```
/*****************************************************************************
  File name:    SM2_KEY_EX.h
  Version:      V1.1
  Date:         Oct 9,2016
  Description:  implementation of SM2 Key Exchange Protocol
  Function List:
    1.SM2_Init            // initiate SM2 curve, should be called before any calculation on
curve.
    2.SM2_KeyEx_Init_I    // Step A1 to A3,the first host (initiator A) generates a random number
rA and
                          // calculates RA which the second host(responder B) receives
    3.SM2_KeyEx_Re_I      // Step B1 to B9, responder B generates RB, and calculates a secret
shared key
                          // out of RA and RB, RB should be sent the initiator A
    4.SM2_KeyEx_Init_II   // Step A4 to A10, initiator A calculates the secret key out of RA and
RB, and calculates a hash
                          // value which responder B might verifies
    5.SM2_KeyEx_Re_II     // Step B10 (optional) verifies the hash value received from initiator
A
    6.SM2_KeyEX_SelfTest  // test whether the calculation is correct by comparing the result with
the standard data
    7.SM2_W               //calculation of w
    8.SM3_Z               //calculation of ZA or ZB
    9.Test_Point          // test if the given point is on SM2 curve
    10.Test_Pubkey        // test if the given public key is valid
    11.SM2_KeyGeneration  //calculate a pubKey out of a given priKey

Declaration:
    The SM2 algorithm source code is for academic, non-profit or non-commercial use only. SM2
implementation is
    based on MIRACL whose copyright belongs to Shamus Software Ltd. We are in no position to provide
MIRACL library
    or any permission to use it. For commercial use, please apply to Shamus Software Ltd for a
license.
Notes:
    The MIRACL system must be initialized before attempting to use any other MIRACL routines.
*****************************************************************************/

#include "miracl.h"
#include "mirdef.h"

#define SM2_WORDSIZE        8
#define SM2_NUMBITS         256
#define SM2_NUMWORD         (SM2_NUMBITS/SM2_WORDSIZE)    //32
```

```
#define ERR_INFINITY_POINT      0x00000001
#define ERR_NOT_VALID_ELEMENT   0x00000002
#define ERR_NOT_VALID_POINT     0x00000003
#define ERR_ORDER               0x00000004
#define ERR_ECURVE_INIT         0x00000005
#define ERR_KEYEX_RA            0x00000006
#define ERR_KEYEX_RB            0x00000007
#define ERR_EQUAL_S1SB          0x00000008
#define ERR_EQUAL_S2SA          0x00000009
#define ERR_SELFTEST_Z          0x0000000A
#define ERR_SELFTEST_INI_I      0x0000000B
#define ERR_SELFTEST_RES_I      0x0000000C
#define ERR_SELFTEST_INI_II     0x0000000D

unsigned char SM2_p[32] =
{0xFF,0xFF,0xFF,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
unsigned char SM2_a[32] =
{0xFF,0xFF,0xFF,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFC};
unsigned char SM2_b[32] =
{0x28,0xE9,0xFA,0x9E,0x9D,0x9F,0x5E,0x34,0x4D,0x5A,0x9E,0x4B,0xCF,0x65,0x09,0xA7,
0xF3,0x97,0x89,0xF5,0x15,0xAB,0x8F,0x92,0xDD,0xBC,0xBD,0x41,0x4D,0x94,0x0E,0x93};
unsigned char SM2_n[32] =
{0xFF,0xFF,0xFF,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x72,0x03,0xDF,0x6B,0x21,0xC6,0x05,0x2B,0x53,0xBB,0xF4,0x09,0x39,0xD5,0x41,0x23};
unsigned char SM2_Gx[32]=
{0x32,0xC4,0xAE,0x2C,0x1F,0x19,0x81,0x19,0x5F,0x99,0x04,0x46,0x6A,0x39,0xC9,0x94,
0x8F,0xE3,0x0B,0xBF,0xF2,0x66,0x0B,0xE1,0x71,0x5A,0x45,0x89,0x33,0x4C,0x74,0xC7};
unsigned char SM2_Gy[32]=
{0xBC,0x37,0x36,0xA2,0xF4,0xF6,0x77,0x9C,0x59,0xBD,0xCE,0xE3,0x6B,0x69,0x21,0x53,
0xD0,0xA9,0x87,0x7C,0xC6,0x2A,0x47,0x40,0x02,0xDF,0x32,0xE5,0x21,0x39,0xF0,0xA0};
unsigned char SM2_h[32]=
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01};

big para_p,para_a,para_b,para_n,para_Gx,para_Gy,para_h;
epoint *G;
miracl *mip;


int SM2_W(big n);
void SM3_Z(unsigned char ID[], unsigned short int ELAN, epoint* pubKey, unsigned char hash[]);
```

```
int Test_Point(epoint* point);
int Test_PubKey(epoint *pubKey);
int SM2_Init();
int SM2_KeyGeneration(big priKey,epoint *pubKey);
int SM2_KeyEx_Init_I(big ra, epoint* RA);
int SM2_KeyEx_Re_I(big rb, big dB, epoint* RA, epoint* PA, unsigned char ZA[],unsigned char
ZB[],unsigned char K[],int klen,epoint* RB, epoint* V,unsigned char hash[]);
int SM2_KeyEx_Init_II(big ra, big dA, epoint* RA,epoint* RB, epoint* PB, unsigned char
ZA[],unsigned char ZB[],unsigned char SB[],unsigned char K[],int klen,unsigned char SA[]);
int SM2_KeyEx_Re_II(epoint *V,epoint *RA,epoint *RB,unsigned char ZA[],unsigned char
ZB[],unsigned char SA[]);
int SM2_KeyEx_SelfTest();
```