CS181A Notes #8 Some Block Ciphers

Encryption modes Let $\mathcal{E} : \{0,1\}^n \to \{0,1\}^n$ be an n-bit encryption mapping (and in some cases let $\mathcal{D} : \{0,1\}^n \to \{0,1\}^n$ be the corresponding n-bit decryption mapping). If the message is N bits in length, then we need to partition the message into blocks of size n bits in order to apply the encryption mapping. Without loss of generality, we assume that N is divisible by n (using padding). So suppose $N = mn$ and let $x_1, \ldots, x_m$ be the plaintext n-bit blocks.

In what follows, we describe four well-known modes to send a sequence of n-bit blocks encrypted using $\mathcal{E}$.

1. *Electronic Code Book* (ECB)
   Here, we encrypt each plaintext block independently using $\mathcal{E}$.
   $$y_i = \mathcal{E}(x_i), \;\; i = 1, \ldots, m.$$
   Likewise, decryption is performed independently for each ciphertext block.

2. *Cipher Block Chaining* (CBC)
   We mix (using XOR) the ciphertext from the previous block with the current plaintext block to create the input to $\mathcal{E}$:
   $$\begin{aligned} y_1 &= \mathcal{E}(x_1) \\ y_{i+1} &= \mathcal{E}(x_i \oplus y_i), \;\; i = 0, \ldots, m-1. \end{aligned}$$
   Decryption proceeds symmetrically using $\mathcal{D}$.

3. *Cipher FeedBack* (CFB)
   Here, we use the encryption map $\mathcal{E}$ mainly to form the masking sequence. Let y_0 be a random string shared by Alice (sender) and Bob (receiver).
   $$y_i = x_i \oplus \mathcal{E}(y_{i-1}), \;\; i = 1, \ldots, m.$$
   Decryption only requires $\mathcal{E}$ and not $\mathcal{D}$.

4. *Output FeedBack* (CFB)
   Here, we use the encryption map $\mathcal{E}$ mainly to form the masking sequence. Let z_0 be a random string shared by Alice (sender) and Bob (receiver).
   $$\begin{aligned} z_i &= \mathcal{E}(z_{i-1}) \\ y_i &= x_i \oplus z_i, \;\; i = 1, \ldots, m. \end{aligned}$$
   Decryption only requires $\mathcal{E}$ and not $\mathcal{D}$.

DES The Data Encryption Standard (DES) is a *pseudo*-random 64-bit mapping that is determined by a 64-bit key. The design is based on repeated application of a *Feistel* cipher map on 64-bit string. Suppose the input is given by a 64-bit string (L, R). Then, the output of the Feistel cipher map is

$$(L', R') \doteq \mathcal{F}(L, R) = (R, L \oplus f(R)),$$

where $f : \{0,1\}^{32} \to \{0,1\}^{32}$ is a mapping chosen based on a given secret key. The permutation $\pi : \{0,1\}^{64} \to \{0,1\}^{64}$ used by DES is built from a composition of 16 Feistel permutations obtained using 16 different choices of f's. More specifically, if we let $\mathcal{F}_i(L, R) = (R, L \oplus f_i(R))$, then

$$\mathrm{DES}_K = \mathcal{F}_1 \circ \ldots \circ \mathcal{F}_{16},$$

where $f_1, \ldots, f_{16}$ is obtained from the 64-bit key K according to some scheduling procedure. For more information on the details of the f-box design, we recommend the article by Coppersmith [1] and by Landau [2]. An interesting work by Luby and Rackoff [3] showed a construction of a pseudorandom permutation using 3 rounds of Feistel mapping using different pseudorandom functions.

AES The Advanced Encryption Standard (AES) is a *pseudo*-random 128-bit mapping that is determined by a key that is either 128-bit (10-round), 192-bit (12-round), or 256-bit (14-round). In what follows, we summarize several interesting features of AES:

- The finite field $\mathbb{F}_{256}$ is ued to represent each ASCII byte (8-bit character).
- The 128-bit (16-byte) input is viewed as a 4×4 matrix over $\mathbb{F}_{256}$.
- The use of four types of transformations:
  1. Byte Substitution (BS):
     The input matrix $A = [a_{jk}]$ is mapped to $B = MA^{(-1)} + C$, where $A^{(-1)} = [a_{jk}^{-1}]$.
  2. Shift Row (SR):
     The i-th row of the input matrix A is circularly rotated i positions, for $i = 0, 1, 2, 3$.
  3. Mix Column (MC):
     The input matrix A is multiplied by a fixed matrix $\tilde{M}$. The goal is to achieve a diffusion among bytes (change in one input byte leads to 4 bytes changed in the output).

4. Add Round Key (ARK):
   The input matrix A is mapped to $B = A \oplus K$, where $\oplus$ is an entry-wise XOR operation and K is a key matrix.

- The use of a *key scheduling* algorithm to extract K for each round from the master 128-bit. We omit details of this algorithm.

References

[1] Don Coppersmith, "The Data Encryption Standard (DES) and its strengths against attacks," *IBM Journal of Research and Development* **38**(3):243-250, 1994.

[2] Susan Landau, "Standing the Test of Time: The Data Encryption Standard," *Notices of the AMS* **47**(3):341-349, 2000.

[3] M. Luby and C. Rackoff, "How to construct pseudo-random permutations from pseudo-random functions," *SIAM J. on Computing* **17**(2):373386, 1988.