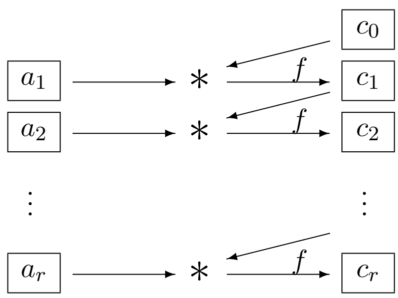
2 CBC = Cipher Block Chaining

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.