



CIS 5371 Cryptography

7. Symmetric encryption



Cryptographic systems

Cryptosystem: (M, C, K, K', G, E, D)

- M , plaintext message space
- C , ciphertext message space
- K, K' , encryption and decryption key spaces
- $G : N \rightarrow K \times K'$, key generation algorithm
- $E : M \times K \rightarrow C$, encryption algorithm
- $D : C \times K' \rightarrow M$, decryption algorithm

G, E, D must be efficient

Examples

- Cryptosystem: (M, C, K, K', G, E, D)
- Substitution Cipher: $M = C = \mathbf{Z}_{26}$, with $K=K'$
- The encryption algorithm is a mapping $E_k: M \rightarrow C$
-- $E_k(x) = \pi(x)$, where $k \in K$ is the key.
- The decryption algorithm is a mapping $D_k: M \rightarrow C$
-- $D_k(y) = \pi^{-1}(y)$.

- Shift Cipher: $M = C = K = K' = \mathbf{Z}_{26}$, with
-- $E_k(x) = x + k \pmod{26}$
-- $D_k(y) = y - k \pmod{26}$
where $x, y \in \mathbf{Z}_{26}$

Examples

Polyalphabetic ciphers: a plaintext message can be encrypted into any ciphertext

Vigenere cipher --

a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Key m i k e → 12 8 10 4

Plaintext c r y p t o g r a p h y → 2 17 24 15 19 14 6 17 0 15 7 24

<i>p l a i n t e x t</i>	2	17	24	15	19	14	6	17	0	15	7	24
<i>k e y</i>	12	8	10	4	12	8	10	4	12	8	10	4
<i>c i p h e r t e x t</i>	14	25	8	19	5	22	16	21	12	23	17	2

Ciphertext o z i t f w q v m x r c



Vernam cipher - the one-time pad

$$\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathcal{K}' = \{0, 1\}^n, n > 1.$$

The keys $k = k_1, k_2, \dots, k_n$ are selected at random in \mathcal{K} with uniform distribution.

Encryption is bit by bit at a time, with each ciphertext bit obtained by XORing each message bit with the corresponding key bit.

Decryption is the same as encryption since the XOR operation is its own inverse.

The special case when $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathcal{K}' = \{0, 1\}^*$, and the key is only used once (one-time key) gives us a cipher with a strong security property: **perfect secrecy**.



Transposition (permutation) cipher

Example

$M = C = (\mathbb{Z}_{26})^m$, $m > 1$, $K=K'$ is the set of all permutations of $\{1, \dots, m\}$.

- For a key (permutation) π
- -- $e_\pi(x_1, \dots, x_m) = (x_{\pi(1)}, \dots, x_{\pi(m)})$
- -- $d_\pi(y_1, \dots, y_m) = (y_{\pi^{-1}(1)}, \dots, y_{\pi^{-1}(m)})$
where $\pi^{-1}(1)$ is the inverse of π .



Structure of classical ciphers

Classical ciphers are based on:

- Substitution and
- Transposition.

This is also the basis for modern ciphers



Cryptanalysis: attacks on cryptosystems

- **Ciphertext only** attacks: the opponent possesses a string of ciphertexts:

Y_1, Y_2, \dots

- **Known plaintext** attack: the opponent possesses a string of plaintexts

X_1, X_2, \dots

and the corresponding string of ciphertexts:

Y_1, Y_2, \dots



Usefulness of classical ciphers

- Cryptanalysis of substitution ciphers:
Known plaintext attack -- easy to get the keys
Ciphertext only attack -- use statistical properties of the language.
- Cryptanalysis of polyalphabetic (Vigenere) cipher:
Known plaintext attack -- easy to get the keys
Ciphertext only attack -- use statistical properties of the language.




Requirements for secure use of classical ciphers

The notion of information-theoretic cryptographic security was developed by Shannon and requires that:

- $|K| \geq |M|$
- $k \in_{\mathcal{U}} K$ and is used only once in each encryption

This kind of security is not practical for most applications.



The one-time-pad -- Perfect secrecy

Assume that there is a **distribution** on P, K .

Then the plaintext and the keys are chosen with a certain probability.

That is we have:

$$\Pr[\mathbf{x} = x] \text{ and } \Pr[\mathbf{k} = k],$$

where x, \mathbf{k} are **random variables** (r.v.'s).



The one-time-pad -- Perfect secrecy

The probability distribution on P and K induces a distribution on C , for which:

$$\Pr [y = y] = \sum_{k: y=e_k(x), x \in P} \Pr [\mathbf{k} = k] \Pr [x = d_k(y)]$$

For this distribution we have,

$$\Pr [x, y] = \Pr [x = x] \times \sum_{k: x=d_k(y)} \Pr [\mathbf{k} = k]$$

Perfect secrecy

Using Bayes' theorem we get:

$$\Pr [\mathbf{x} = x \mid \mathbf{y} = y] = \frac{\Pr [\mathbf{x} = x] \times \sum_{K: x=d_K(y)} \Pr [\mathbf{k} = k]}{\sum_{k: y=e_k(x), x \in \mathcal{P}} \Pr [\mathbf{k} = k] \Pr [\mathbf{x} = d_k(y)]}$$



Perfect secrecy

Definition:

We have perfect secrecy if:

$$\Pr[\mathbf{x} = x / \mathbf{y} = y] = \Pr[\mathbf{x} = x] ,$$

for all $x \in \mathcal{P}$, $y \in \mathcal{C}$.



Perfect secrecy

Theorem

The One-Time-Pad provides perfect secrecy.

Proof

We have:

$\Pr [\mathbf{k} = k] = 1 / |\mathbf{K}|$, and

for each $x \in \mathcal{P}$, $y \in \mathcal{C}$ there is exactly one key k with $y = e_k(x)$.



Perfect secrecy

Proof (continued)

Then

$$\begin{aligned}\Pr [\mathbf{y} = y] &= \sum_{k: y=e_k(x), x \in \mathcal{P}} \Pr [\mathbf{k} = k] \cdot \Pr [\mathbf{x} = d_k(y)] \\ &= 1 / |\mathbf{K}| \cdot \sum_{k: y=e_k(x), x \in \mathcal{P}} \Pr [\mathbf{x} = d_k(y)] \\ &= 1/|\mathbf{K}| .\end{aligned}$$



Perfect secrecy

Proof (continued)

Using Bayes' theorem:

$$\begin{aligned}\Pr [\mathbf{x}=\mathbf{x}/\mathbf{y}=\mathbf{y}] &= \Pr [\mathbf{y}=\mathbf{y}/\mathbf{x}=\mathbf{x}] \times \Pr [\mathbf{x}=\mathbf{x}] / \Pr [\mathbf{y}=\mathbf{y}] \\ &= \Pr [\mathbf{k}=\mathbf{k}] \Pr [\mathbf{x}=\mathbf{x}] / \Pr [\mathbf{y}=\mathbf{y}]\end{aligned}$$

We have just seen that: $\Pr [\mathbf{k}=\mathbf{k}] = \Pr[\mathbf{y}=\mathbf{y}]$.

It follows that:

$$\Pr [\mathbf{x}=\mathbf{x}/\mathbf{y}=\mathbf{y}] = \Pr [\mathbf{x}=\mathbf{x}] ,$$

so we have **perfect secrecy**.



Iterating Block ciphers

1. Key schedule

(Binary) key $k \rightarrow$ round keys: k^1, \dots, k^{N_r} ,

2. Round function g

$$w^r = g(w^{r-1}, k^r),$$

where w^{r-1} is the previous state



Iterated cipher ...

Encryption operation:

$$w^0 \leftarrow x$$

$$w^1 = g(w^0, k^1),$$

$$w^2 = g(w^1, k^2),$$

$$w^{Nr} = g(w^{Nr-1}, k^{Nr}),$$

$$y \leftarrow w^{Nr}$$



Iterated cipher ...

For **decryption** we must have:

$g(.,k)$ must be invertible for all k

Then decryption is the reverse of encryption
(bottom-up)



DES

DES is a special type of iterated cipher called a **Feistel cipher**.

Block length 64 bits

Key length 56 bits

Ciphertext length 64 bits



DES

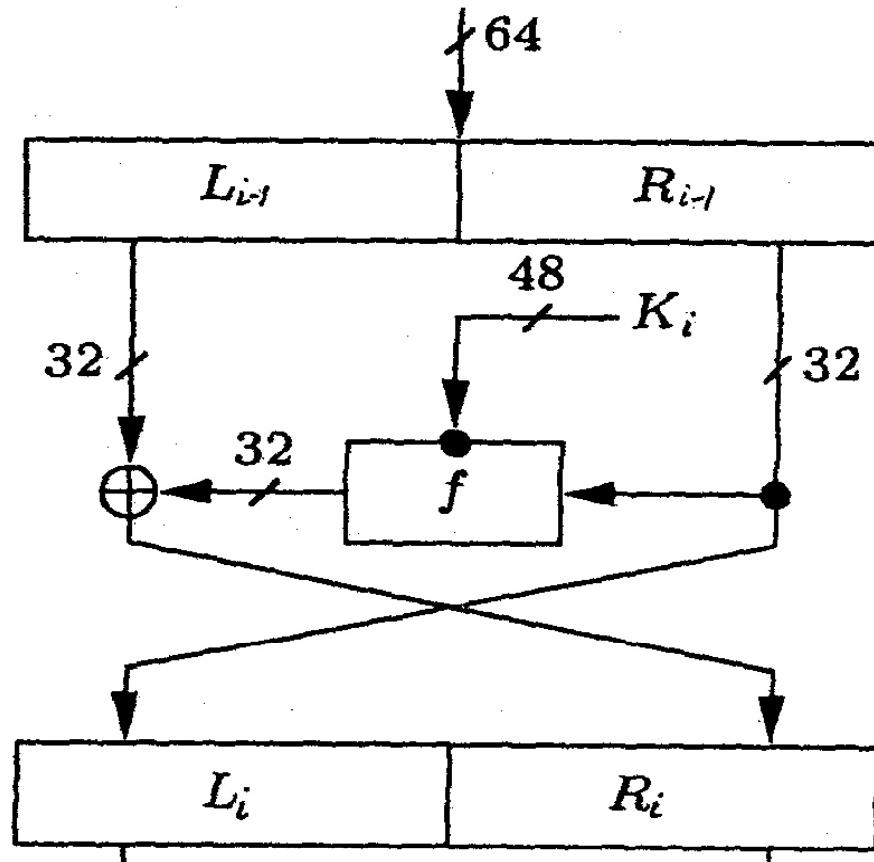
The round function is:

$$g([L_{i-1}, R_{i-1}], K^i) = (L_i, R_i),$$

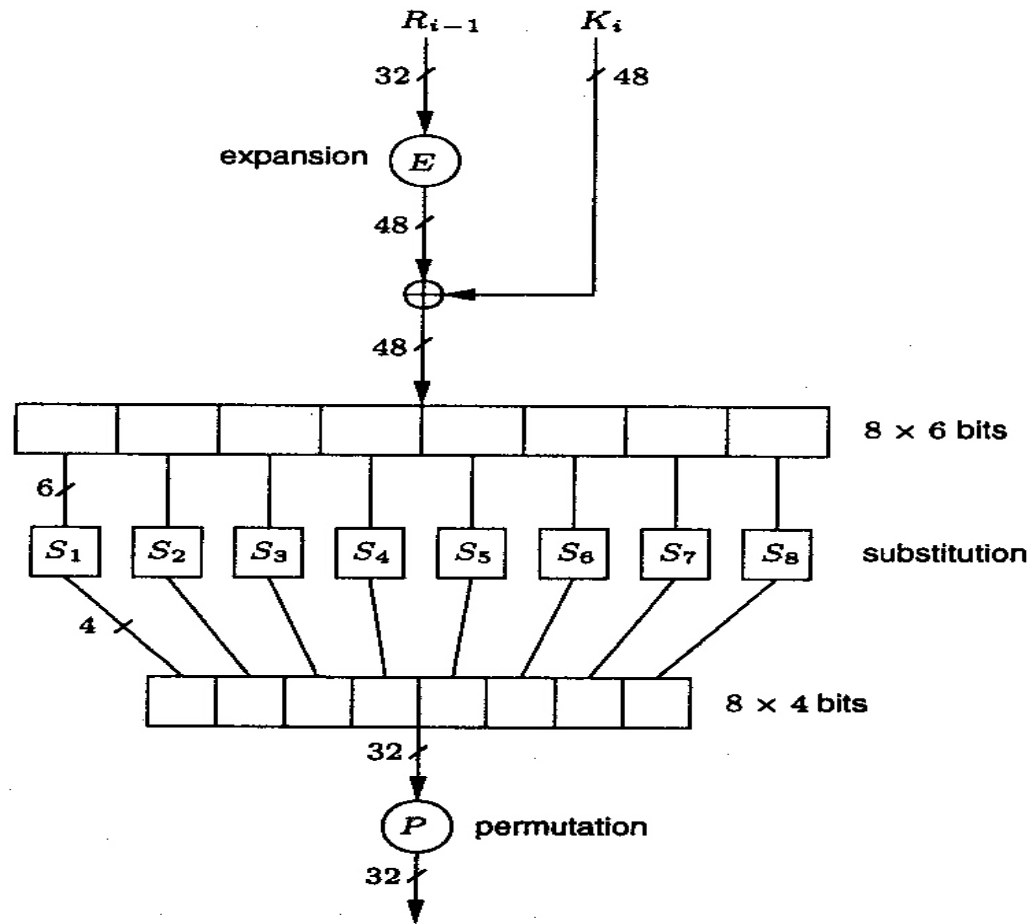
where

$$L_i = R_{i-1} \quad \text{and} \quad R_i = L_{i-1} \text{ XOR } f(R_{i-1}, K_i).$$

DES round encryption



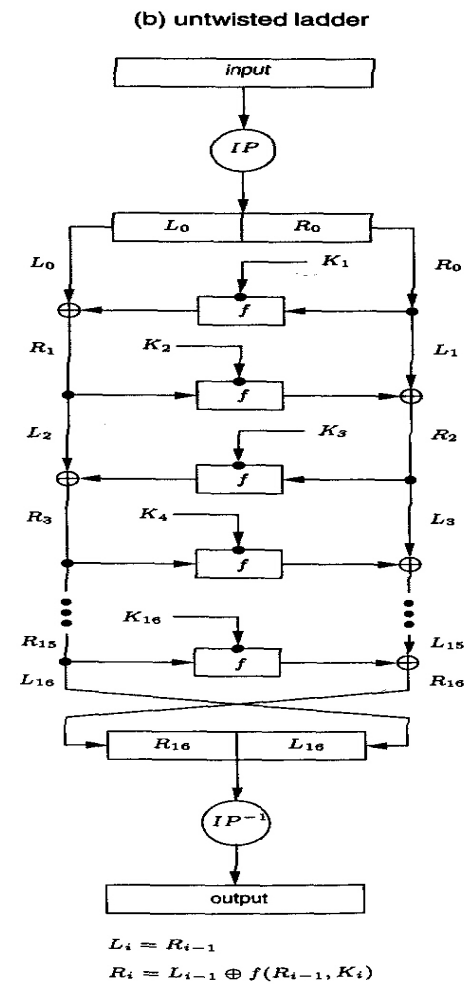
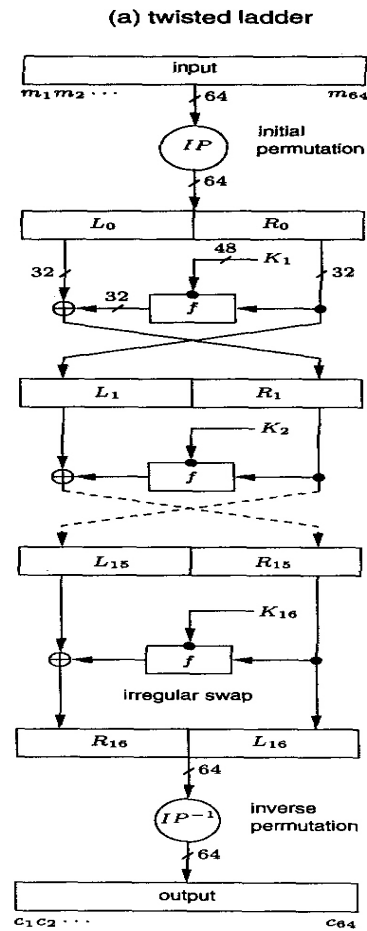
DES inner function



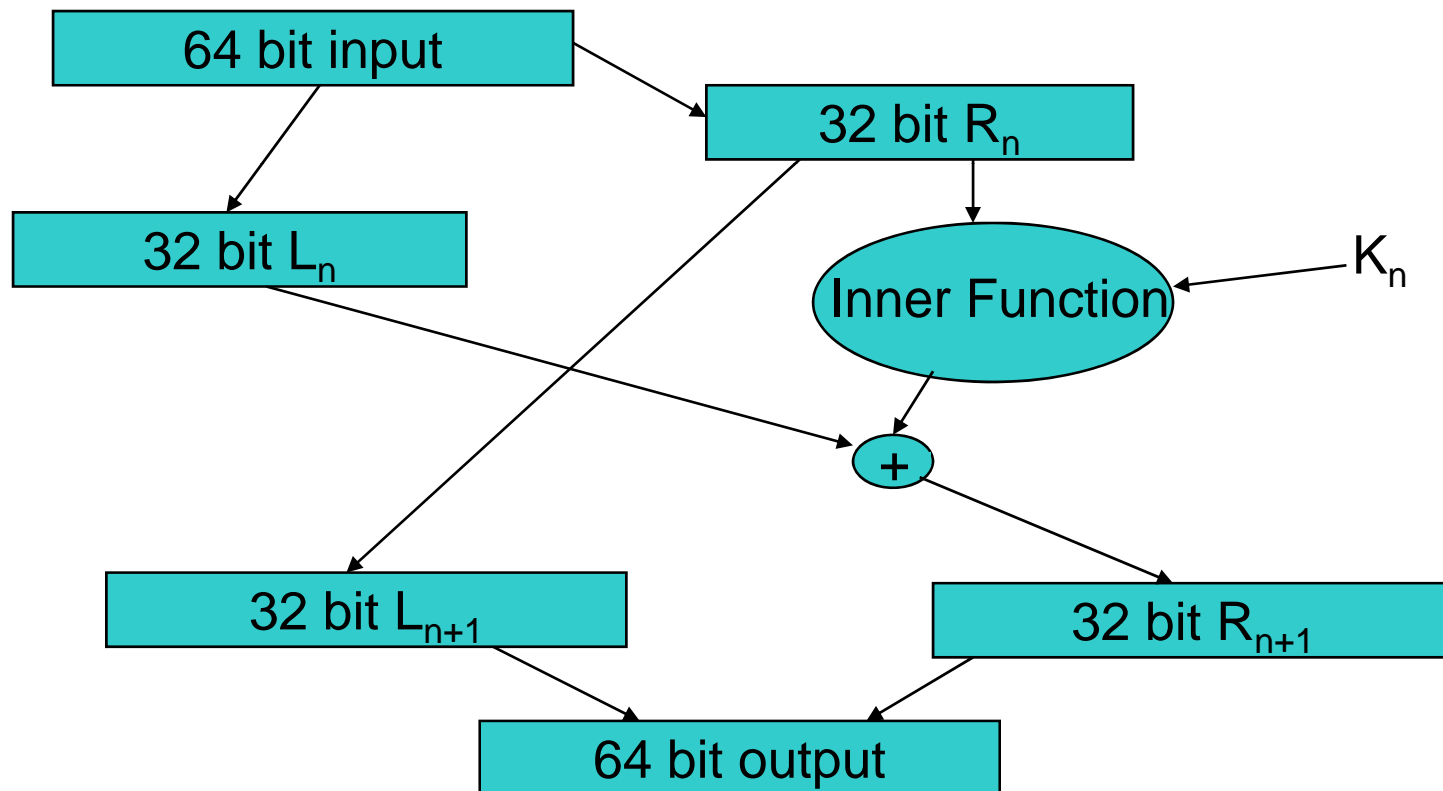
$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

symmetric cryptography

DES computation path



One DES Round





Inner Function

Combine 32 bit input and 48 bit key
into 32 bit output

- Expand 32 bit input to 48 bits
- XOR the 48 bit key with the expanded 48 bit input
- Apply the S-boxes to the 48 bit input to produce 32 bit output
- Permute the resulting 32 bits

Inner Function

Expand 32 bit input to 48 bits by adding a bit to the front and the end of each 4 bit segment.

These bits are taken from adjacent bits.

This String

	1	2	3	4			5	6	7	8				25	26	27	28			29	30	31	32	
32	1	2	3	4	5	4	5	6	7	8	9	...	24	25	26	27	28	29	28	29	30	31	32	1

Notice several bit values are repeated: 4, 5, 8, 9, 28, 29, etc.

Becomes this String



S Boxes

- There are 8 different S-Boxes, 1 for each chunk
- S-box process maps 6 bit input to 4 bit output
- S box performs substitution on 4 bits
- There are 8 possible substitutions in each S box
- Inner 4 bits are fed into an S box
- Outer 2 bits determine which substitution is used

S Boxes

Use bits 1 & 6 to select the row

Bits 2-5 to select the substitution

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101



DES: The Initial and Final Permutations

There is also an initial and a final permutation: the final permutation is the inverse of the initial Permutation.



Decrypting DES

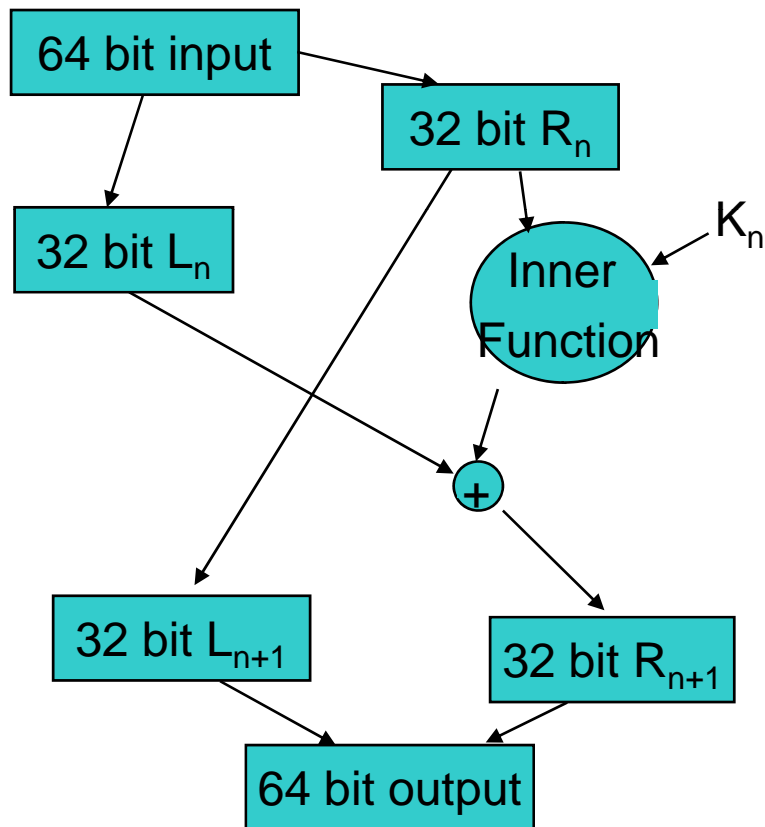
DES (and all Feistel structures) is invertible through “reverse” encryption because

- The input to the n^{th} step is the output of the $n-1^{\text{th}}$ step
- Everything needed (except the key) to produce the input to the inner function of the $n-1^{\text{th}}$ step is available from the output of the n^{th} step.

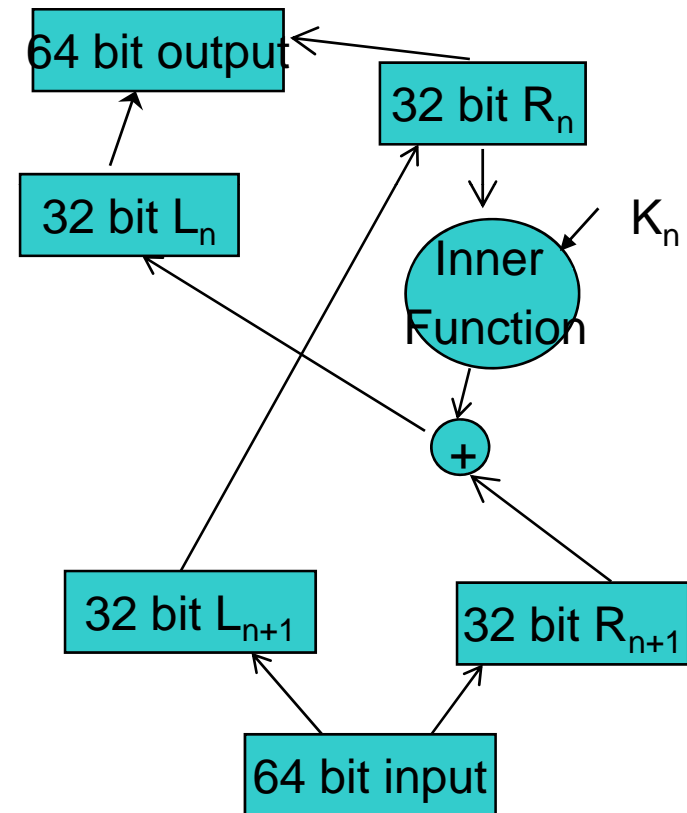
So we can Work backwards to step 1.

Note that the S-boxes are not reversible (and don't need to be)

Encrypt round n



Decrypt round $n+1$





Key schedule

INPUT: 64-bit key: k_1, k_2, \dots, k_{64}

OUTPUT: sixteen 48-bit keys: k_1, k_2, \dots, k_{16}

The algorithm used for generating the key schedule combines and selects bits of K to generate the round keys two bit selection tables.

-- for details see [Handbook of Applied Cryptography](#).



Weak Keys

Let C_0 and D_0 are the 28 bit key halves

- There are 4 weak keys in the keyspace (2^{56})
 - $C_0 = \text{All zeros} \ \& \ D_0 = \text{All zeros}$
 - $C_0 = \text{All ones} \ \& \ D_0 = \text{All zeros}$
 - $C_0 = \text{All zeros} \ \& \ D_0 = \text{All ones}$
 - $C_0 = \text{All ones} \ \& \ D_0 = \text{All ones}$
- There are 12 semi-weak keys, where C_0 & D_0 are the following in some combination
 - All zeros, All ones, 010101..., 101010...

-- for details see [Handbook of Applied Cryptography](#).



Attacks on iterated ciphers

Suppose that there a probabilistic **linear** relation between some plaintext bits and state bits immediately preceding the last round.

Say the bits XOR to 0 with probability bounded away from $\frac{1}{2}$.

Linear cryptanalysis is a known plaintext attack.

The attacker needs to know a large number of pairs (x_i, y_i) encrypted with the same key K , and uses a **linear relation** to decrypt a given cipher



Kerchoffs' assumption

The adversary knows all details of
the encrypting function
except the secret key



Diffusion and Confusion -- Shannon

- **Diffusion.** The relationship between the statistics of the plaintext and the ciphertext is as complex as possible: the value of each plaintext bit affects many plaintext bits.
- **Confusion:** the relationship between the statistics of the ciphertext and the value of the key is as complex as possible.



Attacks on DES

- Brute force
- Linear Cryptanalysis
 - Known plaintext attack
- Differential cryptanalysis
 - Chosen plaintext attack
 - Modify plaintext bits, observe change in ciphertext

No dramatic improvement on brute force



Linear cryptanalysis (known plaintext)

For each pair (x_i, y_i) , decrypt using all possible candidate keys for the last round and determine if the linear relation holds.

If it does, increment a frequency counter for the candidate key used.

Hopefully, at the end, this counter can be used to determine **the correct values for the subkey bits.**



Differential cryptanalysis (chosen plaintext)

Differential cryptanalysis is a **chosen plaintext attack**.

In this case the XOR of two inputs x, x^* is compared with that of the corresponding outputs y, y^* .

In general we look for pairs x, x^* for which $x' = x + x^*$ **is fixed**.

For each such pair, decrypt y, y^* using all possible candidate keys for the last round, and determine if their XOR has a certain value.

Again use a frequency counter.

Hopefully, at the end, this counter can be used to determine **the correct values for the subkey bits**.



The security of DES

None of these attacks have a serious impact on the security of DES.

The main problem with DES is that it has relatively short key length. Consequently it is subject to **brute-force** or exhaustive key search attacks.

One solution to overcome this problem is to run DES a multiple number of times.



Countering Attacks

- Large keyspaces combats brute force attack
- Triple DES, typically two key mode: $E_{k_1} D_{k_2} E_{k_1}$
- Use AES



Triple DES

Encryption:

$$c = E_{k_1}(D_{k_2}(E_{k_1}(m)))$$

Decryption:

$$m = D_{k_1}(E_{k_2}(D_{k_1}(c)))$$



AES

Block length 128 bits.

Key lengths 128 (or 192 or 256).

The AES is an iterated cipher with $N_r=10$ (or 12 or 14)

In each round we have:

- Subkey mixing: $\text{State} \leftarrow \text{Roundkey XOR State}$
- A substitution: $\text{SubBytes}(\text{State})$
- A permutation: $\text{ShiftRows}(\text{State}) \ \& \ \text{MixColumns}(\text{State})$



Modes of operation

Four basic modes of operation are available for block ciphers:

- Electronic codebook mode: ECB
- Cipher block chaining mode: CBC
- Cipher feedback mode: CFB
- Output feedback mode: OFB



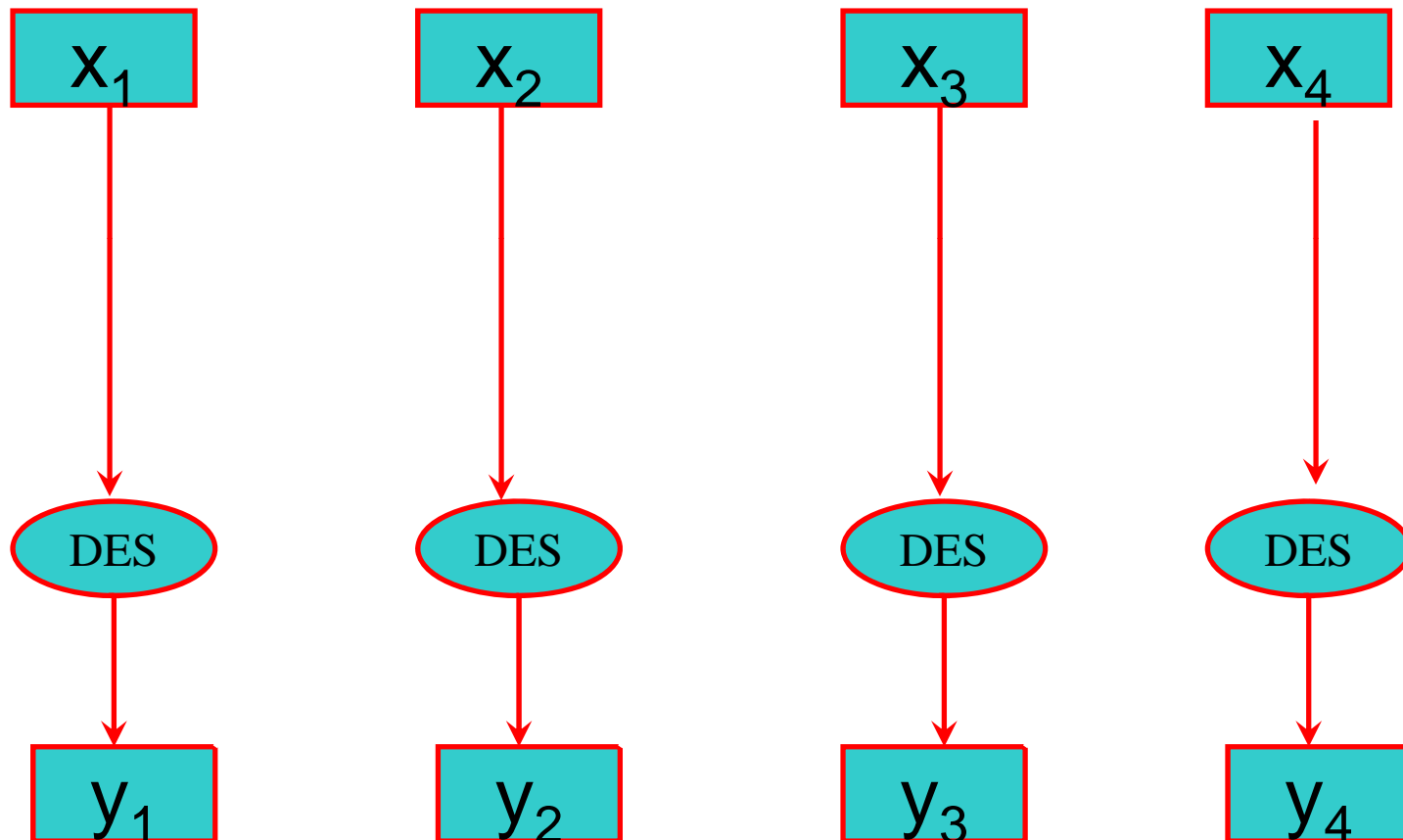
Electronic Codebook mode, ECB

Each plaintext x_i is encrypted with the same key K :

$$y_i = e_K(x_i).$$

So, the naïve use of a block cipher.

ECB





Cipher Block Chaining mode, CBC

Each cipher block y_{i-1} is XOR-ed with the next plaintext x_i :

$$y_i = e_K(y_{i-1} \text{ XOR } x_i)$$

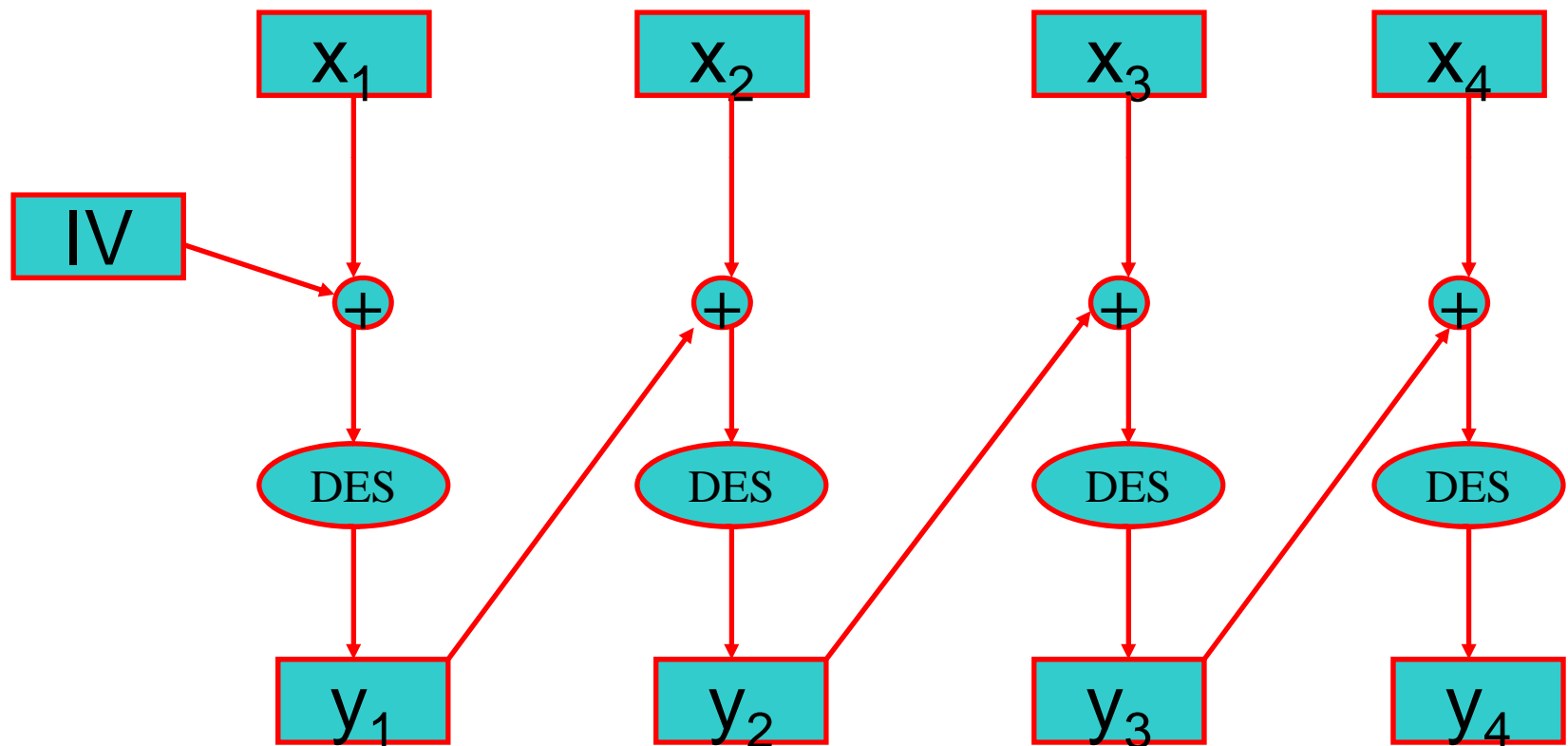
before being encrypted to get the next plaintext y_i .

The chain is initialized with

an initialization vector: $y_0 = IV$

with length, **the block size**.

CBC





Cipher and Output feedback modes (CFB & OFB)

CFB

$z_0 = IV$ and recursively:

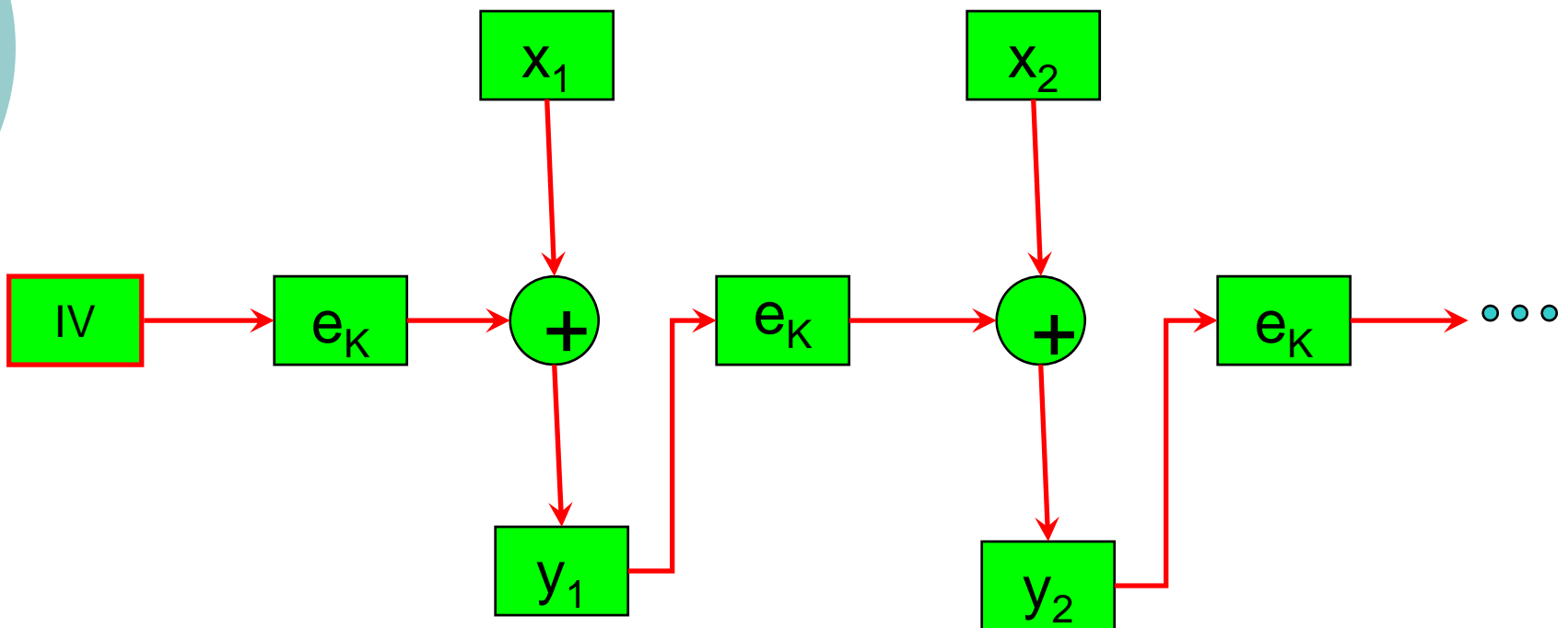
$$z_i = e_K(y_{i-1}) \text{ and } y_i = x_i \text{ XOR } z_i$$

OFB

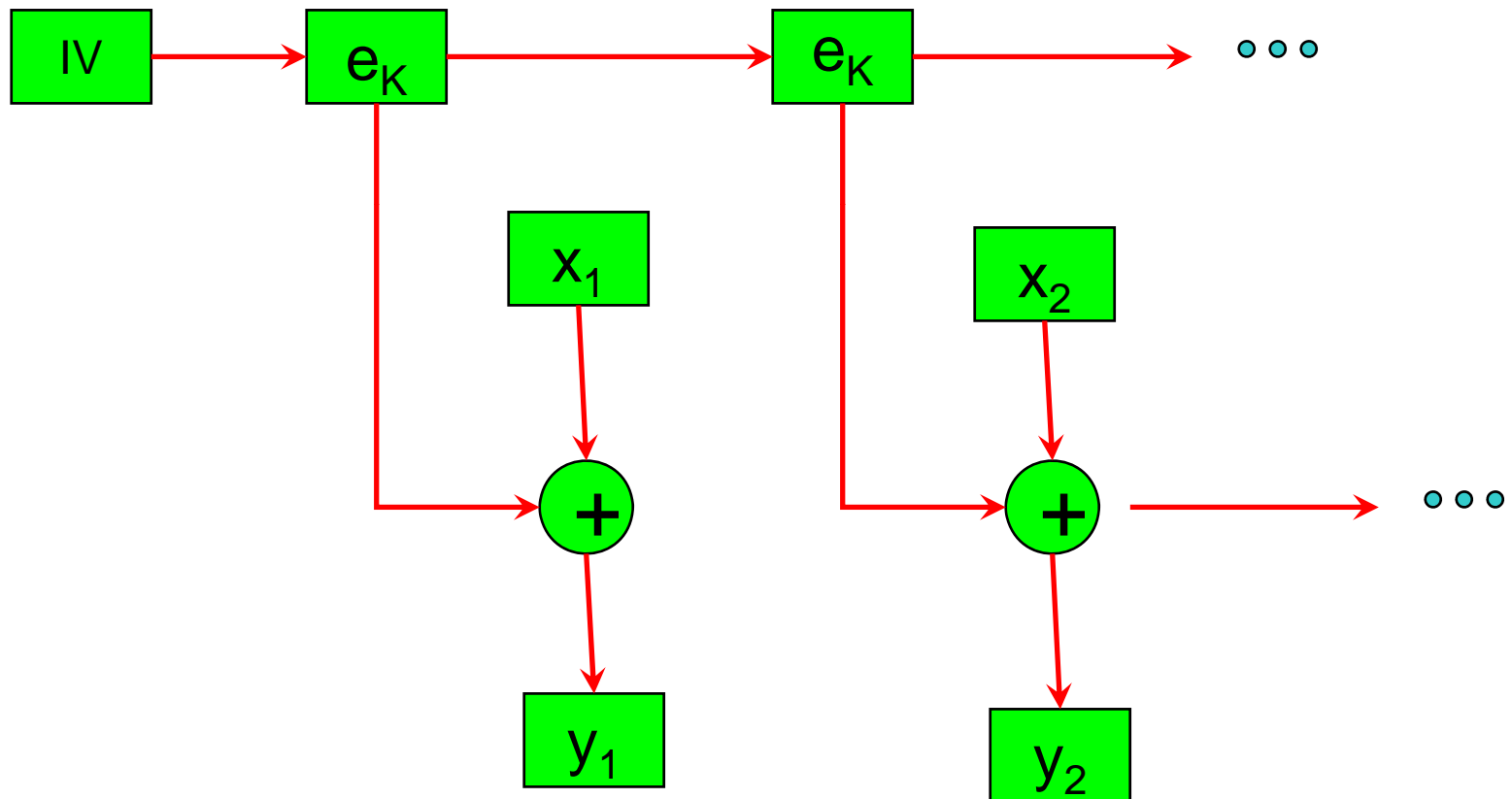
$z_0 = IV$ and recursively:

$$z_i = e_K(z_{i-1}) \text{ and } y_i = x_i \text{ XOR } z_i$$

CFB mode



OFB mode





Key Channel Establishment for symmetric cryptosystems

- Conventional techniques
- Public-key techniques
- Quantum Key distribution techniques