GB

NATIONAL STANDARD OF THE

PEOPLE'S REPUBLIC OF CHINA

ICS 35.040

L 80

GB/T 17964-2008

Replacing GB/T 17964-2000

Information technology - Security techniques - Modes of operation for a block cipher

信息安全技术

分组密码算法的工作模式

Issued on: June 26, 2008 **Implemented on: November 01, 2008**

Issued by: General Administration of Quality Supervision, Inspection and Quarantine;

Standardization Administration of the People's Republic of China.

Table of Contents

Information technology - Security techniques - Modes of operation for a block cipher

1 Scope

This Standard specifies seven operation modes of block cipher algorithm, so as to standardize the use of block cipher.

2 Normative references

The provisions in following documents become the provisions of this Standard through reference in this Standard. For dated references, the subsequent amendments (excluding corrigendum) or revisions do not apply to this Standard, however, parties who reach an agreement based on this Standard are encouraged to study if the latest versions of these documents are applicable. For undated references, the latest edition of the referenced document applies.

GB/T 1988-1998, *Information technology - 7-bit Coded character set for information interchange* (eqv ISO/IEC 646:1991)

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1 Terms

3.1.1 block chaining (BC) operation mode

an operation mode of block cipher algorithm; the current plaintext block is different from the exclusive OR values of all previous ciphertext blocks or operated then encrypted to obtain the current ciphertext block

3.1.2 block cipher

also known as block cipher algorithm; it is a symmetric cryptographic algorithm; it divides plaintext into fixed-length blocks for encryption

3.1.3 block cipher operation mode

a use mode of block cipher algorithm, mainly including electronic codebook (ECB) operation mode, cipher block chaining (CBC) operation mode, cipher

starting data that is brought for data transformation so as to increase security or synchronize cipher devices during cryptographic transformation

3.1.14 key

key information or parameter that controls cryptographic transformation

3.1.15 output feedback with a nonlinear function (OFBNLF) operation mode

an operation mode of block cipher algorithm; it is a variant of OFB and ECB; its key varies with each block

3.1.16 output feedback (OFB) operation mode

an operation mode that block cipher algorithm is used to construct sequence cipher; use the output of the current time of this algorithm as the input of the next moment

3.1.17 plain text/clear text

data to be encrypted

3.2 Definitions

3.2.1 Encryption expression

In this Standard, the functional relationship specified by block cipher is recorded as:

$$C = E_K(P)$$

Where,

P is plaintext block;

C is ciphertext block;

K is key;

E_K is encryption operation that uses key K.

3.2.2 Decryption expression

The corresponding decryption function is recorded as:

$$P = D_K(C)$$

A special case of this function starts with the m-bit variable I(m) of all "1"s and moves the k-bit variable F into it. The result is:

$$S_k(I(m)|F)=(1,1,\cdots,1,f_1,f_2,\cdots,f_k) \quad (k<m)$$
$$S_k(I(m)|F)=(f_1,f_2,\cdots,f_k) \quad (k=m)$$

Where, the left-most m-k bit is "1".

4 Abbreviations and symbols

AES	advanced encryption standard
BC	block chaining
CBC	cipher block chaining
CFB	cipher feedback
CTR	counter
DEA	data encryption algorithm
ECB	electronic codebook
IV	initialization value
OFB	output feedback
OFBNLF	output feedback with a nonlinear function

5 Electronic codebook (ECB) operation mode

5.1 Definition of variable

a) Sequence consisting of q plaintext blocks P_1, P_2, ..., P_q. Each block is n bits.

b) Key K.

c) Result sequence consisting of q ciphertext blocks C_1, C_2, ..., C_q. Each block is n bits.

5.2 Description of ECB encryption

$$C_i = E_K(P_i) \quad i=1,2,\cdots,q$$

5.3 Description of ECB decryption

$$P_i = D_K(C_i) \quad i=1,2,\cdots,q$$

7 Cipher feedback (CFB) operation mode

7.1 Definition of parameter

- Size of feedback buffer r ($n \le r \le 2n$);
- Size of feedback variable k ($1 \le k \le n$);
- Size of plaintext variable j ($1 \le j \le k$).

NOTE: r-k can be less than n. Figure 2 shows special case of r-k>n.

7.2 Definition of variable

a) Input variable

1) Sequence consisting of q plaintext blocks P_1, P_2, ..., P_q. Each block is j bits.

2) Key K.

3) Initialization value of r-bit IV.

b) Intermediate results

1) Sequence consisting of q key input blocks X_1, X_2, ..., X_q. Each block is n bits.

2) Sequence consisting of q key output blocks Y_1, Y_2, ..., Y_q. Each block is n bits.

3) Sequence consisting of q variables Z_1, Z_2, ..., Z_q. Each block is j bits.

4) Sequence consisting of q-1 feedback variables F_1, F_2, ..., F_{q-1}. Each variable is k bits.

5) Sequence consisting of q-1 feedback buffer contents FB_1, FB_2, ..., FB_{q-1}. Each block is n bits.

c) Output variable

Sequence consisting of q ciphertext variables C_1, C_2, ..., C_q. Each block is j bits.

7.3 Description of CFB encryption

Initialization value of feedback buffer FB is:

$$P_i = C_i \oplus Z_i$$

e) Generate feedback variable:

$$F_i = S_j(I(k) \mid C_i)$$

f) FB bit-shift operation:

$$FB_{i+1} = S_k(FB_i \mid F_i)$$

For i=1, 2, …, q, repeat the above steps. The last cycle ends at step d). This process is shown in the right half of Figure 2. The leftmost j bit of output block Y of block cipher is used to decrypt j-bit ciphertext variable through modulo 2 plus. Other bits of Y are discarded. The plaintext and ciphertext variables are numbered from 1 to j.

Place k-j "1" bits in the leftmost position of ciphertext variable. Extend ciphertext variable to a k-bit feedback variable F. Then move the bits of the feedback buffer FB to the left by k positions. Place F into the rightmost k positions. Generate a new feedback buffer FB value. In this bit-shift operation, the leftmost k bit of FB is discarded. The new n-bit at the far left of FB is used as the next input X in encryption process.

NOTE: See Annex A for operation nature of CFB mode.

Example: See Annex B for example of CFB mode.

7.5 Suggestion

It is recommended to use CFB method that j and k values are equal. According to this suggested form (j=k), the steps e) of encryption operation and decryption operation can be written as:

$$F_i = C_i \quad \text{(when j=k)}$$

8 Output feedback (OFB) operation mode

8.1 Definition of parameter

OFB operation is defined by one parameter. This parameter is the size of plaintext variable j (1≤j≤n).

$$Z_i = Y_i \sim j$$

c) Generate plaintext variable:

$$P_i = C_i \oplus Z_i$$

d) Feedback operation:

$$X_{i+1} = Y_i$$

For i=1, 2, …, q, repeat the above steps. The last cycle ends at step c). This process is shown in the right half of Figure 3. Values of X_i and Y_i, during encryption, shall be same with corresponding values. Only step c) is different.

NOTE: See Annex A for operation nature of OFB mode.

Example: See Annex B for example of OFB mode.

9 Counter (CTR) operation mode

9.1 Definition of variable

a) Input variable

1) Sequence consisting of q plaintext variables P_1, P_2, ..., P_q (where, P_1, P_2, …, P_{q-1} are all n bits, P_q is k bits).

2) Key K.

3) q count sequences T_1, …, T_{q-1}, T_q. Each block is n bits.

b) Intermediate results

1) Sequence consisting of q cipher output blocks X_1, X_2, ..., X_q. Each block is n bits.

2) k-bit cipher output block Z.

c) Output variable

Sequence consisting of q ciphertext variables C_1, C_2, ..., C_q (where, C_1, C_2, …, C_{q-1} are all n bits, C_q is k bits).

Annex A

(normative)

Nature of operation mode

A.1 Nature of electronic codebook (ECB) operation mode

A.1.1 Environment

Binary data exchanged between various computers or between people may have duplicate or shared sequences. In ECB mode, same plaintext block (for same key) generates same ciphertext block.

A.1.2 Nature

The natures of ECB mode are:

a) Encryption or decryption of a block can be performed independently of the other;

b) Rearrangement of ciphertext shall result in a corresponding rearrangement of plaintext block;

c) Same plaintext block (for same key) always generates same ciphertext block, which makes it vulnerable to a "dictionary attack". Such a dictionary is composed of corresponding plaintext and ciphertext blocks.

For information with more than one block, it is generally recommended not to use ECB mode. For those special use cases where repeatability is acceptable or individual blocks must be accessed separately, the usage of ECB can be specified in future standards.

A.1.3 Filling requirements

Only multiples of the block length can be encrypted or decrypted. Other lengths need to be filled to the block length boundary.

A.1.4 Error diffusion

In ECB mode, one or more bit-errors in a ciphertext block shall only affect decryption of the block in which the error occurs. Decryption of a ciphertext block with one or more error bits shall result in a 50% probability of error for each plaintext bit in the corresponding plaintext block.

A.1.5 Block boundary

If the block boundary of decryption or between decryptions is lost (for example due to a bit slip), then before re-establishing the correct block boundary, the synchronization between encryption and decryption shall be lost. If the block boundary is lost, the result of all decryption operations shall be incorrect.

A.3 Nature of cipher feedback (CFB) operation mode

A.3.1 Environment

As long as same key and initialization value are used to encrypt the same plaintext, CFB mode shall generate same ciphertext. Users who care about this nature need to use some way to change the start, key, or initialization value of the plaintext. One possible approach is to add a unique identifier (for example, an incremental counter) to the beginning of each CFB information. When it encrypts a record of which its size cannot be increased, it may use another approach. It uses some value such as initialization value. This value can be calculated from the record and it is unnecessary to know its content (for example, its address in random access storage mode).

A.3.2 Nature

The natures of CFB are:

a) Chaining operation makes ciphertext variable depend on the current and all previous plaintext variables except one variable of which number is certain. This number depends on selection of r, k and j (see Figure 2). Therefore, the rearrangement of the j-bit ciphertext variable does not result in a rearrangement of the corresponding j-bit plaintext variable;

b) Use different IV values to prevent the same plaintext from being encrypted to become the same ciphertext;

c) All encryption and decryption in CFB mode use block cipher encryption operation;

d) The strength of CFB mode depends on the size of k (maximum when j=k) and the relative sizes of j, k, n, and r;

NOTE: j<k shall increase the probability that the value of input block occurs repeatedly. This recurrence shall reveal the linear relationship between the plaintexts.

e) A smaller j value, for each plaintext unit, shall require more block cipher operations. Therefore, it shall cause greater processing overhead;

f) Select r≥n+k to enable pipelined continuous operation of block ciphers.

A.3.3 Filling requirements

d) CTR mode does not depend on plaintext to generate key stream that is used to perform modulo 2 plus to plaintext.

A.5.3 Filling requirements

Counter mode solves the n-bit output problem of which OFB mode is less than block length. It can handle information of any length. Filling is unnecessary.

A.5.4 Error diffusion

CTR mode does not output the diffusion ciphertext error in the generated plaintext. Each error bit in the ciphertext shall only cause an error bit in the decrypted plaintext.

A.5.5 Synchronization

CTR mode is not automatically synchronized. If encryption and decoding are not synchronized, the system needs to be reinitialized. This loss of synchronization may be caused by the insertion or loss of any number of ciphertexts.

A new counter value shall be used for each reinitialization. It is different from the previous counter value that is used with the same key. The reason is that for the same parameters, the same bit stream is generated each time, which shall be vulnerable to "known plaintext attacks".

A.6 Block chaining (BC) operation mode

A.6.1 Environment

In order to use block algorithm in block chaining (BC) mode, it can simply perform exclusive OR to input of block cipher algorithm with exclusive OR values of all previous ciphertext blocks. Just as CBC algorithm, the process shall start with an initialization vector IV.

As long as the same key and initialization plaintext are used to encrypt the same plaintext, BC mode shall generate same ciphertext. Users who care about this nature need some way to change the start, key, or initialization value of the plaintext.

A.6.2 Nature

BC mode has the following natures:

a) Chaining operation makes ciphertext block depend on previous and current plaintext blocks. Therefore, the rearrangement of ciphertext blocks does not result in a rearrangement of the corresponding plaintext blocks;

This is an excerpt of the PDF (Some pages are marked off intentionally)

------ The End ------