM3\_done.GM/T 0004-2012 requires to use

big-endian.

//if CPU uses little-endian, BigEndian function is a necessary call to

change the

//little-endian format into big-endian format.

10.SM3\_SelfTest //test whether the SM3 calculation is correct by comparing the hash result

with the standard data History:

1. Date: Sep 18,2016

Author: Mao Yingying, Huo Lili

Modification: 1)add notes to all the functions 2)add SM3\_SelfTest function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <string.h>

#define SM3\_len 256

#define SM3\_T1 0x79CC4519

#define SM3\_T2 0x7A879D8A

#define SM3\_IVA 0x7380166f

#define SM3\_IVB 0x4914b2b9

#define SM3\_IVC 0x172442d7

#define SM3\_IVD 0xda8a0600

#define SM3\_IVE 0xa96f30bc

#define SM3\_IVF 0x163138aa

#define SM3\_IVG 0xe38dee4d

#define SM3\_IVH 0xb0fb0e4e

/\* Various logical functions \*/

#define SM3\_p1(x) (x^SM3\_rotl32(x,15)^SM3\_rotl32(x,23))

#define SM3\_p0(x) (x^SM3\_rotl32(x,9)^SM3\_rotl32(x,17))

#define SM3\_ff0(a,b,c) (a^b^c)

#define SM3\_ff1(a,b,c) ((a&b) | (a&c) | (b&c))

#define SM3\_gg0(e,f,g) (e^f^g)

#define SM3\_gg1(e,f,g) ((e&f) | ((~e)&g))

#define SM3\_rotl32(x,n) ((((unsigned int) x) << n) | (((unsigned int) x) >> (32 - n)))

#define SM3\_rotr32(x,n) ((((unsigned int) x) >> n) (((unsigned int) x) << (32 - n)))

typedef struct {

unsigned int state[8]; unsigned int length;

unsigned int curlen;

unsigned char buf[64];

} SM3\_STATE;

void BiToWj(unsigned int Bi[], unsigned int Wj[]);

void WjToWj1(unsigned int Wj[], unsigned int Wj1[]);

void CF(unsigned int Wj[], unsigned int Wj1[], unsigned int V[]);

void BigEndian(unsigned char src[], unsigned int bytelen, unsigned char des[]);

void SM3\_init(SM3\_STATE\* md);

void SM3\_compress(SM3\_STATE\* md);

void SM3\_process(SM3\_STATE\* md, unsigned char buf[], int len);

void SM3\_done(SM3\_STATE\* md, unsigned char\* hash);

void SM3\_256(unsigned char buf[], int len, unsigned char hash[]);

int SM3\_SelfTest();