Modes of Operation of Block Ciphers

Klaus Pommerening
Fachbereich Physik, Mathematik, Informatik
der Johannes-Gutenberg-Universität
Saarstraße 21
D-55099 Mainz

April 7, 1997—English version August 19, 2014
last change January 20, 2021

A bitblock encryption function $f\colon \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ is primarily defined on blocks of fixed length n. To encrypt longer (or shorter) bit sequences the sender must

1. split the sequence into n-bit blocks,
2. pad the last block if necessary with
   - zeroes or
   - random values or
   - context information.

Then each block is encrypted by f, but in general one uses some sort of "chaining". Four chaining procedures, called "modes of operation" were standardized together with DES:

- ECB,
- CBC,
- CFB,
- OFB.

These chaining procedures apply to each block cipher. The standardization in the context of AES added two more modes:

- CTR,
- XTS.

For a description of the modes a suitable general framework is a "block alphabet" Σ, with $\mathbb{F}_2^n$ as most important example, equipped with a group composition $*$. Furthermore we fix an encryption function

$$f : \Sigma \longrightarrow \Sigma.$$

The dependence on the key doesn't matter in this context and therefore is dropped in the notation.

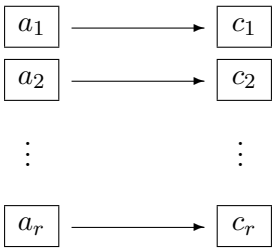
1 ECB = Electronic Code Book

Description

Let r be the number of blocks of the plaintext $(a_1, \ldots, a_r)$.

Encryption: In ECB mode each block is encrypted independently of the other blocks:

$$a = (a_1, \ldots, a_r) \mapsto c = (c_1, \ldots, c_r) \in \Sigma^r \quad \text{with } c_i = f(a_i).$$



Decryption: $a_i = f^{-1}(c_i)$.

Properties

ECB mode simply is a monoalphabetic substitution on Σ. For sufficiently large $\#\Sigma$ this is secure from a ciphertext-only attack. But there are several disadvantages:

- ECB encryption leaks information on identical blocks. Even if the plaintext is not random, the rule of thumb from the Birthday Paradox applies in the interpretation (for $\Sigma = \mathbb{F}_2^n$): "After $2^{n/2}$ bits ECB encryption begins to leak information." Wikipedia has a nice illustration of this effect, see `http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation` The other modes significantly enlarge this bound.

- Building a "codebook" from known plaintext blocks is not unrealistic. For structured messages, say bank transactions, there occur many blocks of known plaintext.

- An active attack by exchanging or inserting single blocks of ciphertext (for example with known, "sympathic" plaintext) is possible. For example an attacker who knows which block contains the receiver of a money transfer could exchange this block with a corresponding block from another transfer for another receiver. He doesn't need to know the key.

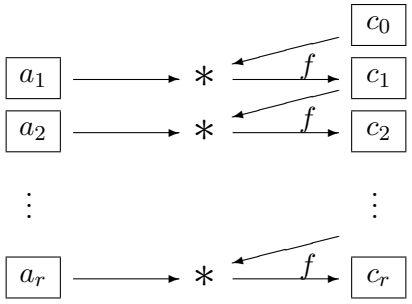
- If the situation allows for an attack with *chosen* plaintext (as in a black box analysis), trial encryption and dictionary attacks can be mounted.

In view of these problems generating diffusion between the plaintext blocks seems a much better approach. In the following sections we look at modes of operation that achieve this effect.

2 CBC = Cipher Block Chaining

Description

Choose a start value c_0 at random (also called IV = "Initialization Vector"). Then the procedure looks like this:



Encryption: In CBC mode the formula for encryption is:

$$\begin{aligned} c_i &:= f(a_i * c_{i-1}) \quad \text{for } i = 1, \ldots, r \\ &= f(a_i * f(a_{i-1} * \cdots f(a_1 * c_0) \ldots)). \end{aligned}$$

Decryption: $a_i = f^{-1}(c_i) * c_{i-1}^{-1}$ for $i = 1, \ldots, r$.

Properties

- Each ciphertext block depends on *all previous* plaintext blocks (diffusion).
- An attacker is not able to replace or insert text blocks unnoticeably.
- Identical plaintext blocks in general encrypt to different ciphertext blocks.
- On the other side an attack with known plaintext is not more difficult, compared with ECB mode.
- Each plaintext block depends on two ciphertext blocks.
- As a consequence a transmission error in a single ciphertext block results in (only) two corrupted plaintext blocks ("self synchronisation" of CBC mode).

Question: *Does it make sense to treat the initialization vector c_0 as secret and use it as an additional key component?* (Then for the example of DES we had 56 proper key bits plus a 64 bit initialization vector, making a total of 120 key bits.)

Answer: No!

Reason: In the decryption process only a_1 depends on c_0. This means that keeping c_0 secret conceals known plaintext only for the first block. If the attacker knows the second or a later plaintext block, then she may determine the key as in ECB mode (by exhaustion, or by an algebraic attack, or by any other attack with known plaintext).

Remarks

1. CBC is the composition $f\circ$ (ciphertext autokey). In the trivial case $f = \mathbf{1}_\Sigma$ only the (completely unsuited) ciphertext autokey cipher with key length 1 is left.

2. (John Kelsey in the mailing list `cryptography@c2.net`, 24 Nov 1999) If there occurs a "collision" $c_i = c_j$ for $i \neq j$, then $f(a_i * c_{i-1}) = f(a_j * c_{j-1})$, hence $a_i * c_{i-1} = a_j * c_{j-1}$ and therefore $a_j^{-1} * a_i = c_{j-1} * c_{i-1}^{-1}$. In this way the attacker gains some information on the plaintext.

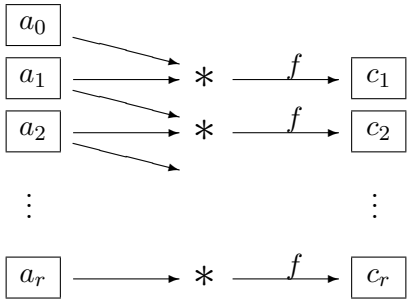
   By the Birthday Paradox this situation is expected after about $\sqrt{\#\Sigma}$ blocks.

   The longer the text, the more such collisions will occur. This effect reassures the rule of thumb for the frequency of key changes: change the key in good time before you encrypt $\sqrt{\#\Sigma}$ blocks.

3 Variants of CBC

Plaintext Autokey

Replacing the ciphertext autokey encryption for CBC mode by plaintext autokey yields the following scheme:



that sometimes is called PBC = Plaintext Block Chaining.

Encryption: After choosing an initialization vector a_0 the formula for encryption is:

$$c_i := f(a_i * a_{i-1}) \quad \text{for } i = 1, \ldots, r.$$

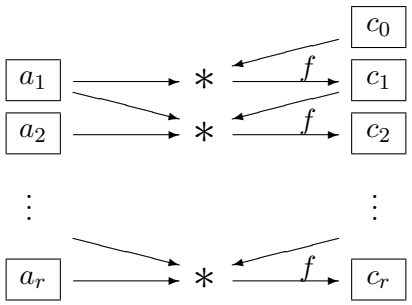
Decription: The formula is:

$$a_i = f^{-1}(c_i) * a_{i-1}^{-1} \quad \text{for } i = 1, \ldots, r.$$

However this method seems not to be in widely accepted use, and there seem to be no relevant results on its security.

PCBC = error-Propagating CBC

This procedure mixes CBC and PBC. It follows the scheme:



Encryption: After choosing the initialization vector $a_0 = e$ (neutral element of the group) encryption is by the formula

$$c_i := f(a_i * a_{i-1} * c_{i-1}) \quad \text{for } i = 1, \ldots, r.$$

In the case of a bitblock cipher we choose $a_0 = 0$, the null block.

Decryption: The formula is

$$a_i = f^{-1}(c_i) * c_{i-1}^{-1} * a_{i-1}^{-1} \quad \text{for } i = 1, \ldots, r.$$

This mode was implemented in early versions of Kerberos but then abandoned.

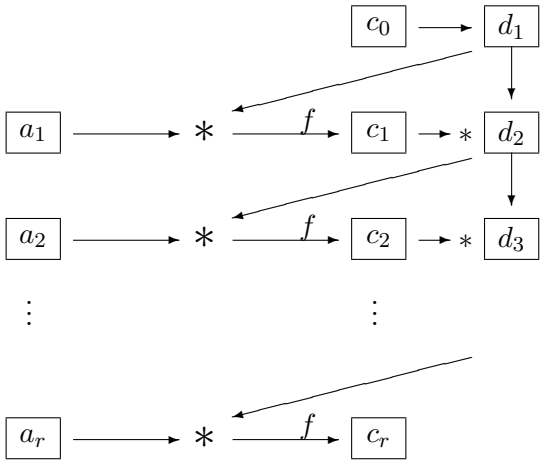
Generalization by Meyer/Matyas

$$c_i := f(a_i * h(a_{i-1}, c_{i-1})) \quad \text{for } i = 1, \ldots, r,$$

where in the case $\Sigma = \mathbb{F}_2^n$ addition modulo 2^n is suggested for h.

BCM = Block Chaining Mode

This mode follows the scheme:



Formula for encryption:

$$\begin{aligned} d_i &:= c_0 * \ldots * c_{i-1}, \\ c_i &:= f(a_i * d_i) \quad \text{for } i = 1, \ldots, r. \end{aligned}$$

An Application of CBC

CBC-MAC (= "Message Authentication Code") is a key-dependent "hash function" that serves for checking the integrity of messages. It is standardized in ISO/IEC 9797 and used in electronic banking.

Sender and receiver of the message—these could be the same person if the MAC used for securing the integrity of a stored file—share the key k and use the encryption function $f = f_k$.

The MAC of a text $a = (a_1, \ldots, a_r)$ is the last ciphertext block where a is encrypted in CBC mode. Hence

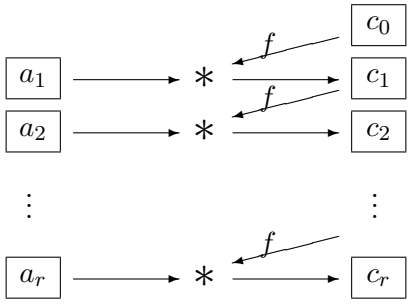
$$\mathrm{MAC}(a) = c_r = f(a_r * f(a_{r-1} * \cdots f(a_1 * c_0) \ldots)).$$

If $\mathrm{MAC}(a)$ is sent together with a, then the receiver may check the authenticity of the sender and the integrity of the content. Only someone who has the key can calculate this value correctly.

The disadvantage of this procedure is the need of sharing a secret k. In a legal dispute each of the two parties can contend a forgery by the other one.

4 CFB = Cipher Feedback

Description (of the simplest version)



Encryption in CFB mode is by the formula

$$\begin{aligned} c_i &:= a_i * f(c_{i-1}) \quad \text{for } i = 1, \ldots, r \\ &= a_i * f(a_{i-1} * f(\cdots a_1 * f(c_0) \ldots)). \end{aligned}$$

Decryption: $a_i = c_i * f(c_{i-1})^{-1}$ for $i = 1, \ldots, r$.

Properties

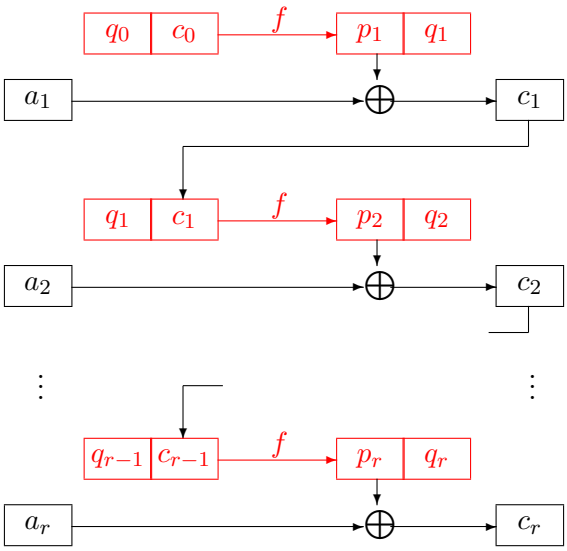
- As before the initialization vector is unsuited as additional key component.
- As before this mode doesn't make an attack with known plaintext more difficult.
- Note that also decryption uses f, not f^{-1}. Therefore:
  - CFB mode doesn't make sense for asymmetric ciphers.
  - On the other hand CFB mode may be used with a (key dependent) one-way or hash function f.
- For the identical map $f = \mathbf{1}_\Sigma$ CFB again reduces to ciphertext autokey.
- (David Wagner) ECB $\circ$ CFB = CBC:

For a proof take c_0 as initialization vector for CFB, and $c'_0 := f(c_0)$ as initialization vector for CBC. Then

$$\begin{aligned} c_1 &= \mathrm{CFB}(a_1) = a_1 * f(c_0), \\ c'_1 &= \mathrm{ECB}(c_1) = f(a_1 * f(c_0)) = f(a_1 * c'_0) = \mathrm{CBC}(a_1), \\ c_2 &= \mathrm{CFB}(a_2) = a_2 * f(c_1), \\ c'_2 &= \mathrm{ECB}(c_2) = f(a_2 * f(c_1)) = f(a_2 * c'_1) = \mathrm{CBC}(a_2), \\ &\text{etc.} \end{aligned}$$

The Standardized Version

... uses a shift register, hence is defined only in the case of $\Sigma = \mathbb{F}_2^n$. Here $1 \leq t \leq n$, and the encryption procedure uses blocks $a_i \in \mathbb{F}_2^t$ of length t. The current ciphertext block c_i of length t is shifted from the right into the shift register (drawn in red):

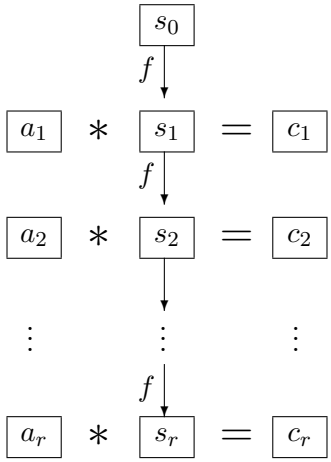


The q_i are bitblocks of length $n - t$.

As it turned out later the security of this more general version decreases with t. Therefore its use is not recommended.

5 OFB = Output Feedback

Description (of the simplest version)



This mode also was originally defined as shift register version. Here too using a blocklength of $t < n$ weakens the security [Jueneman, Crypto 82].

Encryption in OFB mode is by the formula

$$c_i := a_i * s_i, \quad s_i := f(s_{i-1}) \quad \text{for } i = 1, \ldots, r.$$

Decryption by the formula

$$a_i = c_i * s_i^{-1}, \quad s_i := f(s_{i-1}) \quad \text{for } i = 1, \ldots, r.$$

Properties

- There is no diffusion. However identical plaintext blocks in general yield different ciphertext blocks.
- In the case $\Sigma = \mathbb{F}_2^s$ OFB simply is a bitstream cipher where f serves as "random generator".
- If encryption or decryption is time critical, the sender or the receiver (or both) might precalculate the "key stream" s_i.
- Here too the decryption uses only f, not f^{-1}.
- For $\Sigma = \mathbb{F}_2^s$ the cipher is an involution, that is encryption and decryption are the same function. More generally this holds when the group Σ has exponent 2.

- Under an attack with known plaintext the pair (a_1, c_1) reveals the value of s_1, the next pair (a_2, c_2), the value of $s_2 = f(s_1)$. This leads to an attack with known plaintext against the function f itself.
- Keeping the initialization vector s_0 secret doesn't increase the security of the cipher for OFB (like for the other modes).

Variant: Counter Mode CTR

The simplest case is

$$c_i := a_i * f(i) \quad \text{for } i = 1, \ldots, r.$$

There are same slight variants, for example starting with another number than 1.