```
/*************************************************************************
  File name:    SM2_sv.h
  Version:      SM2_sv_V1.0
  Date:         Sep 27,2016
  Description:  implementation of SM2 signature algorithm and verification algorithm
  Function List:
        1.SM2_Init                      //initiate SM2 curve
        2.Test_Point                    //test if the given point is on SM2 curve
        3.Test_PubKey                   //test if the given public key is valid
        4.Test_Zero                     //test if the big x equals zero
        5.Test_n                        //test if the big x equals n
        6.Test_Range                    //test if the big x belong to the range[1,n-1]
        7.SM2_KeyGeneration             //generate public key
        8.SM2_Sign                      //SM2 signature algorithm
        9.SM2_Verify                    //SM2 verification
        10.SM2_SelfCheck()              //SM2 slef-check
        11.SM3_256()                    //this function can be found in SM3.c and SM3.h

 Notes:
    This SM2 implementation source code can be used for academic, non-profit making or
non-commercial use only.
    This SM2 implementation is created on MIRACL. SM2 implementation source code provider does
not provide MIRACL library, MIRACL license or any permission to use MIRACL library. Any commercial
use of MIRACL requires a license which may be obtained from Shamus Software Ltd.
*************************************************************************/


#include<string.h>
#include<malloc.h>
#include "miracl.h"



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

#define ERR_ECURVE_INIT               0x00000001
#define ERR_INFINITY_POINT            0x00000002
#define ERR_NOT_VALID_POINT           0x00000003
#define ERR_ORDER                     0x00000004
#define ERR_NOT_VALID_ELEMENT         0x00000005
#define ERR_GENERATE_R                0x00000006
#define ERR_GENERATE_S                0x00000007
```

```
#define ERR_OUTRANGE_R                0x00000008
#define ERR_OUTRANGE_S                0x00000009
#define ERR_GENERATE_T                0x0000000A
#define ERR_PUBKEY_INIT               0x0000000B
#define ERR_DATA_MEMCMP               0x0000000C


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_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_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};

big Gx,Gy,p,a,b,n;

epoint *G,*nG;


int SM2_Init();
int Test_Point(epoint* point);
int Test_PubKey(epoint *pubKey);
int Test_Zero(big x);
int Test_n(big x);
int Test_Range(big x);
int SM2_KeyGeneration(unsigned char PriKey[],unsigned char Px[],unsigned char Py[]);
```

```
int SM2_Sign(unsigned char *message,int len,unsigned char ZA[],unsigned char rand[],unsigned char d[],unsigned char R[],unsigned char S[]);
int SM2_Verify(unsigned char *message,int len,unsigned char ZA[],unsigned char Px[],unsigned char Py[],unsigned char R[],unsigned char S[]);
int SM2_SelfCheck();
```