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

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](#bookmark1)

[#define SM2\_NUMBITS 256](#bookmark2)

[#define SM2\_NUMWORD (SM2\_NUMBITS/SM2\_WORDSIZE) //32](#bookmark3)

#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();