CPA

A private-key encryption scheme Π = (Gen, Enc, Dec) has indistinguishable encryptions under a chosen-plaintext attack (or is CPAsecure) if for all probabilistic polynomial-time adversaries A there exists a negligible function negl such that

$$Pr[PrivK^{cpa}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the random coins used by A, as well as the random coins used in the experiment.

Cipher Block Chaining (CBC) Mode

In this mode, a random initial vector (IV) of length n is first chosen. Then, the first ciphertext block is generated by applying the pseudorandom permutation to IV $\oplus$ m1 (i.e., the XOR of the first plaintext block and the IV). The remainder of the ciphertext is obtained by XORing the i^{th} ciphertext block with the i + 1 th plaintext block. Set c0 = IV. Then, for every i > 0 we set ci := $F_k(c_{i-1} \oplus mi)$. The final ciphertext is <IV, c1, . . . , cl>. IV is not kept secret and is sent in the clear as part of the ciphertext. This is crucial so that decryption can be carried out (without the IV , it will be impossible for the recipient to obtain the first plaintext block). Encryption in CBC mode is probabilistic. If F is a pseudorandom permutation, then CBC-mode encryption is CPA-secure. The main drawback of this mode is that encryption must be carried out sequentially because the ciphertext block ci is needed in order to encrypt the plaintext block (unlike decryption which may be executed in parallel). Thus, if parallel processing is available, CBC-mode encryption may not be the best choice.

Output Feedback (OFB) Mode

Essentially, this mode is a way of using a block cipher to generate a pseudo random stream that is then XORed with the message. First, a random IV $\leftarrow \{0, 1\}^n$ is chosen and a stream is generated from IV (independently of the plaintext message) in the following way. Define r_0 = IV , and set the i^{th} r_i block of the stream equal to $r_i := F_k(r_{i-1})$. Finally,

each block of the plaintext is encrypted by XORing it with the appropriate block of the stream; that is, $c_i := m_i \oplus r_i$. As in CBC mode, the IV is included in the clear as part of the ciphertext in order to enable decryption. This mode is also probabilistic. Here, both encryption and decryption must be carried out sequentially. On the other hand, this mode has the advantage that the bulk of the computation (namely, computation of the pseudorandom stream) can be done independently of the actual message to be encrypted. So, it may be possible to prepare a stream ahead of time using pre-processing, after which point the encryption of the plaintext (once it is known) is incredibly fast.

Counter (CTR) Mode

This mode of operation is less common than CBC mode, but has a number of advantages. There are different variants of counter mode; we describe the randomized counter mode. First, a random $IV \in \{0, 1\}^n$ is chosen; here, this IV is often denoted ctr. Then, a stream is generated by computing $r_i := (ctr + i)$. Finally, the i^{th} plaintext block is computed as $c_i := r_i \oplus m_i$. Counter mode has a number of important properties. First and foremost, randomized counter mode (i.e., when ctr is chosen uniformly at random each time a message is encrypted) is CPA-secure. Second, both encryption and decryption can be fully parallelized and, as with OFB mode, it is possible to generate the pseudo random stream ahead of time, independently of the message. Finally, it is possible to decrypt the i th block of the ciphertext without decrypting anything else; this property is called random access. The above makes counter mode a very attractive choice.