BLOCK CIPHERS

MTH 440

Block ciphers

- Plaintext is divided into blocks of a given length and turned into output ciphertext blocks of the same length
- Suppose you had a block cipher, E(x,k) where the input plaintext blocks,x, were of size 5-bits and a 4-bit key, k.
- PT = 1010001010110010 5 (17 bits),
  "Pad" the PT so that its length is a multiple of 5 (we will just pad with 0's – it doesn't really matter)
- PT = 10100010101100101000
- Break the PT into blocks of 5-bits each ($x=x_1x_2x_3x_4$) where each x_i is 5 bits)
- x_1=10100, x_2= 01010, x_3=11001, x_4=01000
- Ciphertext: $c_1c_2c_3c_4$ where
- $c_1=E(x_1,k_1)$, $c_2=E(x_2,k_2)$, $c_3=E(x_3,k_3)$, $c_4=E(x_4,k_4)$
- (when I write the blocks next to each other I just mean concatentate them (not multiply) – we'll do this instead of using the || notation when it is not confusing)
- Note the keys might all be the same or all different

What do the E's look like?

- If y = E(x,k) then we'll assume that we can decipher to a unique output so there is some function, we'll call it D, so that x = D(y,k)
- We might define our cipher to be repeated applications of some function E either with the same or different keys, we call each of these applications "round"
- For example we might have a "3 round" cipher:

$$y = F_k(x) = E(E(E(x,k_1),k_2),k_3)$$

- We would then decipher via

$$x = F_k^{-1}(y) = D(D(D(y,k_3),k_2),k_1)$$

S-boxes (Substitution boxes)

- Sometimes the “functions” used in the ciphers are just defined by a look up table that are often referred to “S-boxes”

$x_1x_2x_3$	$S(x_1x_2x_3)$
000	11
001	01
010	00
011	10
100	01
101	00
110	11
111	10

Define a 4-bit function with a 3-bit key as follows:

$$t_1t_2 = S(x_3x_4x_3 \oplus k_1k_2k_3)$$

$$u_1u_2 = x_1x_2 \oplus t_1t_2$$

$$E(x_1x_2x_3x_4, k_1k_2k_3) = x_3x_4u_1u_2$$

$E(0110,100)$: $t_1t_2 = S(101 \oplus 100) = S(001) = 01$

$u_1u_2 = 01 \oplus 01 = 00$

$E(0110,100) = 1000$

Try it: E(1100,011) (on your worksheet)

Deciphering

- The function E can be inverted as follows:

$x_1x_2x_3$	$S(x_1x_2x_3)$
000	11
001	01
010	00
011	10
100	01
101	00
110	11
111	10

$$t'_1 t'_2 = S(y_1 y_2 y_1 \oplus k_1 k_2 k_3)$$

$$u'_1 u'_2 = y_3 y_4 \oplus t'_1 t'_2$$

$$D(y_1 y_2 y_3 y_4, k_1 k_2 k_3) = u'_1 u'_2 \, y_1 y_2$$

Check this works by deciphering the one on your worksheet

The cipher

- Clearly the function E alone is not a great cipher since two bits of the plaintext appear in the ciphertext
- Instead we might define the cipher as

$$y = F_k(x) = E(E(E(x,k),k),k)$$

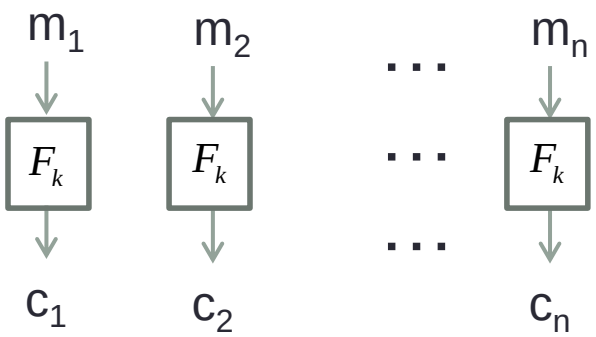
- Here the cipher is formed by 3 applications or “rounds” of E. To decipher we would then get:

$$x = F_k^{-1}(y) = D(D(D(y,k),k),k)$$

- We don’t have to do 3 rounds, we could do as many as we like.
- This might be tedious by hand, but the operations of looking up values in a table and performing addition modulo 2 are very easy on a computer

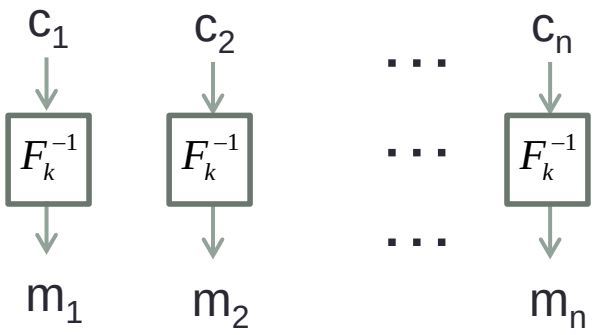
Modes of operation - ECB

- Denote our plaintext by $m = m_1 m_2 ... m_n$
- The key by k and the ciphertext by $c = c_1 c_2 ... c_n$
- Each m_i, c_i are t-bit blocks where t is the length of the block cipher F_k



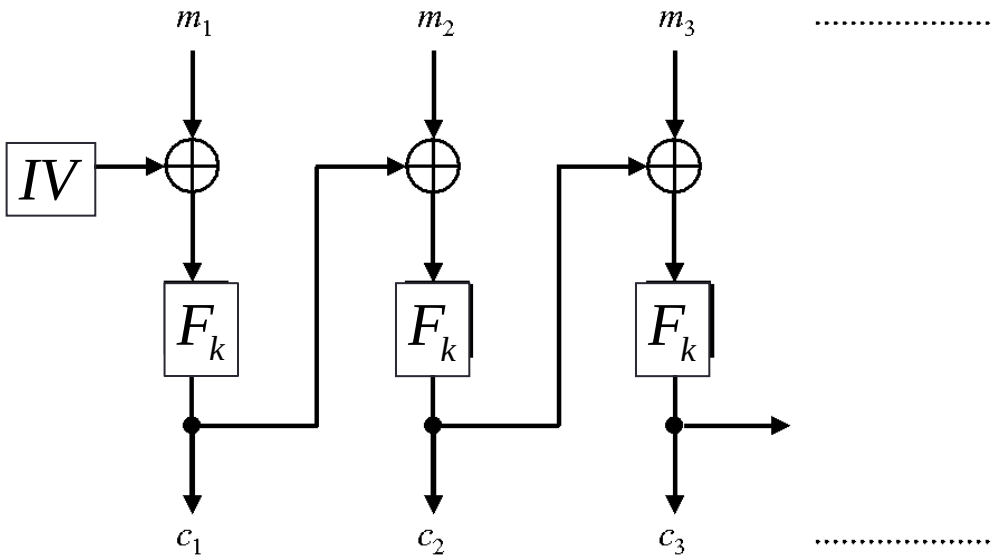
ECB: Electronic Codebook Mode - encipher block by block separately:

$$c_i = F_k(m_i)$$



To decipher in ECB mode just decipher block by block: $m_i = F_k^{-1}(c_i)$

Modes of operation: CBC



CBC: cipher block chaining (most common mode of operation) the output of one block is used in the input to the next block

$IV =$ initialization vector (need not be secret, need not be used)

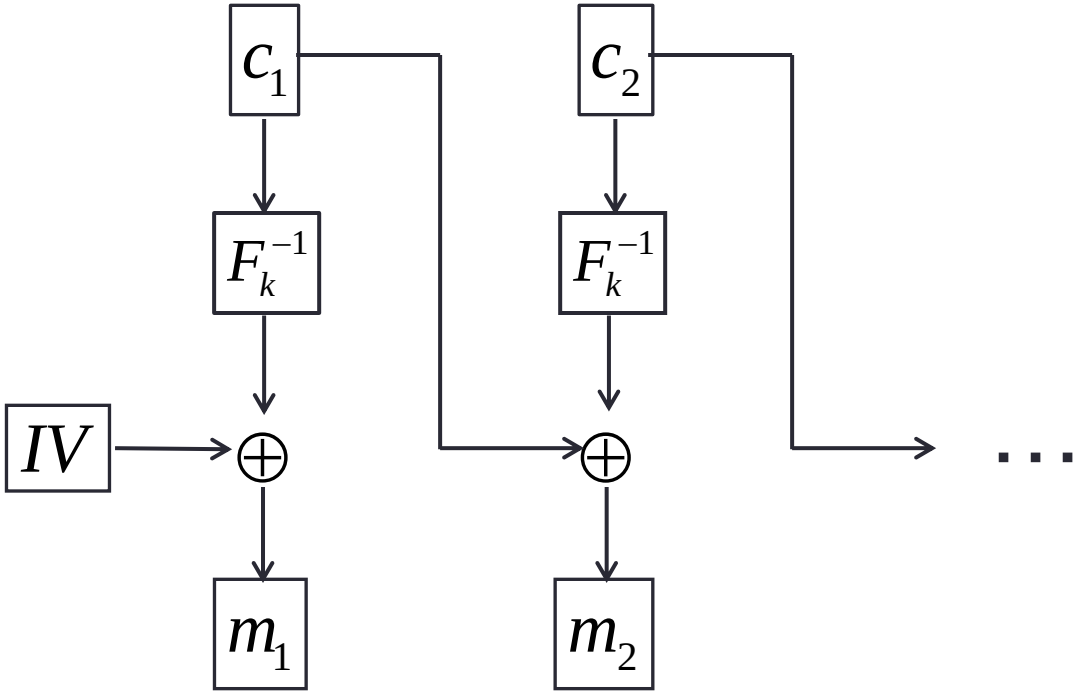
$$c_1 = F_k(IV \oplus m_1)$$

$$c_i = F_k(c_{i-1} \oplus m_i), \quad i = 2,3,\ldots$$

How would you decipher something in CBC mode?

- Work it out

Deciphering in CBC



$$m_1 = F_k^{-1}(c_1) \oplus IV = F_k^{-1}(F_k(IV \oplus m_1) \oplus IV = IV \oplus m_1 \oplus IV = m_1$$

$$m_i = F_k^{-1}(c_i) \oplus c_{i-1}, \quad i = 2,3,...$$

Try it

- Define F to be a 4-bit block cipher with a 2-bit key defined as

$$F_k(m) = m_1 m_2 m_3 m_4 \oplus k_1 k_2 k_2 k_1$$

- Clearly

$$F_k^{-1}(m) = c_1 c_2 c_3 c_4 \oplus k_1 k_2 k_2 k_1$$

- Given and IV = 1011 and k=1001, encipher m=110101010100 in **CBC mode** then decipher to check work (4&5 on handout)

- **WE ENDED CLASS HERE ON FRIDAY**

Other modes of operation

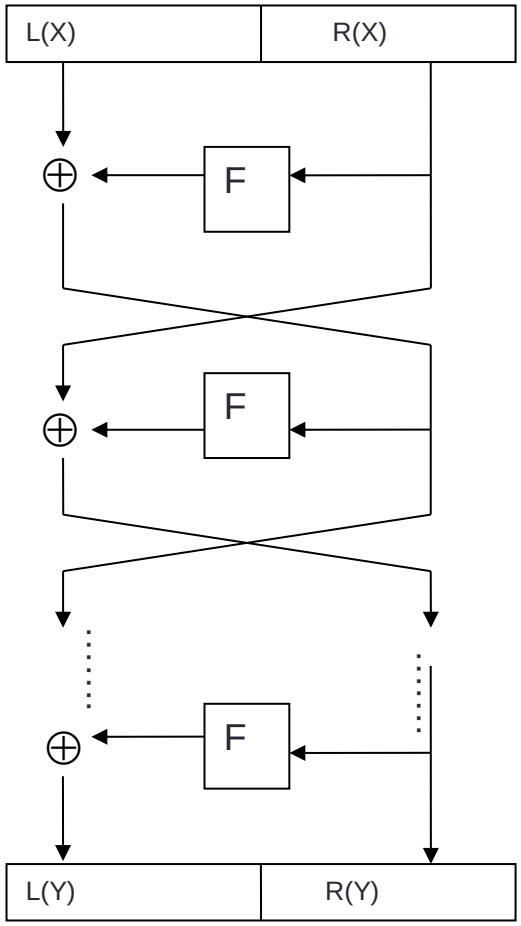
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)
- Both of these modes allow a block cipher to be used more like a stream cipher

The Feistel Structure

- Named for Horst Feistel who worked at IBM and invented the Data Encryption Standard (*DES*) which was a national standard in use from 1977 until it was replaced in 2000 by the Advanced Encryption Standard (AES)

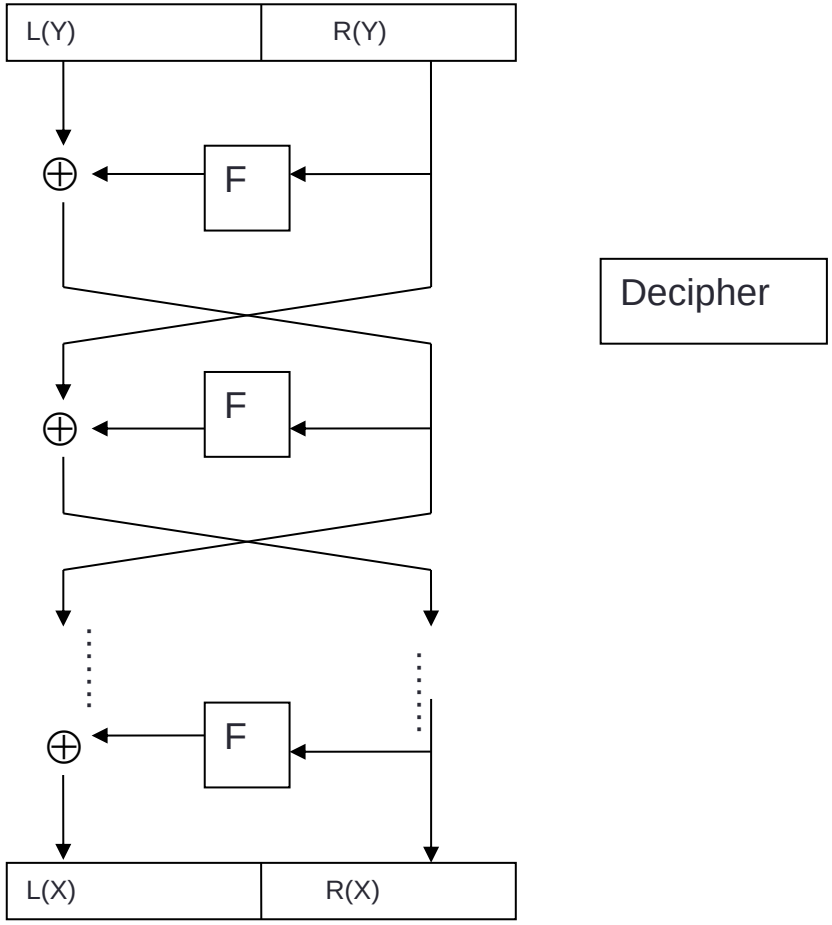
Feistel Ciphers

- We define a round function, F, which takes an n-bit input and produces an n-bit output according to some rule.
- Round functions are often defined by S-boxes
- The Feistel cipher is a **2n**-bit cipher and incorporates F as a "round function". The cipher can have any number of rounds.
- Take a 2n bit block $x=x_1x_2..x_n\ x_{n+1}\ldots x_{2n}$ and break it into a left half and a right half $x = L(x)||R(x)$
  - $L(x) = x_1x_2..x_n$
  - $R(x)= x_{n+1}\ldots x_{2n}$



Encipher

To decipher either reverse all the arrows or just put the L(Y) and R(Y) in the top to get the L(X) an R(X) out at the bottom as illustrated

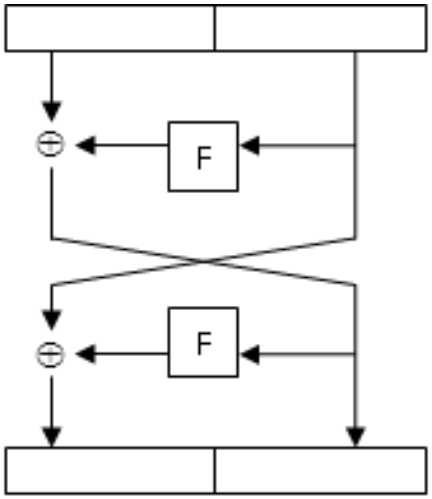


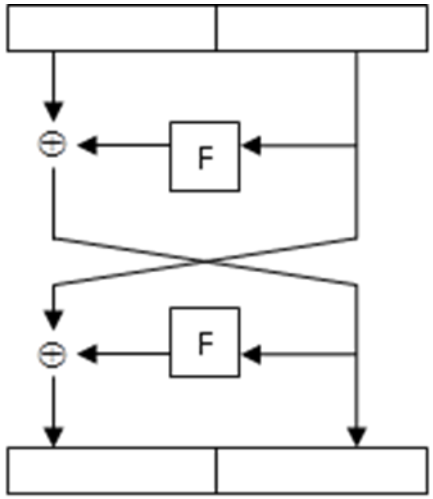
Example:

S-box for round function

$x_1x_2x_3$	$F(x_1x_2x_3)$
000	110
001	011
010	110
011	100
100	111
101	000
110	001
111	011

With a 3-bit round function we can create a 6-bit Feister Cipher. For this example we will do two rounds. **Encipher** 101110010111 in **ECB Mode**





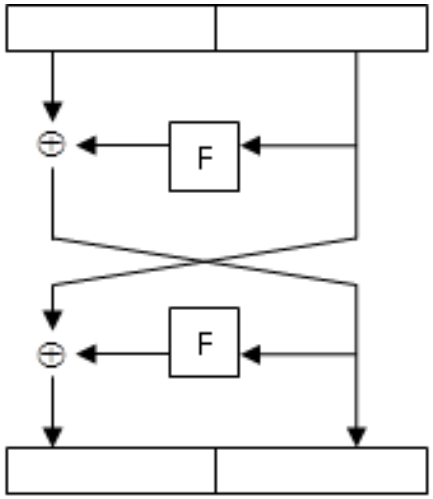
Example:

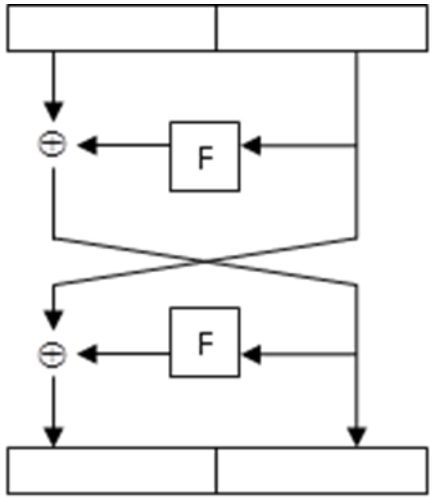
S-box for round function

$x_1x_2x_3$	$F(x_1x_2x_3)$
000	110
001	011
010	110
011	100
100	111
101	000
110	001
111	011

Notice F is not invertible but we can still decipher

With a 3-bit round function we can create a 6-bit Feister Cipher. For this example we will do two rounds. **Decipher your result**



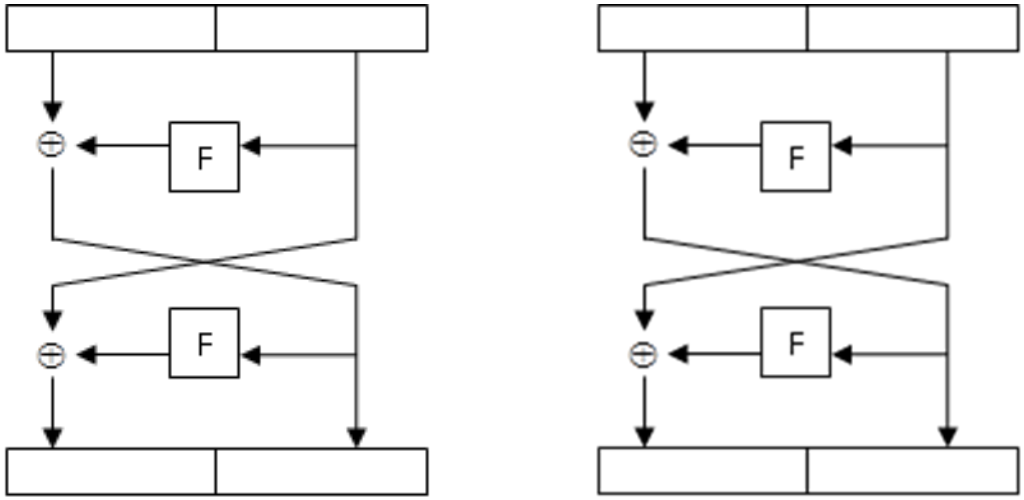


Example:

S-box for round function

$x_1x_2x_3$	$F(x_1x_2x_3)$
000	110
001	011
010	110
011	100
100	111
101	000
110	001
111	011

With a 3-bit round function we can create a 6-bit Feister Cipher. For this example we will do two rounds. **Encipher** 101110010111 in using initialization vector IV = 110 010 **CBC Mode**

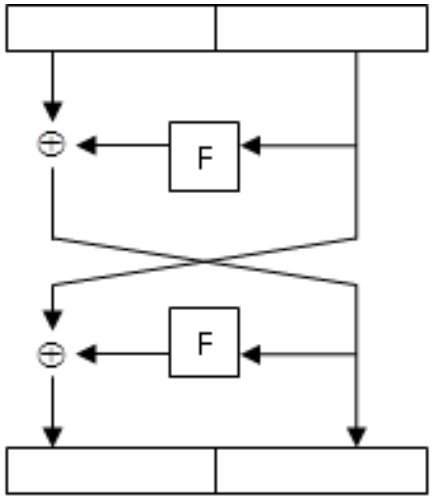


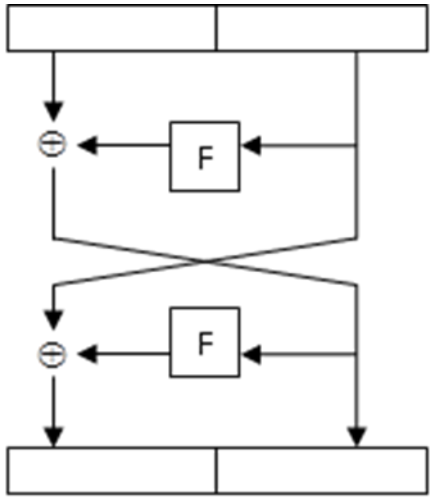
Example:

S-box for round function

$x_1x_2x_3$	$F(x_1x_2x_3)$
000	110
001	011
010	110
011	100
100	111
101	000
110	001
111	011

With a 3-bit round function we can create a 6-bit Feister Cipher. For this example we will do two rounds. **Decipher your result (**IV = 110 010)





Types of attacks on ciphers

- An attack on a cipher does not mean that a key was found or the cipher is completely broken. It means that something was done that an “authorized” person should be able to do (aka someone with a key)
- As a simple example, no one who is not in possession of the key should be able to encipher/decipher a plaintext/ciphertext

Types of attacks on ciphers

- When considering the security of a cipher we generally make one or more assumptions such as: An attacker:
  - Knows what cipher is in use
  - Has access to A LOT of plaintext/ciphertext pairs **(plaintext/ciphertext attack)**
  - An attacker can request and get access to a list of ciphertext/plaintext corresponding to the plaintext/ciphertext according to their choosing **(chosen plaintext/ciphertext attack)**
  - An attacker can make multiple request for plaintext/ciphertext pairs – this means they can make their choices for their plaintext/ciphertext requests based on analysis of previously acquired results **(adaptive chosen plaintext/ciphertext attack)**
  - The attacker gets access to these plaintext/ciphertext pairs by asking an “orcale” (meaning a black box/ software / or person who is able to get them such information)