1 CPA- Security

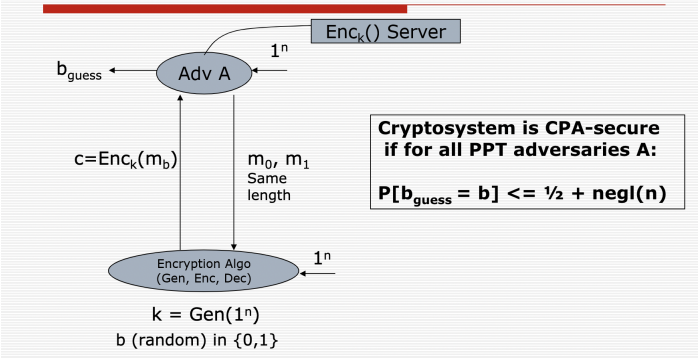

Figure 1: CPA Secure Scheme

1.1 Chosen Plaintext Attack

A chosen-plaintext attack (CPA) is an attack model where the attacker can obtain the ciphertexts for arbitrary plaintexts. The goal of the attack is to gain information that reduces the security of the encryption scheme. Since whatever security scheme we have designed is still for one-time mapping, (i.e if we encrypt same message again and again, same cipher text will be generated), to thwart a CPA attack, we need to move from the deterministic encryption to a probabilistic encryption method.

1.2 Moving from Deterministic to Probabilistic Encryption

If we want that everytime we encrypt a message, the cipher-text generated should be different, a deterministic encryption won't work. But if the cipher text keeps changing when we encrypt the same message, how will the decryption still be deterministic?
The basic idea is to not encrypt the message with probabilistic encryption but use a random noise everytime and encrypt that and xor it with message. Then send both the noise and the xor output i.e the cipher, for decryption. This way, encryption will be probabilistic but decryption will be deterministic.

$$c = (r, F_k(r) \oplus m)$$

1.3 Designing CPA-Secure Encryption scheme

Let F be a pseudorandom function. We define a private-key encryption scheme for messages of length n as follows:
- Gen: on input 1^n, choose $k \leftarrow \{0,1\}^n$ uniformly at random and output it as

the key.
- Enc: on input a key $k \in \{0,1\}^{\pi}$ and a message $m \in \{0,1\}^n$, choose $r + \{0,1\}^n$ uniformly at random and output the ciphertext

$$c := \langle r_1 F_k(r) \oplus m \rangle .$$

- Dec on input a key $k \in \{0,1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message

$$m := F_k(r) \oplus s.$$

1.4 Modes of Operations

1.4.1 CBC- Cipher Block Chaining

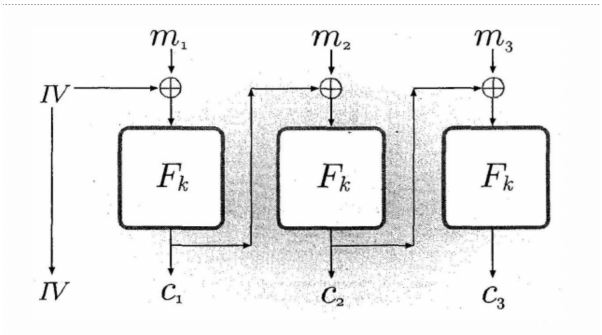

Figure 2: Cipher Block Chaining Mode

1.4.2 Output Feedback Mode

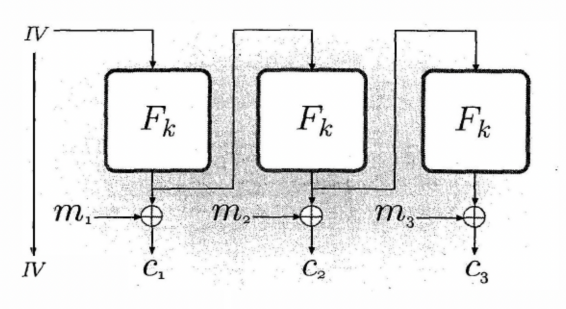

Figure 3: Output Feedback Mode

1.4.3 Randomized Counter Mode

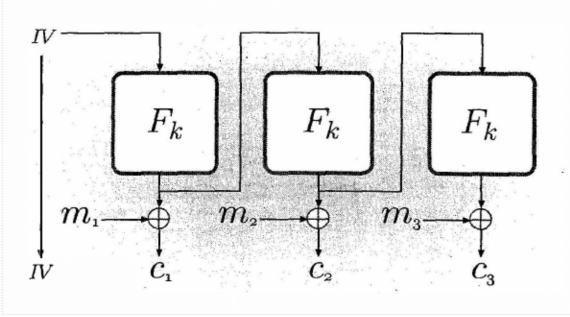

Figure 4: Randomized Counter Mode