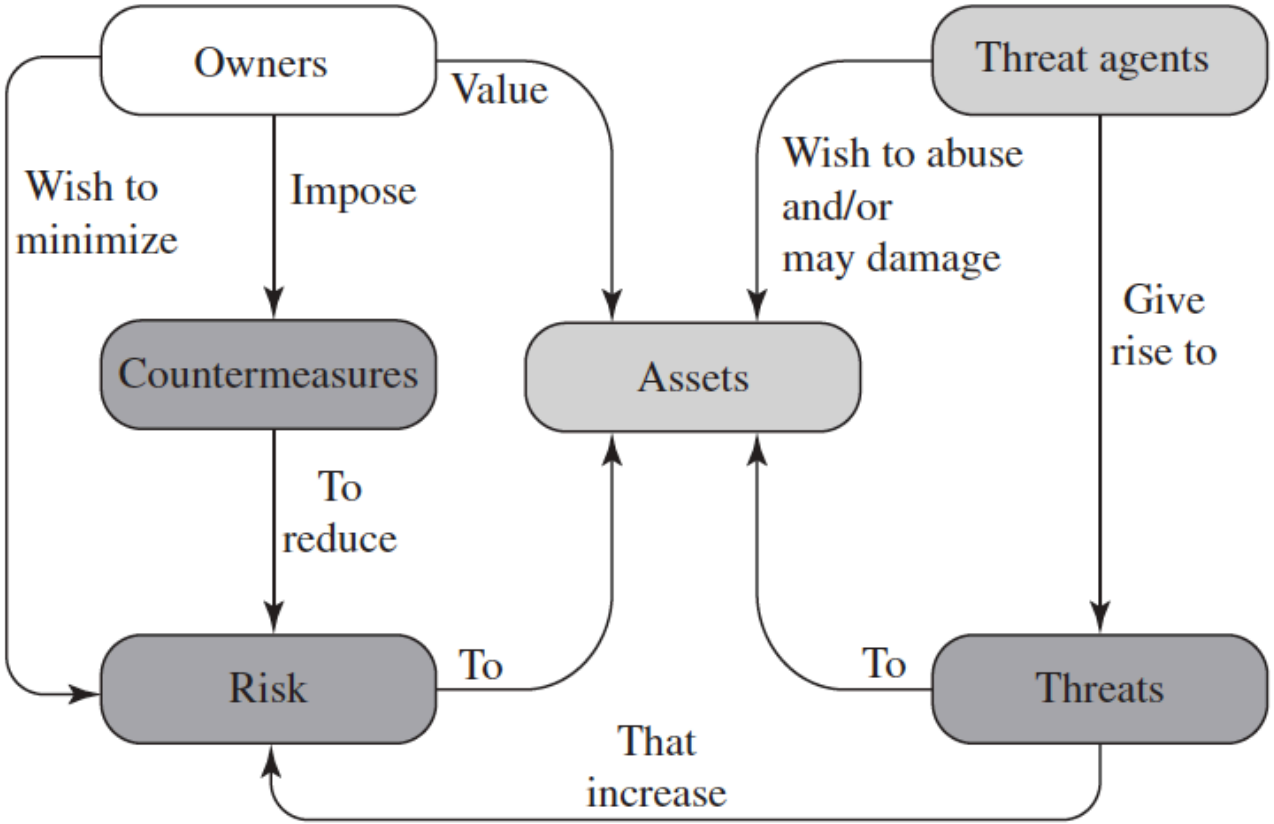
Computer Security Strategy

… helps to reply these questions:

1. What assets do we need to protect?
2. How are those assets threatened?
3. What can we do to counter those threats?

Security Concepts and Relationships



Computer Security Overview

- The NIST Computer Security Handbook defines the term computer security as:

Computer Security: The protection afforded to an automated Information system in order to attain the applicable objectives of **preserving the integrity, availability, and confidentiality of information system resources** (includes hardware, software, firmware, information/data, and telecommunications).

Three key objectives 1/3

Confidentiality: This term covers two related concepts:

- **Data confidentiality :** Assures that private or confidential information is not made available or disclosed to unauthorized individuals.
- **Privacy :** Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Three key objectives 2/3

Integrity: This term covers two related concepts:

- **Data integrity :** Assures that information and programs are changed only in a specified and authorized manner.
- **System integrity :** Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Three key objectives 3/3

- **Availability**: Assures that systems work promptly and service is not denied to authorized users.

In terms of Security Controls…

- **Confidentiality**: Preserving authorized restrictions on information access and disclosure.
- **Integrity**: Guarding against improper information modification or destruction, including ensuring information non-repudiation and authenticity.
- **Availability**: Ensuring timely and reliable access to and use of Information.

Additional security concepts

- **Authenticity**: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.
- **Accountability**: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. Because truly secure systems aren't yet an achievable goal, we must be able to trace a security breach to a responsible party.

Basic Cryptography

- Classical Cryptography
- Public Key Cryptography

Overview

- Security Requirements
- Classical Cryptography
  - Cæsar cipher
  - Vigènere cipher
  - DES
  - 3DES
- Public Key Cryptography
  - RSA

Security Requirements (1)

- Confidentiality
  - Only the owner of the private key knows it, so text enciphered cannot be read by anyone except the owner of the private key
- Authentication
  - Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner

Security Requirements (2)

- Integrity
  - Enciphered letters cannot be changed undetectably without knowing private key
- Non-Repudiation
  - Message enciphered with private key came from someone who knew it

Cryptosystem

- Quintuple $(\mathcal{E}, \mathcal{D}, \mathcal{M}, \mathcal{K}, C)$
  - $\mathcal{M}$ set of plaintexts
  - $\mathcal{K}$ set of keys
  - C set of ciphertexts
  - $\mathcal{E}$ set of encryption functions e: $\mathcal{M} \times \mathcal{K} \rightarrow C$
  - $\mathcal{D}$ set of decryption functions d: $C \times \mathcal{K} \rightarrow \mathcal{M}$

Example

- Example: Cæsar cipher
  - $\mathcal{M}$ = { sequences of letters }
  - $\mathcal{K}$ = { i | i is an integer and $0 \le i \le 25$ }
  - $\mathcal{E}$ = { E_k | $k \in \mathcal{K}$ and for all letters m,
    $E_k(m) = (m + k) \bmod 26$ }
  - $\mathcal{D}$ = { D_k | $k \in \mathcal{K}$ and for all letters c,
    $D_k(c) = (26 + c - k) \bmod 26$ }
  - $C = \mathcal{M}$

Attacks

- Opponent whose goal is to break cryptosystem is the *adversary*
  - Assume adversary knows algorithm used, but not key
- Three types of attacks:
  - *ciphertext only*: adversary has only ciphertext; goal is to find plaintext, possibly key
  - *known plaintext*: adversary has ciphertext, corresponding plaintext; goal is to find key
  - *chosen plaintext*: adversary may supply plaintexts and obtain corresponding ciphertext; goal is to find key

Basis for Attacks

- Mathematical attacks
  - Based on analysis of underlying mathematics
- Statistical attacks
  - Make assumptions about the distribution of letters, pairs of letters (digrams), triplets of letters (trigrams), *etc*.
    - Called *models of the language*
  - Examine ciphertext, correlate properties with the assumptions.

More Definitions

- **unconditional security**
  - no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- **computational security**
  - given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Brute Force Search

- always possible to simply try every key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu$s = 35.8 minutes	2.15 milliseconds
56 (DES)	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu$s = 1142 years	10.01 hours
128 (AES)	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu$s = 5.4×10^{24} years	5.4×10^{18} years
168 (3-DES)	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu$s = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu$s = 6.4×10^{12} years	6.4×10^6 years

Classical Cryptography

- Sender, receiver share common key
  - Keys may be the same, or trivial to derive from one another
  - Sometimes called *symmetric cryptography*
- Two basic types
  - Transposition ciphers
  - Substitution ciphers
  - Combinations are called *product ciphers*

Transposition Cipher

- Rearrange letters in plaintext to produce ciphertext
- Example (Rail-Fence Cipher)
  - Plaintext is `HELLO WORLD`
  - Rearrange as

    ```
    HLOOL
    ELWRD
    ```

  - Ciphertext is `HLOOL ELWRD`

Attacking the Cipher

- Anagramming
  - If 1-gram frequencies match English frequencies, but other *n*-gram frequencies do not, probably transposition
  - Rearrange letters to form *n*-grams with highest frequencies

Example

- Ciphertext: `HLOOLELWRD`
- Frequencies of 2-grams beginning with H
  - HE 0.0305
  - HO 0.0043
  - HL, HW, HR, HD < 0.0010
- Frequencies of 2-grams ending in H
  - WH 0.0026
  - EH, LH, OH, RH, DH ≤ 0.0002
- Implies E follows H

Example

- Arrange so the H and E are adjacent

```
HE
LL
OW
OR
LD
```

- Read off across, then down, to get original plaintext

Row Transposition Ciphers

- a more complex transposition
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

```
Key:        4 3 1 2 5 6 7
Plaintext: a t t a c k p
           o s t p o n e
           d u n t i l t
           w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Substitution Ciphers

- Change characters in plaintext to produce ciphertext;
- Note on char codes: A=0, B=1,C=2,….
- Example (Cæsar cipher)
  - Plaintext is `HELLO WORLD`
  - Change each letter to the third letter following it (X goes to A, Y to B, Z to C)
    - Key is 3
  - Ciphertext is `KHOOR ZRUOG`

Attacking the Cipher

- Exhaustive search
  - If the key space is small enough, try all possible keys until you find the right one
  - Cæsar cipher has 26 possible keys
- Statistical analysis
  - Compare to 1-gram model of English

Statistical Attack

- Compute frequency of each letter in ciphertext:

  G 0.1 H 0.1 K 0.1 O 0.3

  R 0.2 U 0.1 Z 0.1

- Apply 1-gram model of English
  - Frequency of characters (1-grams) in English is on next slide

Character Frequencies

a	0.080	h	0.060	n	0.070	t	0.090
b	0.015	i	0.065	o	0.080	u	0.030
c	0.030	j	0.005	p	0.020	v	0.010
d	0.040	k	0.005	q	0.002	w	0.015
e	0.130	l	0.035	r	0.065	x	0.005
f	0.020	m	0.030	s	0.060	y	0.020
g	0.015					z	0.002

Statistical Analysis

- $f(c)$ frequency of character c in ciphertext;
- $p(x)$ is frequency of character x in English;
- $\varphi(i)$ correlation of frequency of letters in ciphertext with corresponding letters in English, assuming key is i
  - $\varphi(i) = \Sigma_{0 \le c \le 25} f(c)p(c - i)$ so here,

    $\varphi(i) = 0.1p(6 - i) + 0.1p(7 - i) + 0.1p(10 - i) + 0.3p(14 - i) + 0.2p(17 - i) + 0.1p(20 - i) + 0.1p(25 - i)$
- We need to maximize the correlation function

Correlation: $\varphi(i)$ for $0 \le i \le 25$

i	$\varphi(i)$	i	$\varphi(i)$	i	$\varphi(i)$	i	$\varphi(i)$
0	0.0482	7	0.0442	13	0.0520	19	0.0315
1	0.0364	8	0.0202	14	0.0535	20	0.0302
2	0.0410	9	0.0267	15	0.0226	21	0.0517
3	0.0575	10	0.0635	16	0.0322	22	0.0380
4	0.0252	11	0.0262	17	0.0392	23	0.0370
5	0.0190	12	0.0325	18	0.0299	24	0.0316
6	0.0660					25	0.0430

The Result

- Most probable keys, based on φ:
  - $i = 6, \varphi(i) = 0.0660$
    - plaintext `EBIIL TLOLA`
  - $i = 10, \varphi(i) = 0.0635$
    - plaintext `AXEEH PHKEW`
  - $i = 3, \varphi(i) = 0.0575$
    - plaintext `HELLO WORLD`
  - $i = 14, \varphi(i) = 0.0535$
    - plaintext `WTAAD LDGAS`
- Only English phrase is for $i = 3$
  - That's the key (3 or 'D')

Cæsar's Problem

- Key is too short
  - Can be found by exhaustive search
  - Statistical frequencies not concealed well
    - They look too much like regular English letters
- So make it longer
  - Multiple letters in key
  - Idea is to smooth the statistical frequencies to make cryptanalysis harder

Vigènere Cipher

- Like Cæsar cipher, but use a phrase
- Example
  - Message `THE BOY HAS THE BALL`
  - Key `VIG`
  - Encipher using Cæsar cipher for each letter:

```
key     VIGVIGVIGVIGVIGV
plain   THEBOYHASTHEBALL
cipher  OPKWWECIYOPKWIRG
```

Relevant Parts of Tableau

	G	*I*	*V*
A	G	I	V
B	H	J	W
E	L	M	Z
H	N	P	C
L	R	T	G
O	U	W	J
S	Y	A	N
T	Z	B	O
Y	E	H	T

- Tableau shown has relevant rows, columns only
- Example encipherments:
  - key V, letter T: follow V column down to T row (giving “O”)
  - Key I, letter H: follow I column down to H row (giving “P”)

Useful Terms

- *period*: length of key
  - In earlier example, period is 3
- *tableau*: table used to encipher and decipher
  - Vigènere cipher has key letters on top, plaintext letters on the left
- *polyalphabetic*: the key has several different letters
  - Cæsar cipher is monoalphabetic

Attacking the Cipher

- Approach
  - Establish period; call it *n*
  - Break message into *n* parts, each part being enciphered using the same key letter
  - Solve each part
    - You can leverage one part from another
- We will show each step

Establish Period

- Kaskski: *repetitions in the ciphertext occur when characters of the key appear over the same characters in the plaintext*
- Example:

```
key     VIGVIGVIGVIGVIGV
plain   THEBOYHASTHEBALL
cipher  OPKWWECIYOPKWIRG
```

Note the key and plaintext line up over the repetitions (underlined). As distance between repetitions is 9, the period is a factor of 9 (that is, 1, 3, or 9)

We need to evaluate the **index of coincidence**

The Target Cipher

- We want to break this cipher:

```
ADQYS MIUSB OXKKT MIBHK IZOOO
EQOOG IFBAG KAUMF VVTAA CIDTW
MOCIO EQOOG BMBFV ZGGWP CIEKQ
HSNEW VECNE DLAAV RWKXS VNSVP
HCEUT QOIOF MEGJS WTPCH AJMOC
HIUIX
```

The Target Cipher and periods

- Locate the ripetions and periods in the cipher:

ADQYS **MI**USB OXKKT **MI**BHK IZ**OO**O

EQOOG IFBAG KAUM**F** **V**VTAA **CI**DTW

MOCI**O** **EQOOG** BMB**FV** ZGGWP **CI**EKQ

HSNEW VECNE DLAAV RWKXS VNSVP

HCEUT QOIOF MEGJS WTPCH AJ**MOC**

HIUIX

Repetitions in Example

Letters	*Start*	*End*	*Distance*	*Factors*
MI	5	15	10	2, 5
OO	22	27	5	5
OEQOOG	24	54	30	2, 3, 5
FV	39	63	24	2, 2, 2, 3
AA	43	87	44	2, 2, 11
MOC	50	122	72	2, 2, 2, 3, 3
QO	56	105	49	7, 7
PC	69	117	48	2, 2, 2, 2, 3
NE	77	83	6	2, 3
SV	94	97	3	3
CH	118	124	6	2, 3

Estimate of Period

- OEQOOG is probably not a coincidence
  - It's too long for that
  - Period may be 1, 2, 3, 5, 6, 10, 15, or 30
- Most others (7/10) have 2 in their factors
- Almost as many (6/10) have 3 in their factors
- Begin with period of $2 \times 3 = 6$

One-Time Pad

- A Vigenère cipher with a random key at least as long as the message
  - Provably unbreakable
  - Why? Look at ciphertext `DXQR`. Equally likely to correspond to plaintext `DOIT` (key `AJIY`) and to plaintext `DONT` (key `AJDY`) and any other 4 letters
  - Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
    - Approximations, such as using pseudorandom number generators to generate keys, are *not* random

Product Ciphers

- ciphers using substitutions or transpositions are not secure because of language characteristics
- hence consider using several ciphers in succession to make harder, but:
  - two substitutions make a more complex substitution
  - two transpositions make more complex transposition
  - but a substitution followed by a transposition makes a new much harder cipher
- this is bridge from classical to modern ciphers

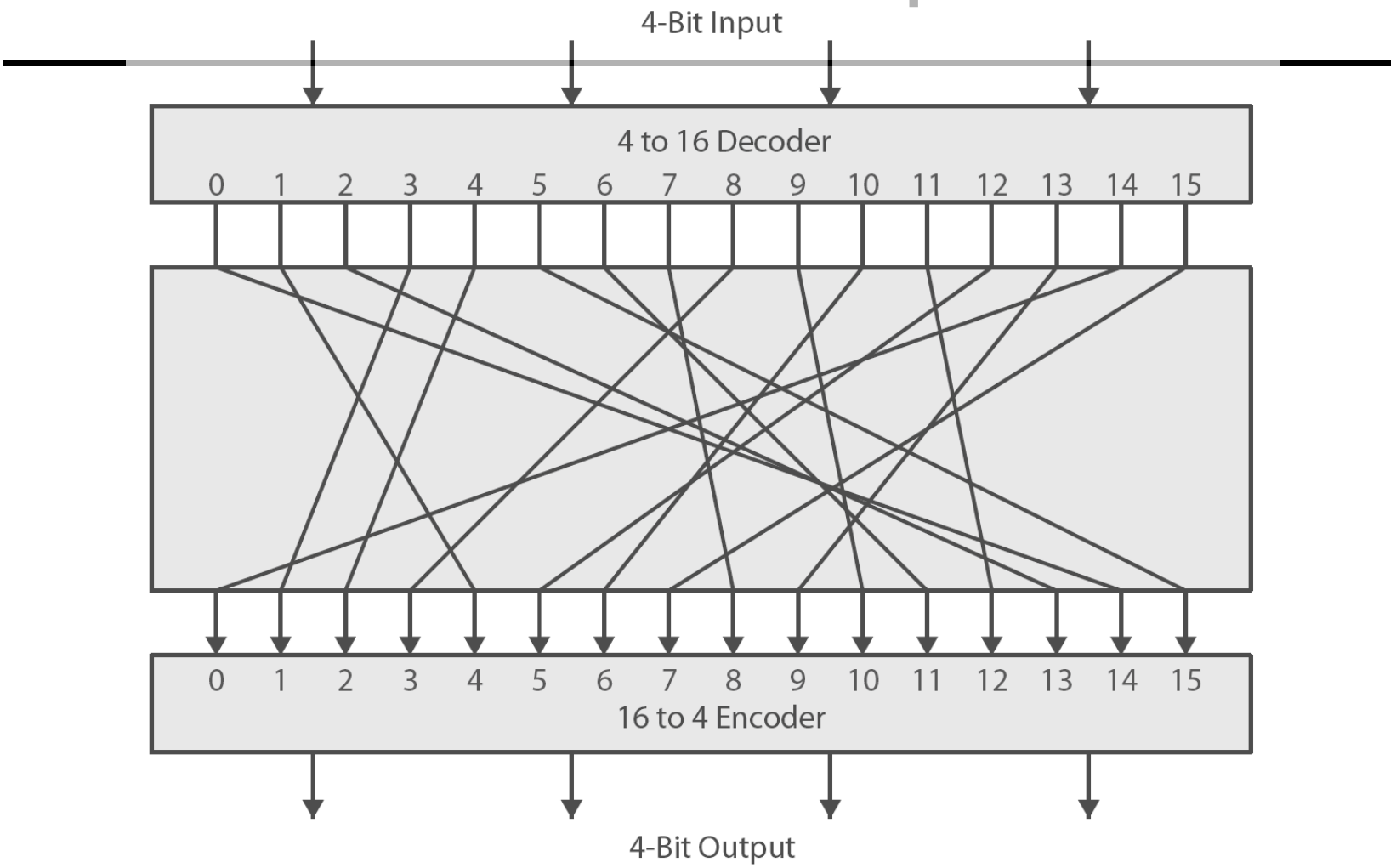
Block vs Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
- broader range of applications

Block Cipher Principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- instead create from smaller building blocks
- using idea of a product cipher

Ideal Block Cipher



Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key

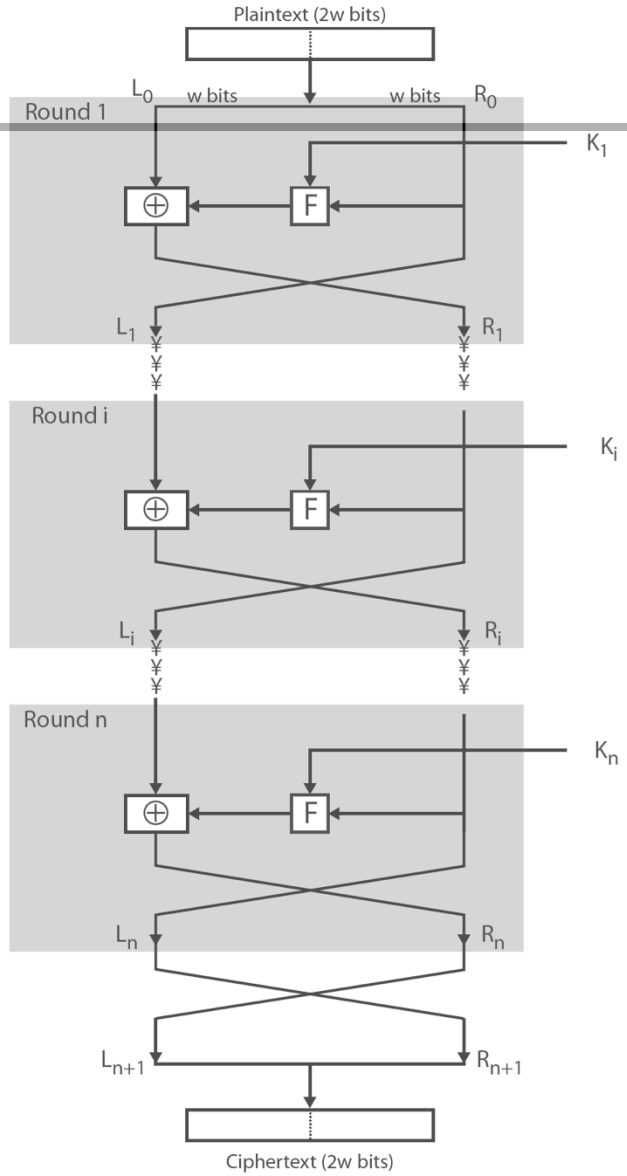
Confusion and Diffusion

- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept

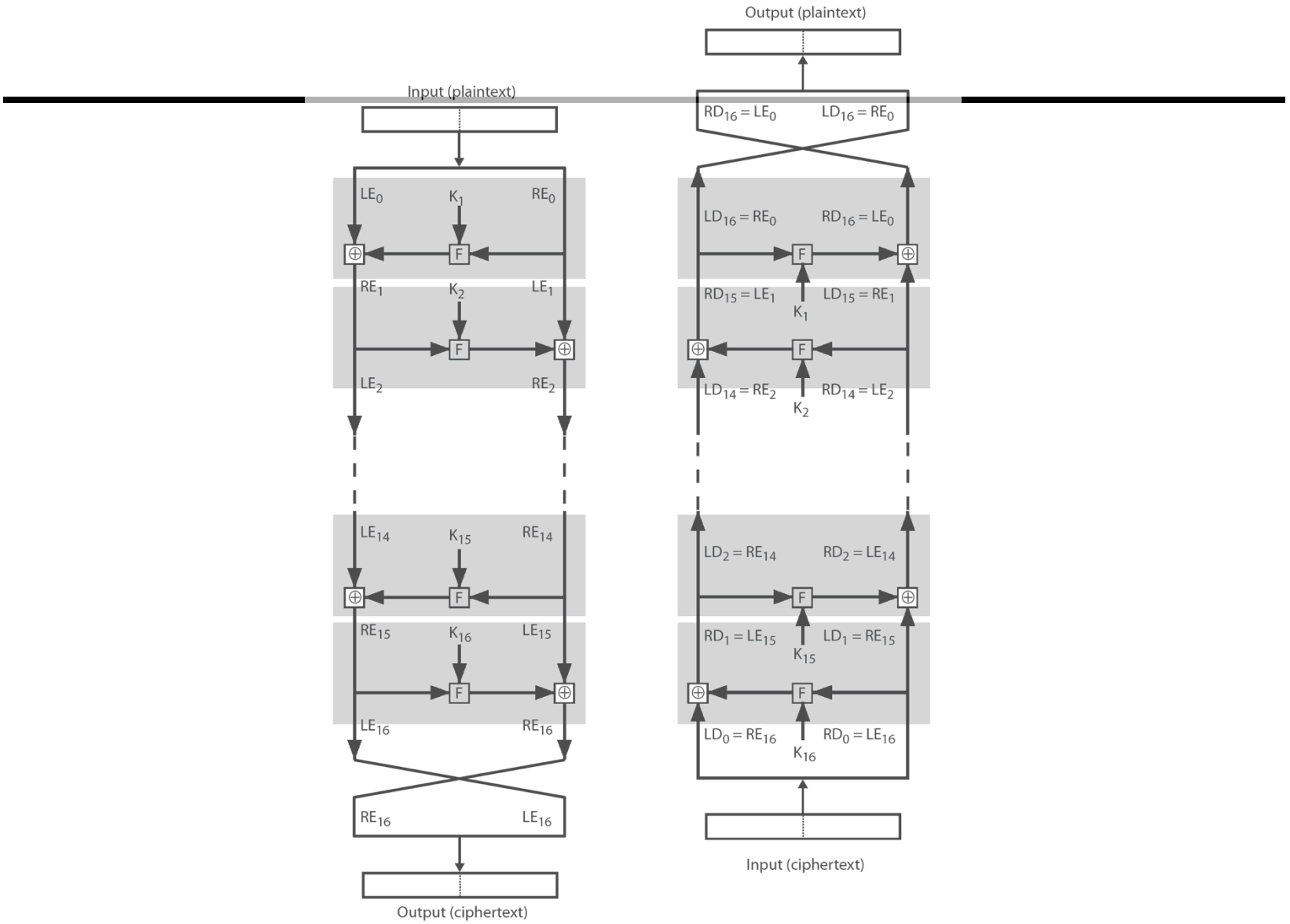
Feistel Cipher Structure



Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis

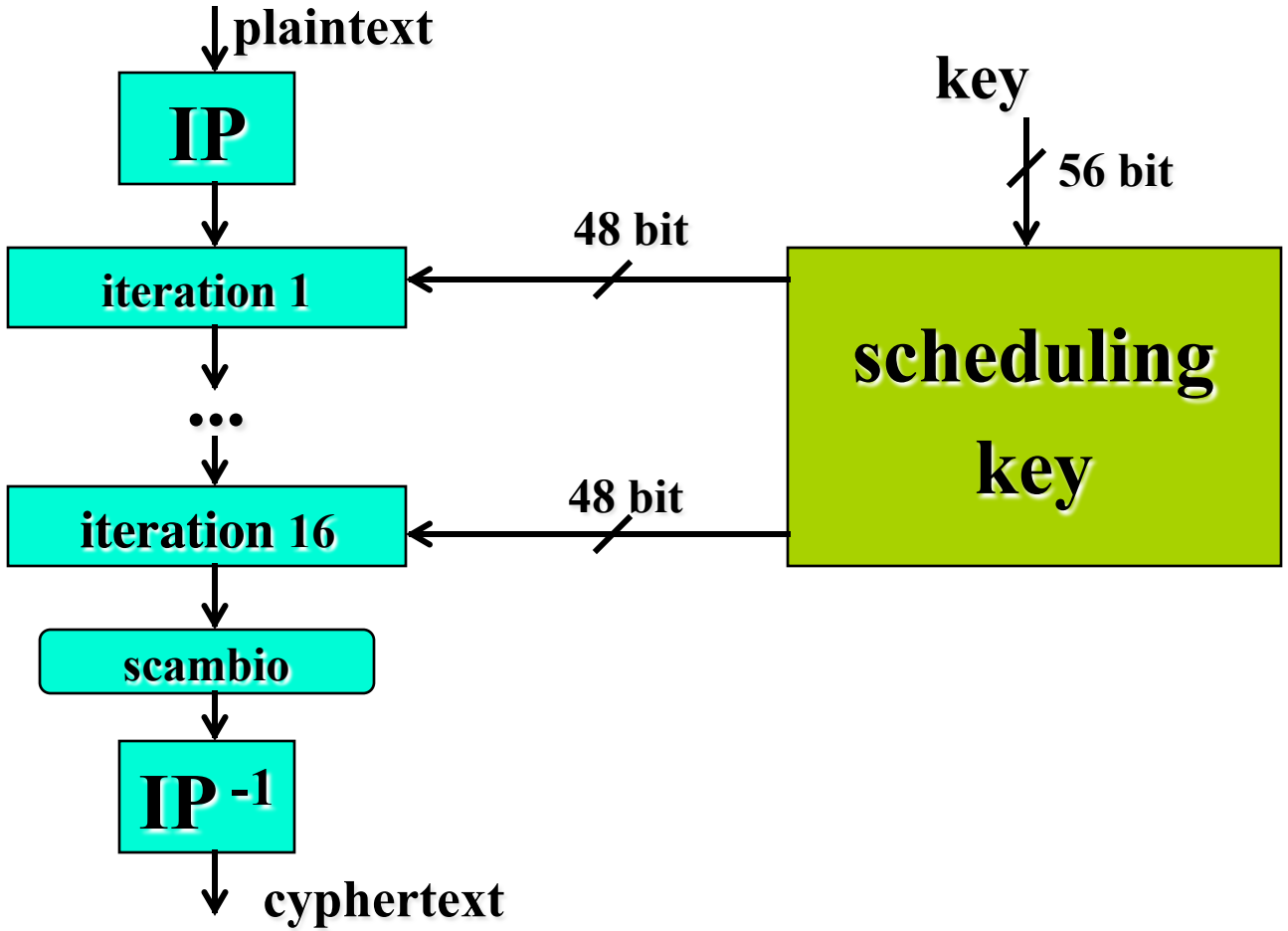
Feistel Cipher Decryption



Final overview of the DES

- A block cipher:
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext
- A product cipher
  - basic unit is the bit
  - performs both substitution and transposition (permutation) on the bits
- Cipher consists of 16 rounds (iterations) each with a round key generated from the user-supplied key
- REFERENCE: Stalling Chapter 3

DES phases

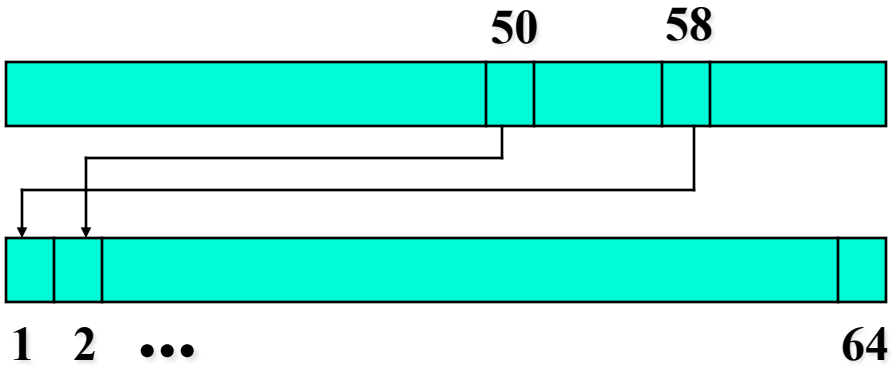

plaintext
IP
iteration 1
...
iteration 16
scambio
IP -1
cyphertext
key
56 bit
48 bit
48 bit
scheduling key

Initial Permutation IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

bit before

bit after

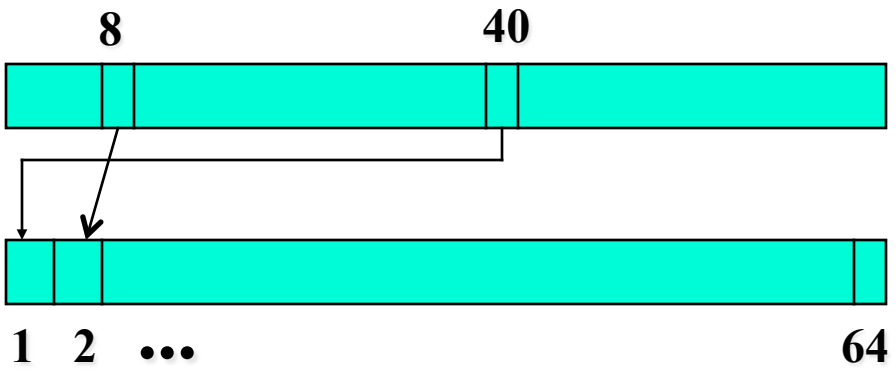


Inverse Permutation IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

bit before

bit after



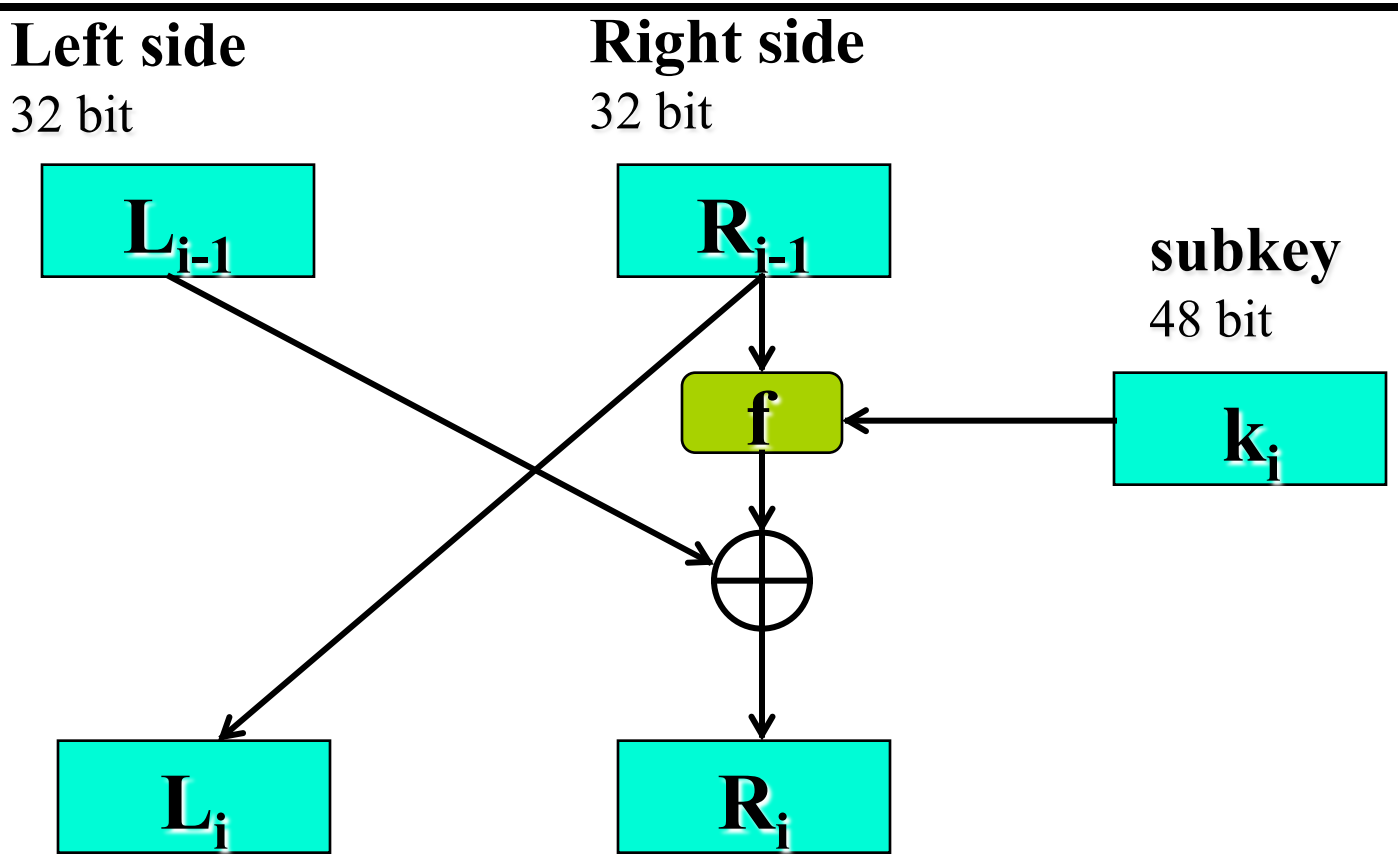
DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

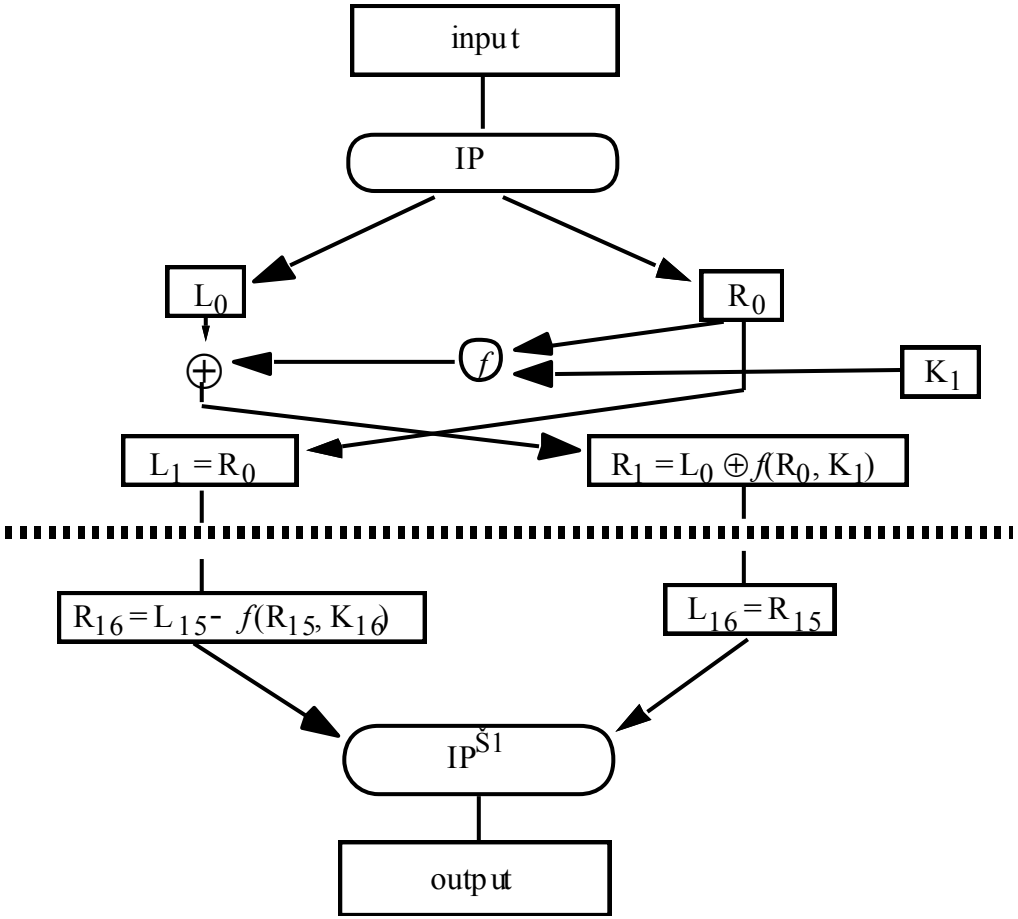
  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$
- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

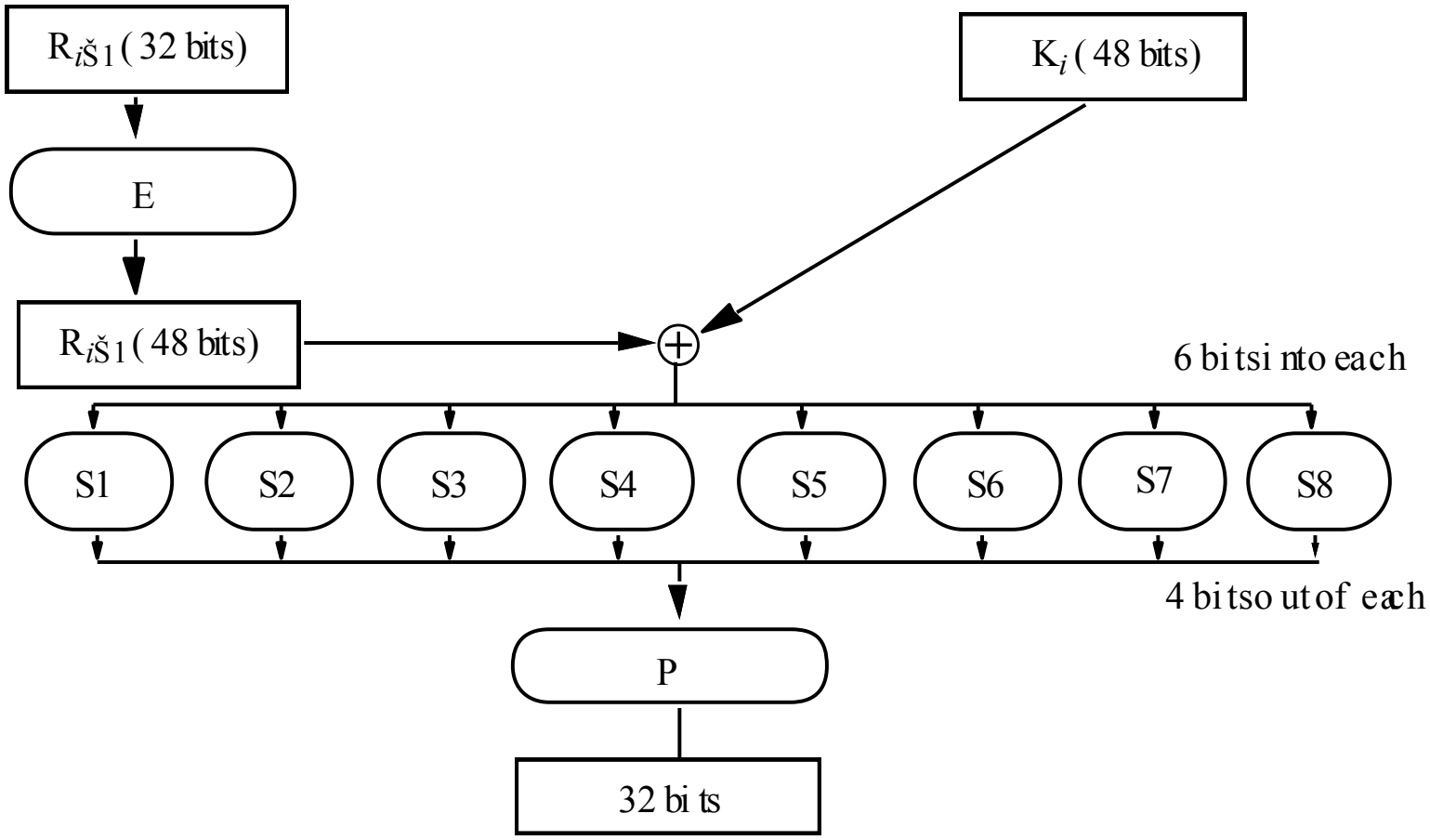
Singol Iteration



Encipherment



The *f* Function



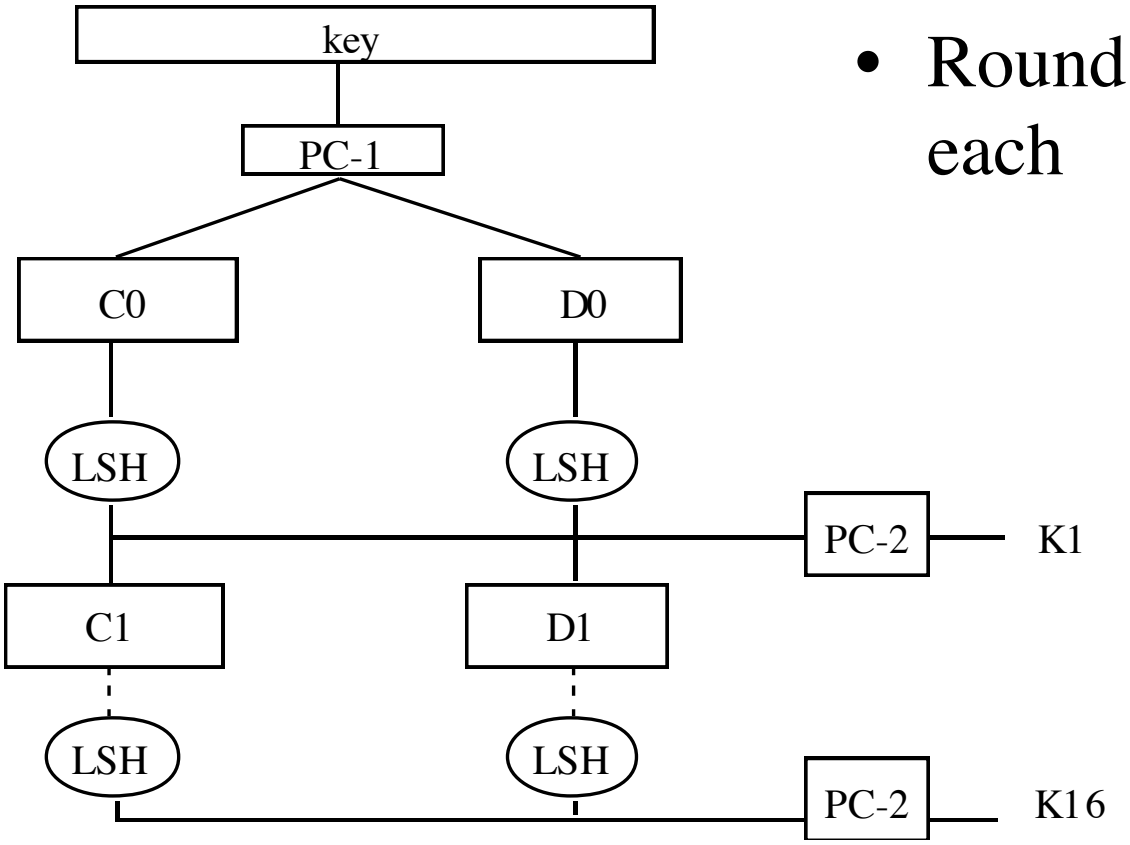
Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one row of 4 defined by the table for Si
  - inner bits 2-5 (**col** bits) select one col of 16
  - The decimal value in tha table is converted in 4 bits
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- Example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half & permuting them by PC2 (Permuted Choice Two) for use in round function F

Generation of Round Keys



- Round keys are 48 bits each

Avalanche Effect

- A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext.
- A change of **one** input or key bit results in changing approx **half** output bits.
- DES exhibits strong avalanche (Stallings Table 3.5)

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- recent advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
- still must be able to recognize plaintext
- must now consider alternatives to DES

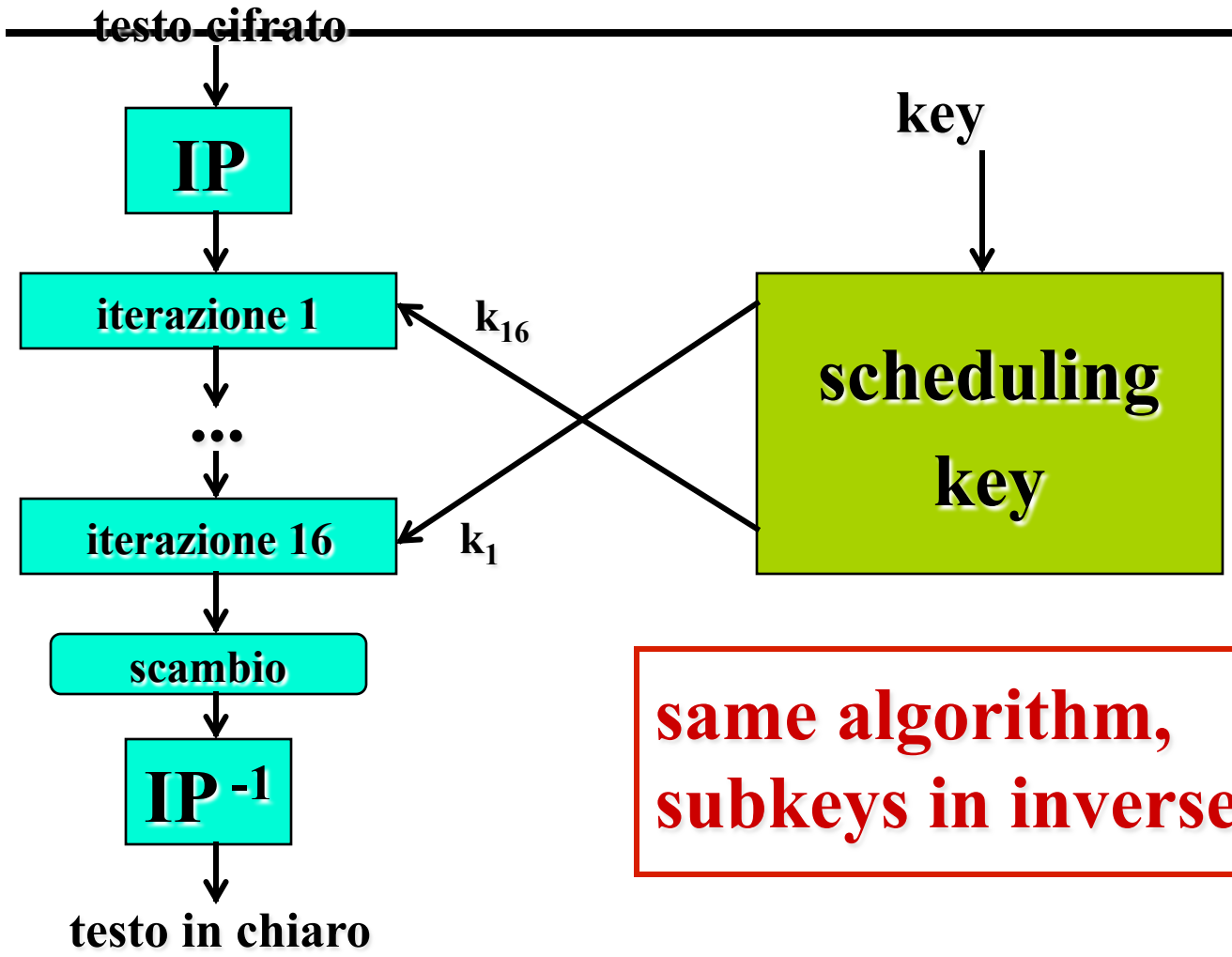
Strength of DES – Analytic Attacks

- now have several analytic attacks on DES
- these utilise some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
- include
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Controversy

- Considered too weak
  - Diffie, Hellman said in a few years technology would allow DES to be broken in days
    - Design using 1999 technology published
  - Design decisions not public (until 1994)
    - S-boxes may have backdoors

Deciphering DES



same algorithm,
subkeys in inverse order

Exsaustive research

- Key space DES = $2^{56} \approx 7{,}2056 \cdot 10^{16}$
- A PC with 500 Mhz that is able to test a key in a clock cycle, needs:

144.115.188 seconds ≈ 834 days ≈ 2 years and 3 months to test $2^{55} \approx 3{,}6 \cdot 10^{16}$ keys

DES challanges

- 10.000 dollars to the one that breaks the *challenge* within the 25% of the previous best time;
- **June 1997**: 39 days, tested 24% of 2^{56} keys, Rocke Verser distributed a research client, 70.000 computer, founded by M. K. Sanders (Pentium 90 MHz, 16M);
- **July 1998**: 56 hours, Deep Crack, EFF, 250.000 dollars;
- **Jenuary 1999**: 22 hours 15 minutes testing 245 billions of keys per second, Distributed.Net 100.000 computers;
- **2008**: COPACOBANA RIVYERA reduced the time to break DES to less than one day, using 128 Spartan-3 5000's;

Undesirable Properties

- 4 weak keys
  - They are their own inverses, so Encryption (E) and decryption (D) under a weak key have the same effect
- 12 semi-weak keys
  - Each has another semi-weak key as inverse, so Encryption with one of the pair of semiweak keys, K1 , operates identically to decryption with the other, K2
- Complementation property
  - $\mathrm{DES}_k(m) = c \Rightarrow \mathrm{DES}_k(m') = c'$
- S-boxes exhibit irregular properties
  - Distribution of odd, even numbers non-random
  - Outputs of fourth box depends on input to third box

DES modes

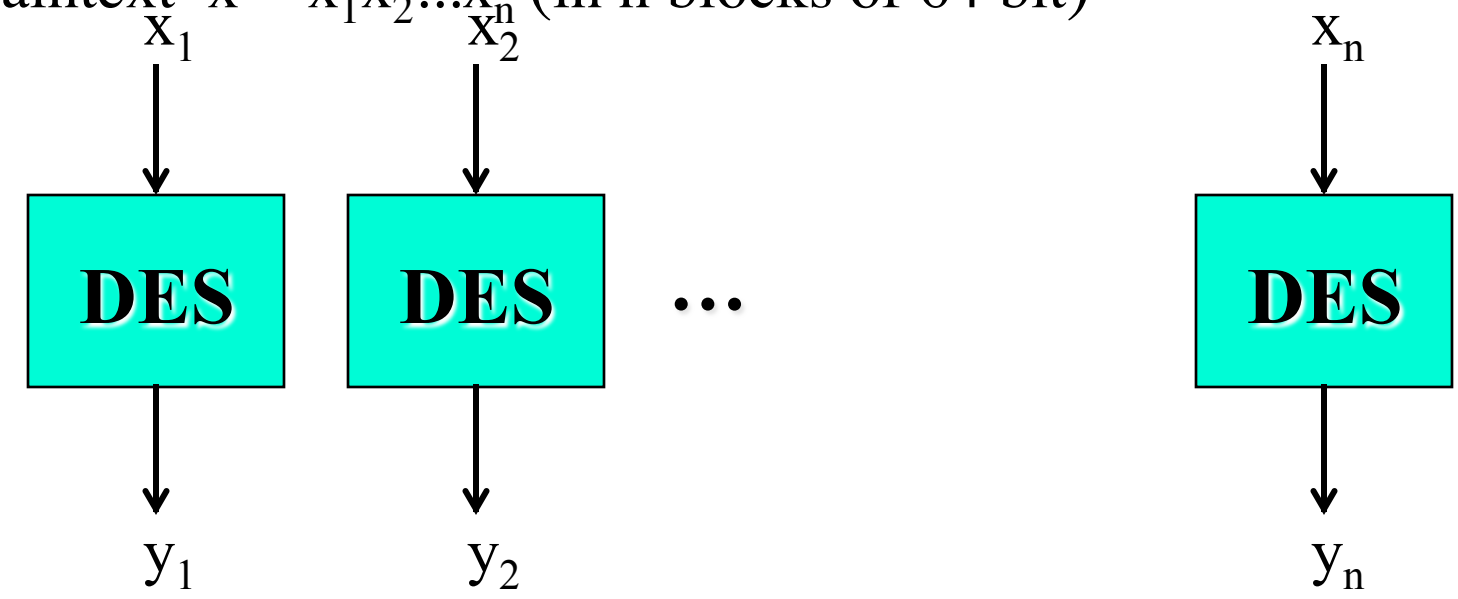
How to cipher text longer than 64 bit?

- Electronic codebook chaining (ECB)
- Cipher block chaining (CBC)
- Cipher feedback (CFB)
- Output feedback (OFB)

REFERENCE: Stalling Chapter 6

Electronic codebook chaining (ECB)

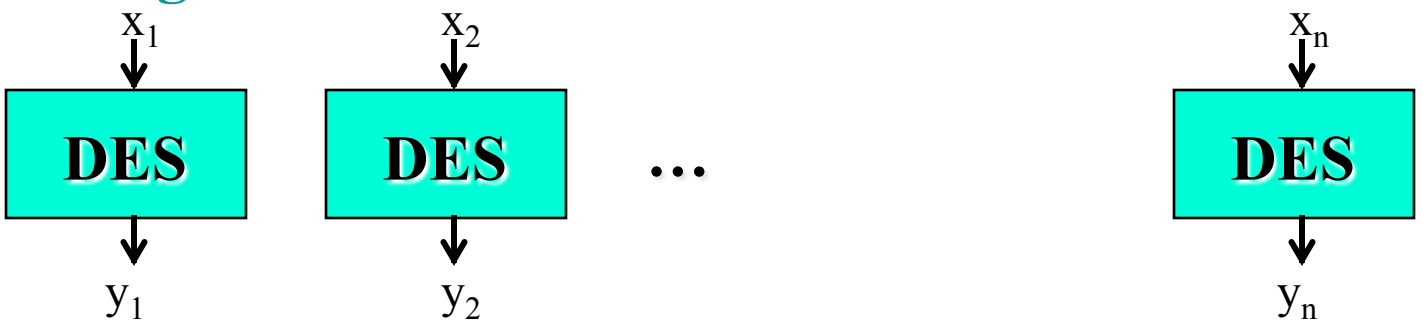
plaintext $x = x_1 x_2 ... x_n$ (in n blocks of 64 bit)



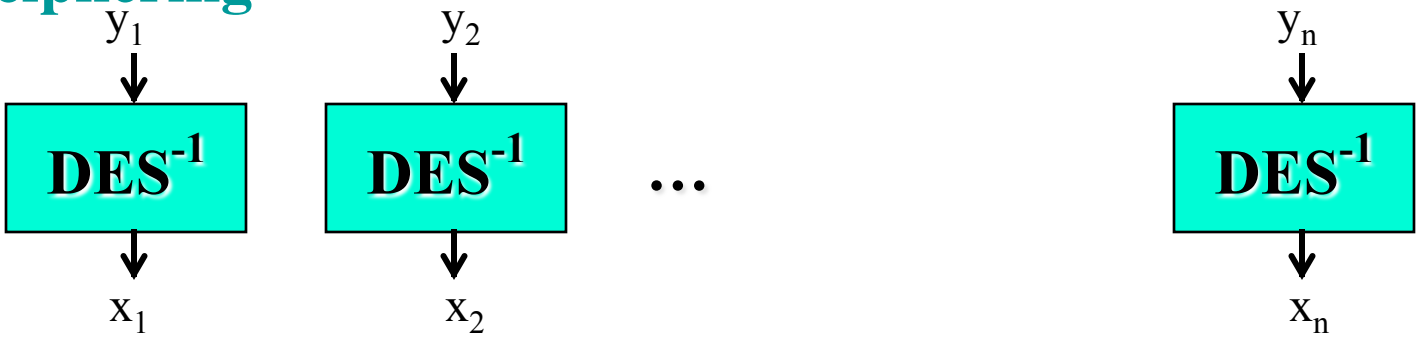
Ciphered text $y = y_1 y_2 \ldots y_n$

Electronic codebook chaining (ECB)

ciphering



deciphering



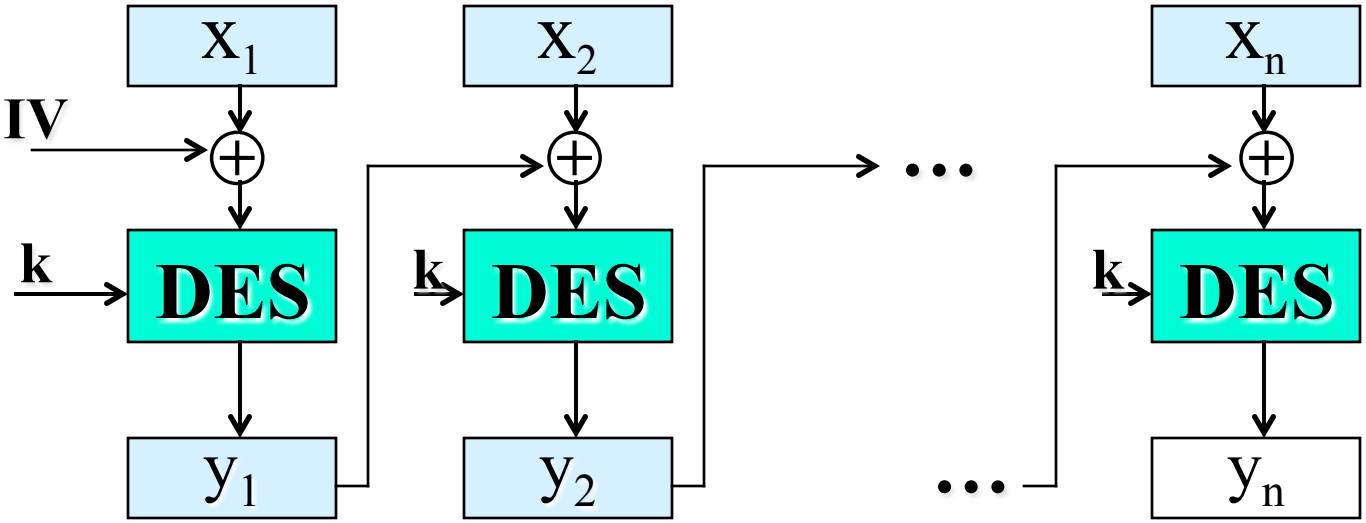
Electronic codebook chaining (ECB)

- ECB is fast
- Errors do not propagate

- Blocks are independent
  - Sobstitution attacks are possible

Cipher Block Chaining (CBC)

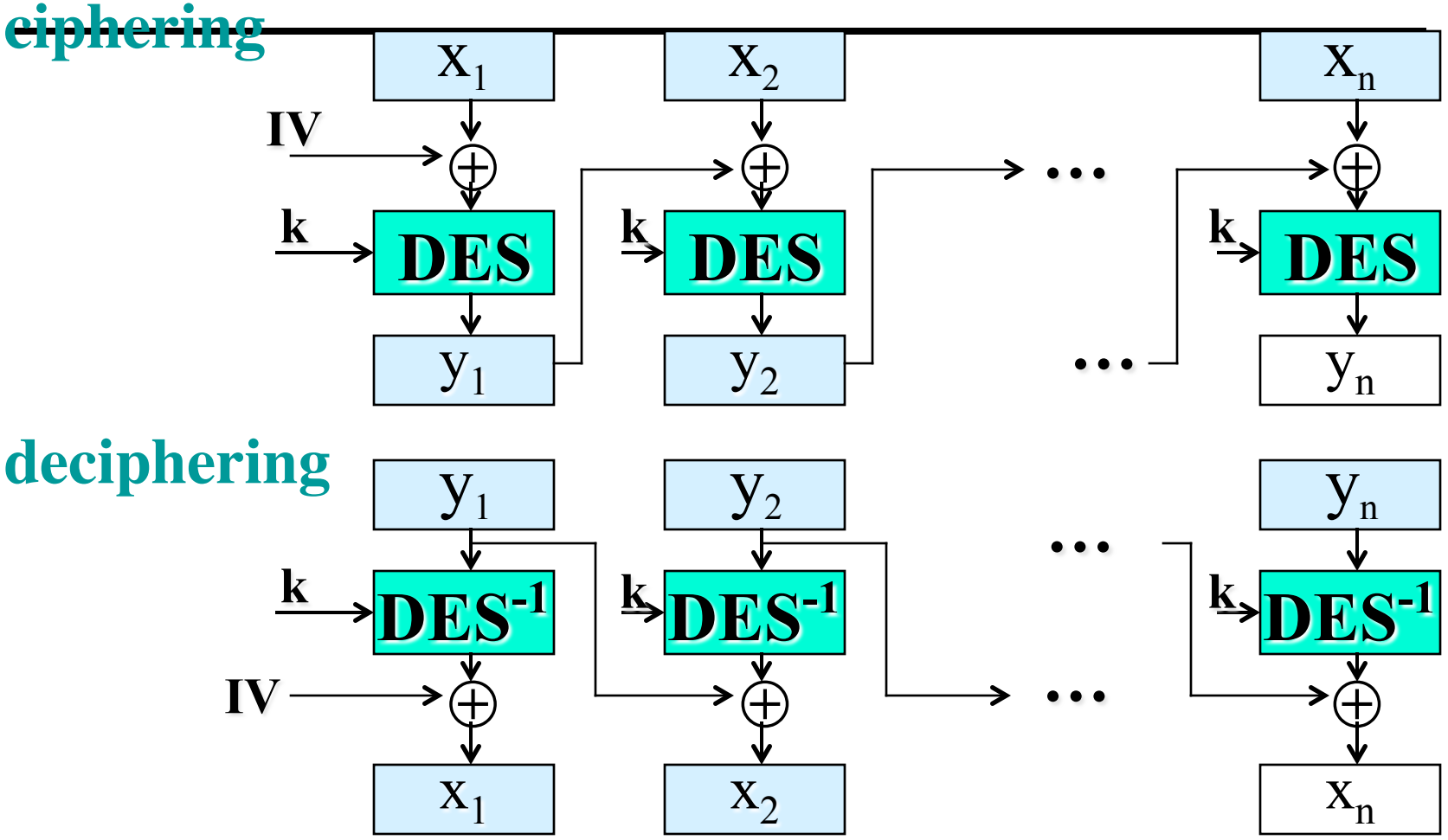
Plaintext $x=x_1x_2\ldots x_n$ (in n blocks of 64 bit)



Ciphered text $y = y_1y_2\ldots y_n$

Inizialization vector IV

Cipher Block Chaining (CBC)



An initialization vector

- An initialization vector (IV) is a block of bits that is used to randomize the encryption and hence to produce distinct ciphertexts even if the same plaintext is encrypted multiple times (without the need for a slower re-keying process).
- IV needs to be secret (new attacks!!).

Padding

- A block cipher works on units of a fixed size (known as a *block size*), but messages come in a variety of lengths. So some modes (namely ECB and CBC) require that the final block be padded before encryption. Several padding schemes exist.
- The simplest is to add null bytes to the plaintext to bring its length up to a multiple of the block size

Cipher Block Chaining (CBC)

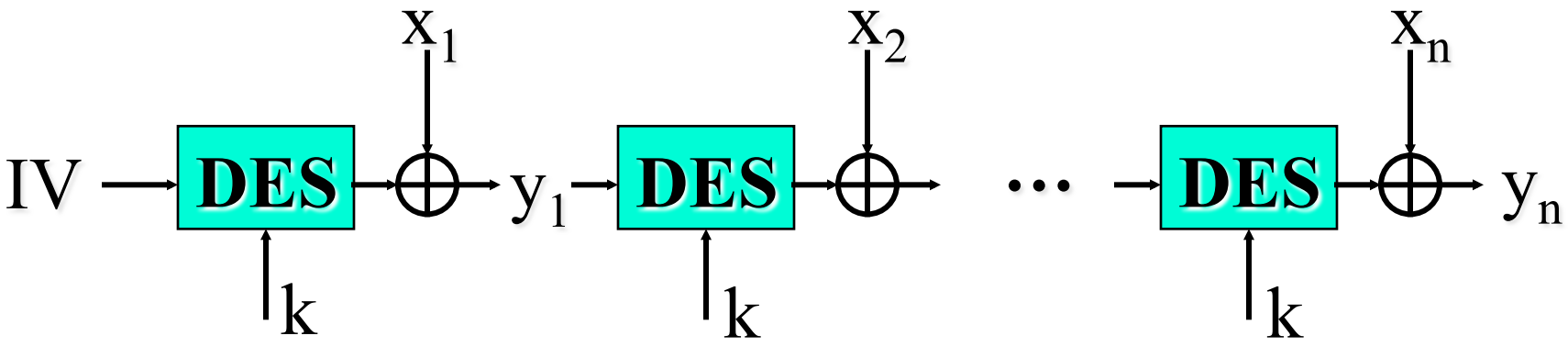
- Slower than ECB
- Errors propagate

- Blocks are dependent each other

  - no sobstitution attacks

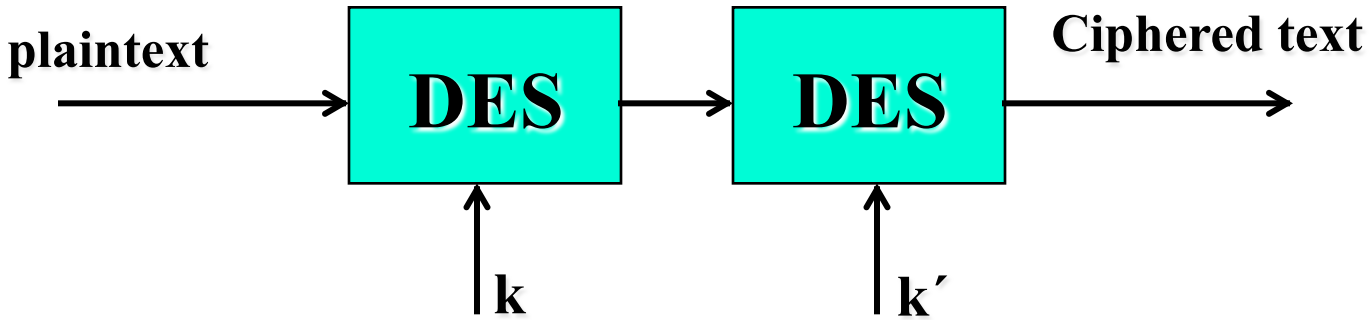
Cipher feedback (CFB)

Plaintext $x=x_1x_2...x_n$ (split in n blocks of 64 bit)



Ciphered text $y = y_1y_2\ldots y_n$

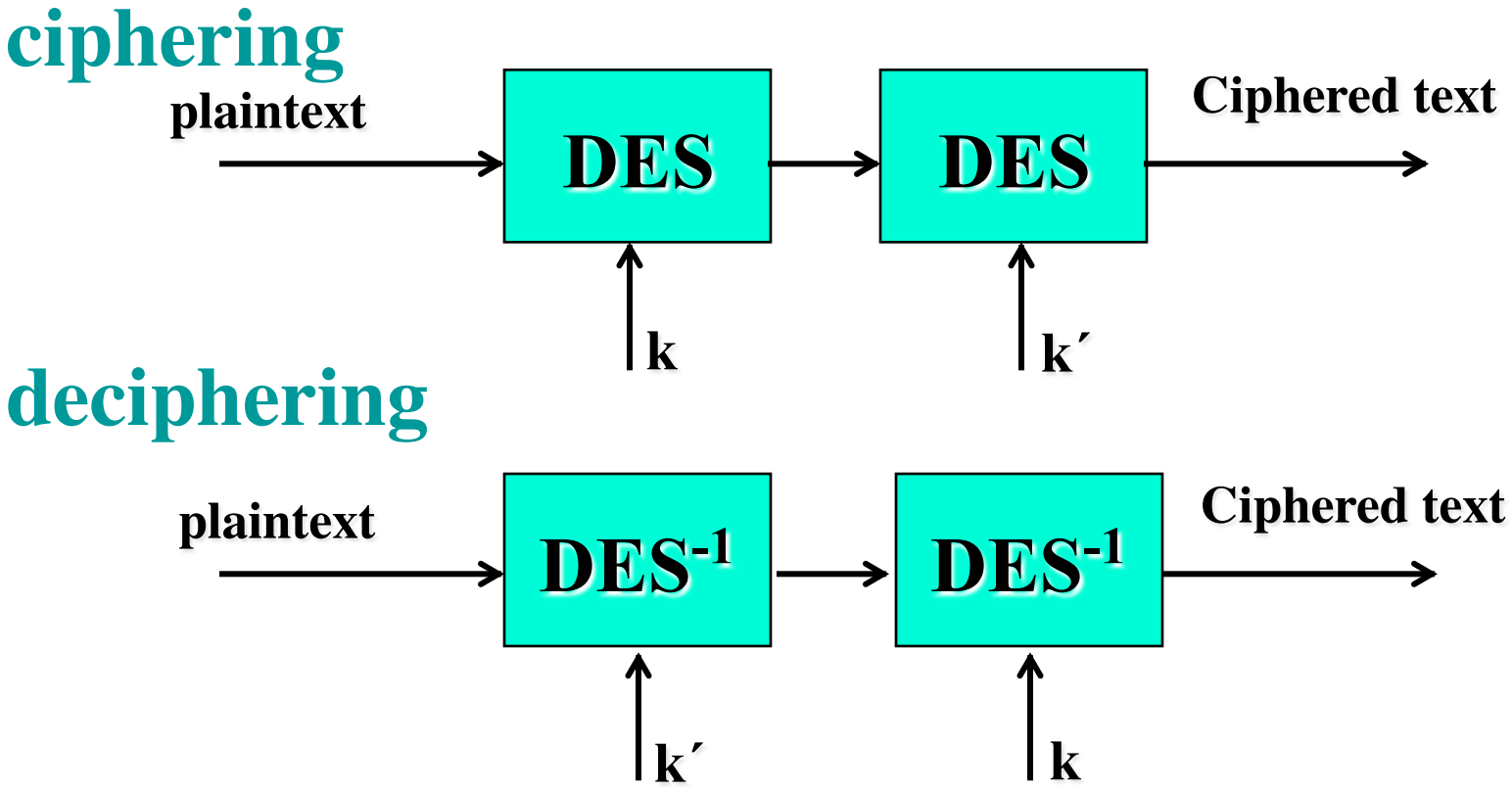
Double DES



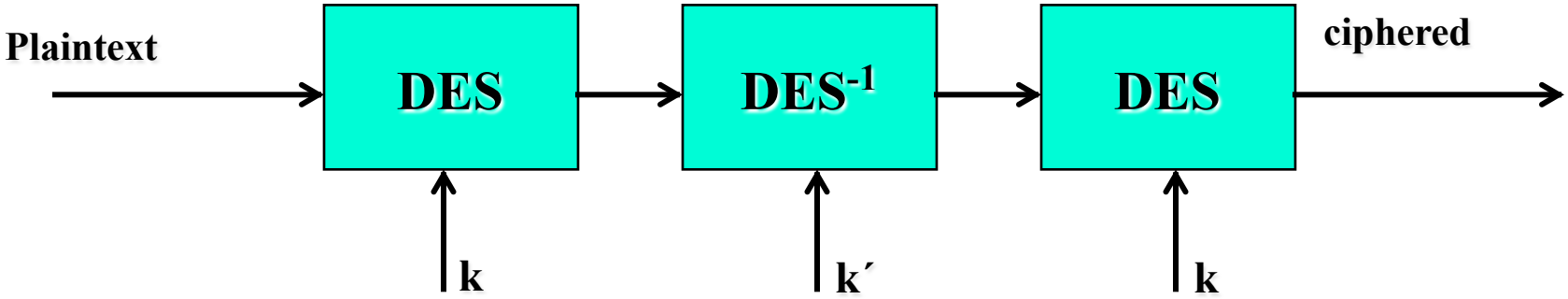
ciphering

Block lenght = 64 bit
key (k, k´) of 56+56 = 112 bit
Meet in the middle attacks!!

Double DES


ciphering
plaintext
DES
DES
Ciphered text
k
k´
deciphering
plaintext
DES-1
DES-1
Ciphered text
k´
k

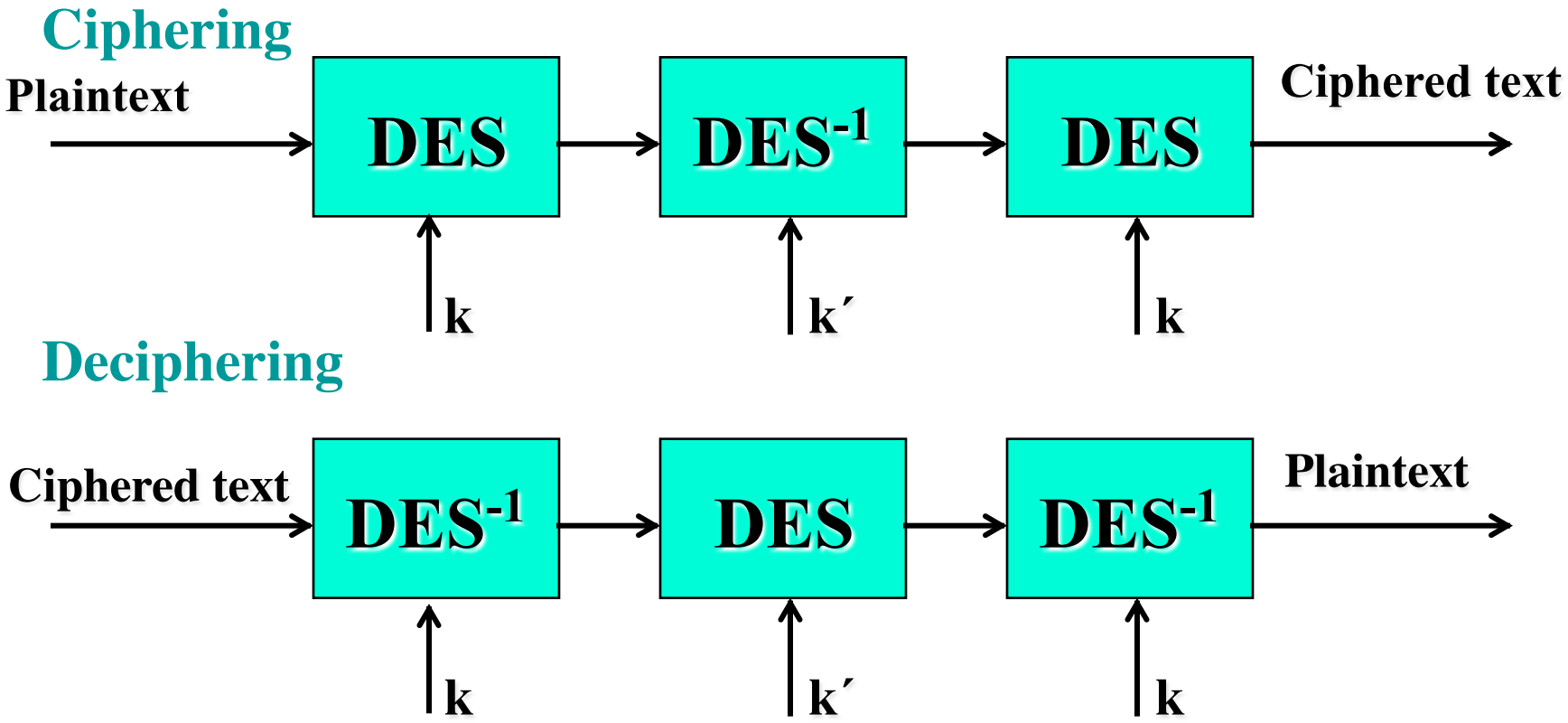
Triple DES



Ciphering

- lunghezza blocco = 64 bit
- chiave (k, k´) lunga 56+56 = 112 bit (more usable that 3 keys, same security)
- spesso chiamato $EDE_{k,k'}$ (acronimo per Encrypt Decrypt Encrypt)
- adottato negli standard X9.17 e ISO 8732

Deciphering Triple DES



Public Key Cryptography

- Two keys
  - *Private key* known only to individual
  - *Public key* available to anyone
    - Public key, private key inverses
- Idea
  - Confidentiality: encipher using public key, decipher using private key
  - Integrity/authentication: encipher using private key, decipher using public one

Requirements

1. It must be computationally easy to encipher or decipher a message given the appropriate key
2. It must be computationally infeasible to derive the private key from the public key
3. It must be computationally infeasible to determine the private key from a chosen plaintext attack

RSA

- Exponentiation cipher
- Relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*

Background

- Totient function $\phi(n)$
  - Number of positive integers less than *n* and relatively prime to *n*
    - *Relatively prime* means with no factors in common with *n*
- Example: $\phi(10) = 4$
  - 1, 3, 7, 9 are relatively prime to 10
- Example: $\phi(21) = 12$
  - 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 are relatively prime to 21

Algorithm

- Choose two large prime numbers p, q
  - Let $n = pq$; then $\phi(n) = (p-1)(q-1)$
  - Choose $e < n$ such that e is relatively prime to $\phi(n)$.
  - Compute d such that $ed \bmod \phi(n) = 1$
- Public key: (e, n); private key: d
- Encipher: $c = m^e \bmod n$
- Decipher: $m = c^d \bmod n$

Example: Confidentiality

- Take $p = 7, q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Bob wants to send Alice secret message HELLO (07 04 11 11 14)
  - $07^{17} \bmod 77 = 28$
  - $04^{17} \bmod 77 = 16$
  - $11^{17} \bmod 77 = 44$
  - $11^{17} \bmod 77 = 44$
  - $14^{17} \bmod 77 = 42$
- Bob sends 28 16 44 44 42

Example

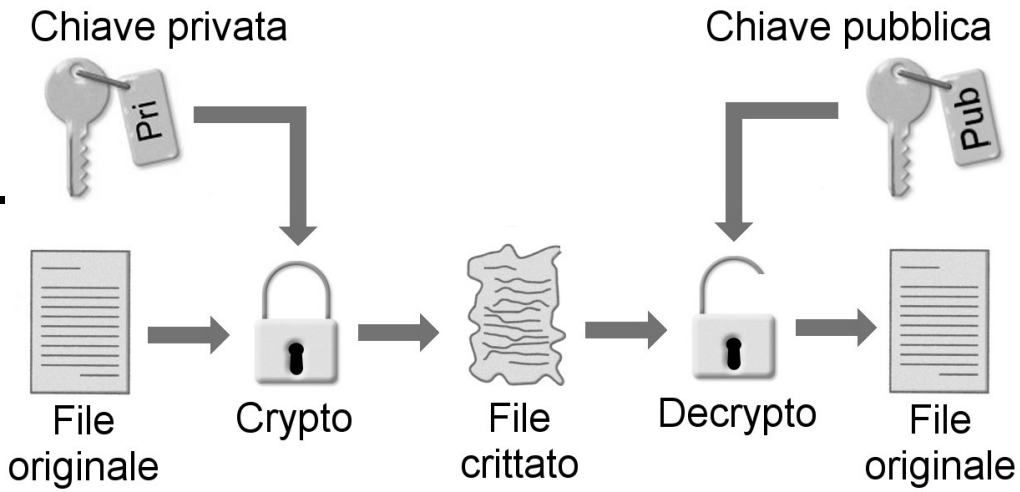
- Alice receives 28 16 44 44 42
- Alice uses private key, $d = 53$, to decrypt message:
  - $28^{53} \bmod 77 = 07$
  - $16^{53} \bmod 77 = 04$
  - $44^{53} \bmod 77 = 11$
  - $44^{53} \bmod 77 = 11$
  - $42^{53} \bmod 77 = 14$
- Alice translates message to letters to read HELLO
  - No one else could read it, as only Alice knows her private key and that is needed for decryption

Example: Integrity/ Authentication

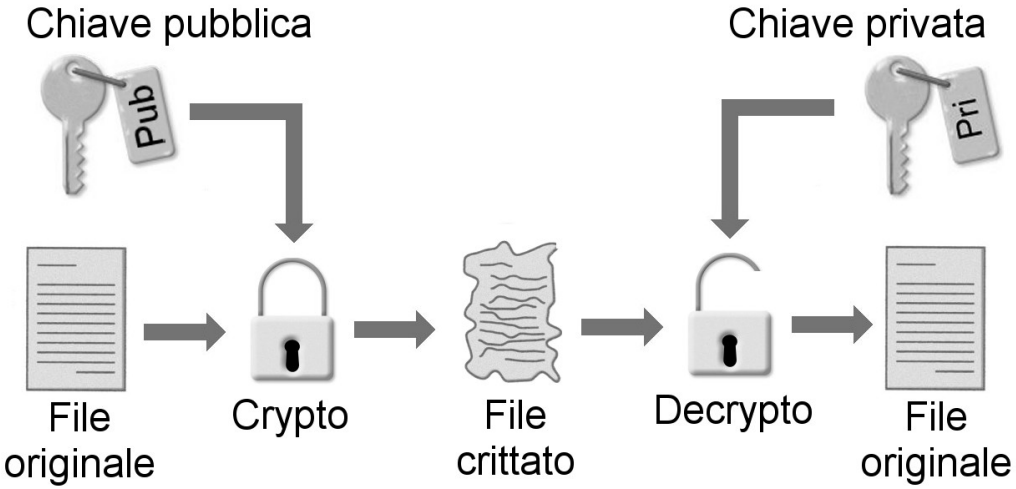
- Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Alice wants to send Bob message HELLO (07 04 11 11 14) so Bob knows it is what Alice sent (no changes in transit, and authenticated)
  - $07^{53} \bmod 77 = 35$
  - $04^{53} \bmod 77 = 09$
  - $11^{53} \bmod 77 = 44$
  - $11^{53} \bmod 77 = 44$
  - $14^{53} \bmod 77 = 49$
- Alice sends 35 09 44 44 49

Example

- Bob receives 35 09 44 44 49
- Bob uses Alice's public key, $e = 17$, $n = 77$, to decrypt message:
  - $35^{17} \bmod 77 = 07$
  - $09^{17} \bmod 77 = 04$
  - $44^{17} \bmod 77 = 11$
  - $44^{17} \bmod 77 = 11$
  - $49^{17} \bmod 77 = 14$
- Bob translates message to letters to read HELLO
  - Alice sent it as only she knows her private key, so no one else could have enciphered it
  - If (enciphered) message's blocks (letters) altered in transit, would not decrypt properly



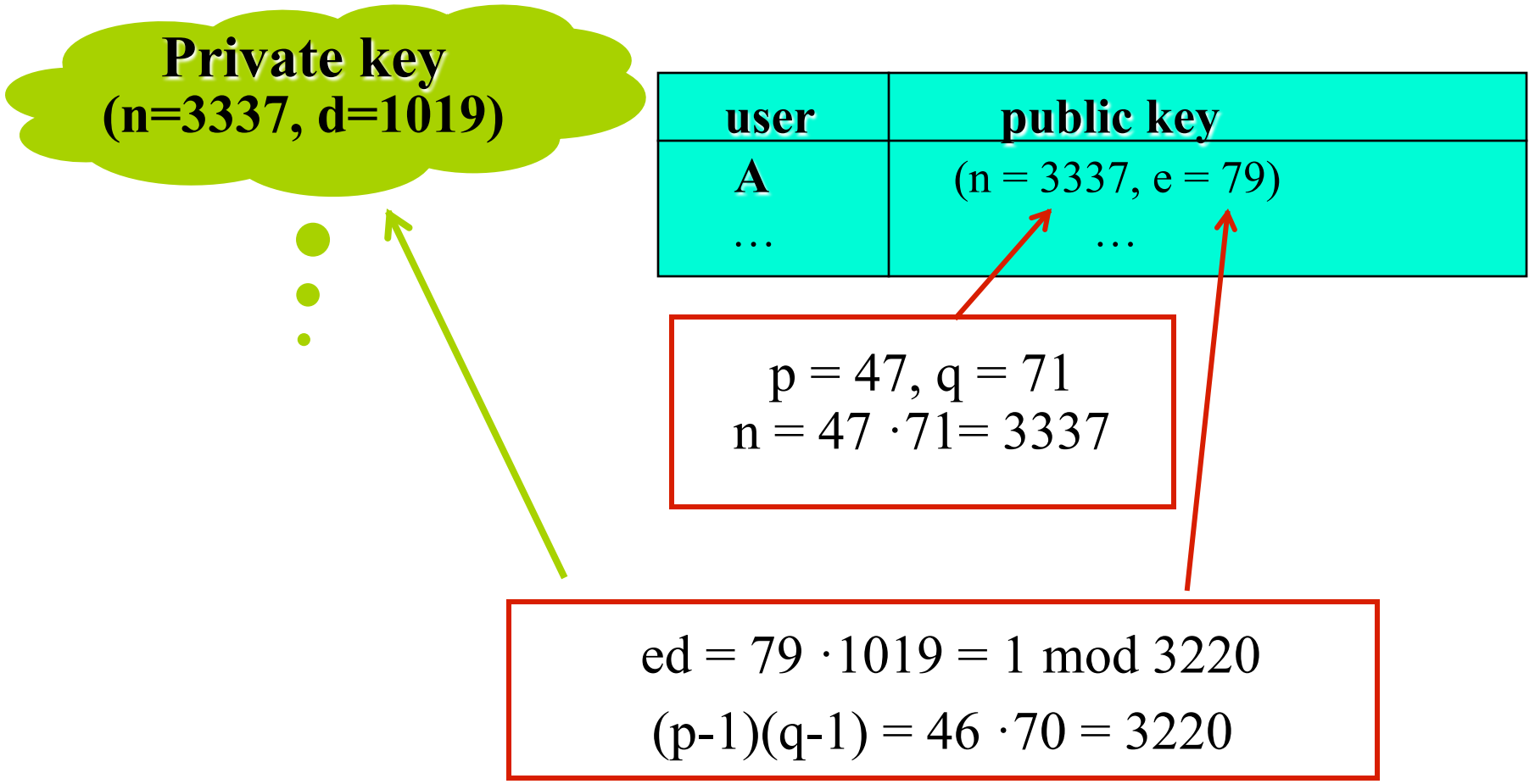
LE CHIAVI SONO INVERTIBILI



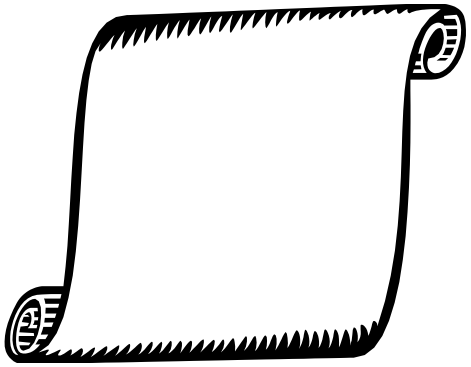
Example: Both

- Alice wants to send Bob message HELLO both enciphered and authenticated (integrity-checked)
  - Alice's keys: public (17, 77); private: 53
  - Bob's keys: public: (37, 77); private: 13
- Alice enciphers HELLO (07 04 11 11 14):
  - $(07^{53} \bmod 77)^{37} \bmod 77 = 07$
  - $(04^{53} \bmod 77)^{37} \bmod 77 = 37$
  - $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
  - $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
  - $(14^{53} \bmod 77)^{37} \bmod 77 = 14$
- Alice sends 07 37 44 44 14

Another example: RSA keys



Another example: RSA keys



user	public key
A	(n = 3337, e = 79)
…	…

ciphering M = 688 to be sent to A

688^{79} mod 3337 → 1570

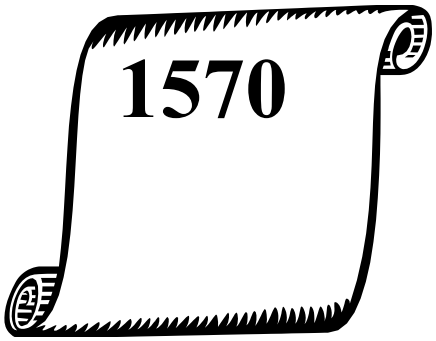
Another example: RSA keys

chiave privata
(n=3337, d=1019)

utente	chiave pubblica
A	(n = 3337, e = 79)
…	…

Deciphering C = 1570

$1570^{1019} \bmod 3337 \rightarrow 688$



Funzione di Eulero

- $\phi(p) = p-1$ se p primo
- $\phi(pq) = (p-1)(q-1)$ se p, q primi
- $\phi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$

  fattorizzazione $n = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$, p_i primo, $e_i \geq 0$
- **Teorema di Eulero**: $x \in Z_n^* \Rightarrow x^{\phi(n)} = 1 \bmod n$

RSA

$$C^d \bmod n = (M^e)^d \bmod n$$
$$= M^{ed} \bmod n$$
$$= M^{1+r(p-1)(q-1)} \bmod n$$
$$= M \bmod n$$
$$= M$$

$ed = 1 \bmod (p-1)(q-1)$

Teorema di Eulero
$x \in Z_n^* \Rightarrow x^{(p-1)(q-1)} = 1 \bmod n$

poichè $0 \leq M < n$

Prova per tutti gli x mediante il teorema del resto cinese

References

- M.Bishop: Cap. 9
- Stallings : cap 3, 6 and 9
- Standards: DES and RSA
- Teoria dei numeri: *Introduzione agli algoritmi e strutture dati*, CAP 31

  T. Cormen et al, Mc Graw Hill

Homework

- Prepare 30 slides to quantitatively illustrate and compare strenght and weakness of 3DES and RSA algorithms
- Deadline: 28 April /5 May