Modes of Operation

In this lecture we discuss different possible ways in which block codes can be utilized to implement a cryptosystem. The possible block cipher *modes of operation* which we treat are identified by the acronyms ECB, CBC, CFB, and OFB. In each case we assume that we have a block cipher of block length n, with enciphering maps E_K and deciphering maps D_K for each key K.

Electronic Codebook Mode (ECB)

Electronic codebook mode is the most obvious way to use a block cipher.

Enciphering.

Input:
k-bit key K
n-bit plaintext blocks $M = M_1M_2 \dots M_t$.

Algorithm:

$$C_j = E_K(M_j).$$

Output:
n-bit ciphertext blocks $C = C_1C_2 \dots C_t$.

Deciphering.

Input:
k-bit key K
n-bit ciphertext blocks $C = C_1C_2 \dots C_t$.

Algorithm:

$$M_j = D_K(C_j).$$

Output:
n-bit plaintext blocks $M = M_1M_2 \dots M_t$.

To explain the name, one should think of this mode as being defined by a lookup table or *codebook*. Consider, for example, DES, which operates on 64 bit (binary) strings. These describe, for instance, 8 characters in 8-bit ASCII (or in 7-bit ASCII with one parity check bit). For each key K, the codebook contains the ciphertext image of each of these 8 character strings as a lookup table. In order to encipher the message, the electronic codebook is consulted for the ciphertext encoding of each block. Note that the number of such hypothetical codebooks is itself enormous – for DES there are 2^{56} possible keys, each with its own codebook.

We now consider some of the properties and limitations of ECB mode. The categories below are chosen for comparison with the modes of operations which follow.

Properties:

1. Identical plaintext. The same plaintext block always maps to the same ciphertext

block.
2. Chaining dependencies. Reordering the plaintext blocks induces a reordering of the same ciphertext blocks.
3. Error propagation. An error in a ciphertext block results in a deciphering error only in the corresponding plaintext block.

Security Remarks:
1. Malicious substitution of a ciphertext block C_j results in substitution of message block M_j.
2. Blocks C_j do not hide patterns – the same block M_j is enciphered in the same way.

Conclusion. Although commonly used, electronic codebook mode is not recommended for use if $t > 1$ with the same key. Security can be improved by inclusion of random padding bits in each block.

Cipher Block Chaining Mode (CBC)

Cipher block chaining mode involves a vector bit sum operation of the message block with the previous ciphertext block prior to enciphering. The ciphertext blocks are initialized with a randomly chosen message which may be transmitted openly, i.e. the security of the cryptosystem is based on the secrecy of the key, not on the secrecy of initialization vector.

Enciphering.

Input:
k-bit key K
n-bit initialization vector C_0
n-bit plaintext blocks $M = M_1M_2 \ldots M_t$.

Algorithm:

$$C_j = E_K(C_{j-1} \oplus M_j).$$

Output:
n-bit ciphertext blocks $C = C_0C_1 \ldots C_t$.

Deciphering.

Input:
k-bit key K
n-bit ciphertext blocks $C = C_0C_1 \ldots C_t$.

Algorithm:

$$M_j = C_{j-1} \oplus D_K(C_j).$$

Output:
n-bit plaintext blocks $M = M_1M_2 \ldots M_t$.

Properties:
1. Identical plaintext. The same sequence of ciphertext blocks result when the same key and the same initialization vector are used.

2. Chaining dependencies. The chaining mechanism causes C_j to depend on C_{j-1} and M_j, so enciphering is not independent of reordering.
3. Error propagation. An error in a ciphertext block C_j affects decipherment of C_j and C_{j+1}. For a reasonable enciphering algorithm, a single bit error affects 50% of the bits in the deciphered plaintext block M'_j, while the bit error affects only that bit of M'_{j+1}.
3. Error recovery. The cryptosystem is said to be self-recovering, in the sense that while an error in C_j results in incorrectly deciphered plaintext M'_j and M'_{j+1}, the ciphertext C_{j+2} correctly deciphers to $M'_{j+2} = M_{j+2}$.

Cipher Feedback Mode (CFB)

Cipher feedback mode allows one to process blocks of size $r < n$ at a time. The typical value for r is 1, while n may be of size 64, using DES.

Enciphering.

Input:
k-bit key K
n-bit initialization vector I_1
r-bit plaintext blocks $M = M_1M_2 \dots M_t$.

Algorithm:

$$\begin{aligned} &C_j = M_j \oplus L_r(E_K(I_j)), \\ &I_{j+1} = R_{n-r}(I_j) \,||\, C_j, \end{aligned}$$

where L_r and R_{n-r} are the operators which take the left-most r-bits and the right-most $n-r$-bits, and $||$ is the concatenation operator.

The vector I_j should be thought of as a *shift register*, a block of n-bits of memory which stores some state of the algorithm. The formation of I_{j+1} is a left-shift by r of this block, discarding the left-most r bits, with the right-most r bits replaced by C_j.

Deciphering.

Input:
k-bit key K
n-bit initialization vector I_1
r-bit ciphertext blocks $C = C_1C_2 \dots C_t$.

Algorithm:
Compute $I_1, \dots, I_t$ as in the enciphering algorithm, which can be generated independently of the deciphered message text, and then compute

$$M_j = C_j \oplus L_r(E_K(I_j)).$$

Note that CFB deciphering requires only the block cipher E_K, not the inverse block deciphering map D_K.

Properties:
1. Identical plaintext. The same sequence of ciphertext blocks results when the same

key and initialization vector is used. Changing the initialization vector changes the ciphertext.

2. Chaining dependencies. Ciphertext block C_j depends on the previous plaintext blocks $M_{j-1}, \ldots, M_1$ as well as M_j, so the ciphertext blocks are not reordering independent.

3. Error propagation. An error in C_j affects the decipherment of the next $[n/r]$ plaintext blocks. The recovered plaintext M'_j will differ from M_j at exactly the bits for which C_j was in error. These bit errors will appear in subsequent blocks M'_{j+k} at translated positions.

4. Error recovery. Proper deciphering requires the shift register to be correct, for which the previous $[n/r]$ ciphertext blocks are required. The decipherment is self-recovering from errors, but only after $[n/r]$ blocks (approximately the same n-bits of the ciphertext block in error).

5. Throughput. The rate of enciphering and deciphering is reduced by a factor of n/r, that is, for every r bits of output the algorithm must carry out one n-bit enciphering operation.

Output Feedback Mode (OFB)

Output feedback mode has a similar use as cipher feedback mode, but is relevant to applications for which error propagation must be avoided. Output feedback mode is an example of a *synchronous stream cipher* (constructed from a block cipher), in which the keystream is created independently of the plaintext stream.

Enciphering.

Input:
k-bit key K
n-bit initialization vector I_0
r-bit plaintext blocks $M = M_1 M_2 \ldots M_t$.

Algorithm:

$$\begin{aligned} I_j &= E_K(I_{j-1}) \\ C_j &= M_j \oplus L_r(I_j) \end{aligned}$$

Deciphering.

Input:
k-bit key K
n-bit initialization vector I_0
r-bit ciphertext blocks $C = C_1 C_2 \ldots C_t$.

Algorithm:
Compute $I_1, \ldots, I_t$ as in the enciphering algorithm.

$$M_j = C_j \oplus L_r(I_j)$$

Properties:

1. Identical plaintext. The same comments for CBC and CFB apply.

2. Chaining dependencies. The ciphertext output is order dependent, but the keystream $I_1, I_2, \ldots$ is plaintext independent.

3. Error propagation. An error in a ciphertext bit affects only that bit of the plaintext.

4. Error recovery. The cipher is self-synchronizing, and bit errors in a ciphertext block affect only that bit of the recovered plaintext. It recovers immediately from bit errors, but bit losses affect alignment.

5. Throughput. As with CFB, the rate of enciphering and deciphering is reduced by a factor of n/r, however the vectors I_j can be precomputed from K and I_0, independently of the ciphertext blocks.