Ciphers are classified as block or stream ciphers. All ciphers split long messages into blocks and encipher each block separately. Block sizes range from one bit to thousands of bits per block.

A block cipher enciphers each block with the same key.

A stream cipher has a sequence or stream of keys and enciphers each block with the next key. The key stream may be periodic, as in the Vigenère cipher or a linear feedback shift register, or not periodic, as in a one-time pad. Ciphertext is often formed in stream ciphers by exclusive-oring the plaintext with the key, as in the Vernam cipher.

Modes of operation of block ciphers:

Electronic Code Book
ECB: $c_i = E_k(m_i)$.

Cipher Block Chaining
CBC: $c_0 = IV$, $c_i = E_k(c_{i-1} \oplus m_i)$.

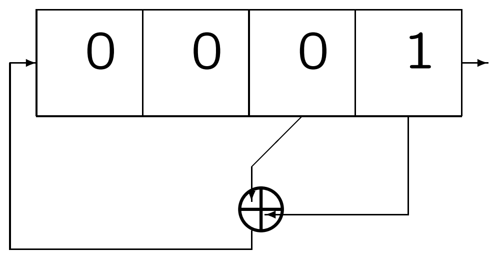
Output Feed Back
OFB: $r_0 = IV$, $r_i = E_k(r_{i-1})$, $c_i = m_i \oplus r_i$.

(n-bit) Counter
CTR: $ctr = IV$, $r_i = E_k((ctr + i) \bmod 2^n)$,
$c_i = m_i \oplus r_i$.

Cipher Feed Back
CFB: $c_0 = IV$, $c_i = E_k(c_{i-1}) \oplus m_i$.

Linear Feedback Shift Registers

A *linear feedback shift register* is a device which generates a key stream for a stream cipher. It consists of an n-bit shift register and an XOR gate. Let the shift register hold the vector $R = (r_0, r_1, \ldots, r_{n-1})$, numbered left to right. The inputs to the XOR gate are several bits selected (tapped) from fixed bit positions in the register. Let the 1 bits in the vector $T = (t_0, t_1, \ldots, t_{n-1})$ specify the tapped bit positions. Then the output of the XOR gate is the scalar product $TR = \sum_{i=0}^{n-1} t_i r_i$ mod 2, and this bit is shifted into the shift register.



Let $R' = (r'_0, r'_1, \ldots, r'_{n-1})$ be the contents of the register after the shift. Then $r'_i = r_{i-1}$ for $1 \le i < n$ and $r'_0 = TR$. In other words, $R' \equiv HR \bmod 2$, where H is the $n \times n$ matrix with T as its first row, 1's just below the main diagonal and 0's elsewhere.

$$H = \begin{pmatrix} t_0 & t_1 & \dots & t_{n-2} & t_{n-1} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}$$

The bit r_{n-1} is shifted out of the register and into the key stream. One can choose T so that the period of the bit stream is $2^n - 1$, which is the maximum possible period. If n is a few thousand, this length may make the cipher appear secure, but the linearity makes it easy to break.

Suppose $2n$ consecutive key bits $k_0, \ldots, k_{2n-1}$ are known. Let X and Y be the $n \times n$ matrices

$$X = \begin{pmatrix} k_{n-1} & k_n & \ldots & k_{2n-2} \\ k_{n-2} & k_{n-1} & \ldots & k_{2n-3} \\ \vdots & \vdots & \ddots & \vdots \\ k_0 & k_1 & \ldots & k_{n-1} \end{pmatrix}$$

$$Y = \begin{pmatrix} k_n & k_{n+1} & \ldots & k_{2n-1} \\ k_{n-1} & k_n & \ldots & k_{2n-2} \\ \vdots & \vdots & \ddots & \vdots \\ k_1 & k_2 & \ldots & k_n \end{pmatrix}$$

From $R' \equiv HR \bmod 2$ it follows that $Y \equiv HX \bmod 2$, so H may be computed from $H \equiv YX^{-1} \bmod 2$. The inverse matrix $X^{-1} \bmod 2$ is easy to compute by Gaussian elimination for n up to at least 10^4. The tap vector T is the first row of H and the initial contents R of the shift register are $(k_{n-1}, \ldots, k_0)$.

See Ding, Xiao and Shan [1991]for more information about linear feedback shift registers and variations of them.

Meet in the Middle Attacks

One might think that a way to make a block cipher more secure is to use it twice with different keys. If the key length is n bits, a brute force known plaintext attack on the basic block cipher would take up to 2^n encryptions. It might appear that the double encryption $C = E_{K_2}(E_{K_1}(M))$, where K_1 and K_2 are independent n-bit keys, would take up to 2^{2n} encryptions to find the two keys.

The meet-in-the-middle attack is a known plaintext attack which trades time for memory and breaks the double cipher using only about 2^{n+1} encryptions, and space to store 2^n blocks and keys. Let M_1 and M_2 be two known plaintexts, and let C_1 and C_2 be the corresponding ciphertexts. (A third pair may be needed.)

For each possible key K store $(K, E_K(M_1))$ in a file. Sort the 2^n pairs by second component. For each possible key K compute $D_K(C_1)$ and look for this value as the second component of a pair in the file.

If it is found, the current key might be K_2 and the key in the pair found in the file might be K_1.

Check whether $C_2 = E_{K_2}(E_{K_1}(M_2))$ to determine whether K_1 and K_2 really are the keys.

In the case of DES, $n = 56$ and the meet-in-the-middle attack requires enough memory to store 2^{56} plaintext-ciphertext pairs, more than is easily available now. But the attack is feasible with cloud computing.