GMT-0009-2012 in English

6 Data Conversion

In SM2 calculations these 4 mechanisms are used to convert between Octet Strings and Bit Strings.

6.1 Bit String to Octet String Conversion

If the bit string length is not a multiple of 8, add ‘0’ to its left to ensure the length is divisible by 8. Then create an octet string as below.

INPUT: Bit String B of length blen
OUTPUT: Octet String M of length mlen, mlen is the whole number portion of (blen + 7)/8.

STEPS: Converting bit string B = B_0 B_1 … B_(blen-1) to octet string M = M_0 M_1 … M_(mlen-1), 0 <= I <= mlen-1.

M_i = B_(blen-8-8(mlen-1-i)) B_(blen-8-8(mlen-1-i)) … B_(blen-1-8(mlen-1-i))
M_0: the leftmost 8-blen%8 position set to 0, the right of it is B_0 B_1 … B_(8-8(len) + blen - 1)

OUTPUT M.

6.2 Octet String to Bit String Conversion

Octet String to Bit String Conversion as follows:

INPUT: Octet String M of length mlen
OUTPUT: Octet String B of length blen = (8 x mlen)

STEPS: Converting octet string M = M_0 M_1 … M_(mlen-1) to bit string B = B_0 B_1 … B_(blen-1), 0 <= I <= mlen-1.

B_(8i) B_(8i+1) … B_(8i+7) = M_i

OUTPUT B.

6.3 Integer to Octet String Conversion

Converting an integer into an octet string, the basic method is to first represent the integer in binary, and the output the bit string as an octet string. This is the conversation flow.

INPUT: A non-negative x; expected octet string length mlen. Where 2^(8 (mlen)) > x
OUTPUT: Octet string M of length mlen

STEPS: Convert a number x based on 2^8 = 256, x = x_{mlen-1}2^(8(mlen-1)) + x_{mlen-2}2^(8(mlen-2)) + … x_{1}2^8 + x_0 into M = M_{0}M_{1} … M_{mlen-1}

Given 0 <= I <= mlen － 1, set M_{i} = x_{mlen-1-i}

OUTPUT: M

6.4 Octet String to Integer Conversion

It is simple to convert an Octet String into a Base 256 Integer. Conversion method below.

INPUT: Octet String M
OUTPUT: Whole number x

STEPS: Convert M = M_0 M_1 … M_{mlen-1} into whole number x.

M_i is like a whole number within [0~255]

x = SUM_{i=0}^{mlen-1} 2^{8(mlen-1-i)} M_i

Output x.

7 Data Format

7.1 Secret Key Data Format

SM2 secret key data format as described in ASN.1 is：

```
SM2PrivateKey ::= INTEGER
```

SM2 public key data format as described in ASN.1 is：

```
SM2PublicKey ::= BIT STRING
```

SM2PublicKey is of type BIT STRING，content is 04‖X‖Y，within that，X and Y specifies the x- and y-coordinates of the public key，each 256-bits long.

7.2 Encrypted Data Format

SM2 encrypted data format as described in ASN.1 is：

```
SM2Cipher ::= SEQENCE{
XCoordinate       INTEGER,                      -- x-coordinate
YCoordinate       INTEGER,                      -- y-coordinate
HASH              OCTET STRING SIZE(32),        -- hash value
CipherText        OCTET STRING                  -- ciphertext

}
```

HASH is the hash value calculated from SM3, with a fixed bit length of 256-bits. CipherText is of same length as its plaintext.

7.3 Signature Data Format

SM2 signature data format as described in ASN.1 is：

```
SM2Signature ::= SEQUENCE{
R                    INTEGER,              -- first portion of signature
```

```
  S                    INTEGER               -- second portion of signature
  }
```

R and S are of 256 bits long.

7.4 Enveloped Secret Key data format

When transferring a SM2 secret key, the SM2 secret key should be encrypted. The encryption method is:

Create a symmetric secret key;

According to the necessary calculation methods, encrypt the SM2 private key to obtain the private key’s ciphertext. If the symmetric encryption method is a block cipher, utilize ECB mode;

Utilize SM2 public key to encrypt the symmetric secret key to obtain symmetric secret key ciphertext;

Put the SM2 private key ciphertext, symmetric secret key ciphertext into an Enveloped Key Data Format.

SM2 Enveloped Secret Key data format as described in ASN.1 is:

```
  SM2EnvelopedKey ::=    SEQUENCE{
  symAlgID                  AlgorithmIdentifier,    -- Symmetric Encryption
                                                    Algorithm ID
  symEncryptedKey           SM2Cipher,              -- Symmetric Encryption Key
                                                    encrypted by SM2 Public Key
  Sm2PublicKey              SM2PublicKey,           -- SM2 Public Key
  Sm2EncryptedPrivateKey    BIT STRING              -- SM2 Private Key Encrypted
                                                    by Symmetric Encryption
  }
```

8 Pre-processing

8.1 Pre-processing 1

Pre-processing 1 is to use the signing party’s identifier and signature public key, to calculate value Z. Z is used in pre-processing 2, which is the SM2 key negotiation.

INPUT:	ID	Byte String	User Identifier
	Q	SM2PublicKey	User Public Key
OUTPUT:	Z	Byte String	Output of Pre-processing 1

Formula is:

$$Z = SM3(ENTL \| ID \| a \| b \| x_G \| y_G \| x_A \| y_A)$$

Where:

ENTL	is a 2 byte field indicating bit-length of ID;
ID	is the User Identifier;
a、b	is the System Curve Parameter;
x_G、y_G	are the base points;
x_A、y_A	represents the User’s Public Key 。

For detailed calculations see GB/T 0003 and GB/T 0004.

8.2 Pre-processing 2

Pre-processing 2 is the process of using value Z and the message to be signed, utilize SM3 to calculate hash value H. Hash value H is used for SM2 digital signature.

INPUT:	Z	B	Input to Pre-processing 2
	M	Byte String	Message to be signed
OUTPUT:	H	Byte String	Hash Value

Calculation:

H = SM3(Z ‖ M)

For detailed calculations see GB/T 0003 and GB/T 0004.

9 Calculation Process

9.1 Generation of secret key

SM2 secret key generation is the process of using SM2 calculations to create a pair of keys, this pair of keys include a private key and the corresponding public key. The private key is of 256-bits long and the public key 512-bits long.

INPUT:	None		
OUTPUT:	k	SM2PrivateKey	SM2 Private Key
	Q	SM2PublicKey	SM2 Public Key

For detailed calculations see GB/T 0003.

9.2 Encryption

SM2 Encryption is to use the public key of the given key pair to perform encryption, in order to generate ciphertext. This ciphertext can only be decrypted by the corresponding private key.

INPUT:	Q	SM2PublicKey	SM2 Public Key
	m	Byte String	Plaintext To Be Encrypted
OUTPUT:	c	SM2Cipher	Ciphertext

Where:

Output c is in the format defined by 7.2;

Output c’s XCoordinate, YCoordinate are randomly generated x- and y-coordinates;

Output c’s HASH was calculated as:

HASH = SM3(x ‖ m ‖ y)

Wherein, x, y is Q’s x- and y-coordinates;

Output c’s CipherText is the ciphertext, its length is identical to that of the plaintext.

For detailed calculations see GB/T 0003 and GB/T 0004.

9.3 Decryption

SM2 Decryption means using a private key to decrypt a ciphertext encrypted by the corresponding public key to obtain the plaintext.

INPUT:	d	SM2PrivateKey	SM2 private key
	c	SM2Cipher	Ciphertext
OUTPUT:	m	Byte String	Corresponding plaintext to ciphertext

m is the decrypted plaintext of SM2Cipher, the length of plaintext is the same as that of the input ciphertext c.

For detailed calculations see GB/T 0003.

9.4 Digital Signature

SM2 signature is to obtain a signature by using the result of pre-processing 2’s together with the signer’s private key through the signing process.

INPUT:	d	SM2PrivateKey	Signer’s Private Key
	H	Byte String	Result of Pre-processing 2
OUTPUT:	sign	SM2Signature	Signature value

For detailed calculations see GB/T 0003.

9.5 Signature Verification

SM2 signature verification is to verify a signature through using the result of pre-processing 2, the signature value and the signer’s public key, through a verification process.

INPUT:	H	字节串	Result of Pre-processing 2
	sign	SM2Signature	Signature value
	Q	PublicKey	Signer’s Public Key.

OUTPUT: “true” if “validation passed”, “false” if “validation failed”.

For detailed calculations see GB/T 0003.