

## **fondamenti di sicurezza**

- requisiti di sicurezza
- crypto
  - glossario
  - cifratura asimmetrica
  - elaborazione messaggio
  - attacchi
- cifratura classica
  - cifratura simmetrica
- cifratura moderna
  - DES
  - cifratura asimmetrica

## **requisiti di sicurezza**

- Autenticità
- Segretezza dei dati
- **Confidenzialità** (controllo degli accessi)
- **Integrità dei dati**
- Non Ripudiabilità
- **Disponibilità**

## **crypto**

### **glossario**

- testo in chiaro (plaintext) - messaggio originale
- algoritmo di cifratura (cipher) – algoritmo che trasforma il testo in chiaro in testo cifrato
- testo cifrato (ciphertext) – messaggio codificato prodotto come output dall'algoritmo di cifratura: dipende dal testo in chiaro e dalla chiave
- chiave – usata come input dell'algoritmo di cifratura, valore indipendente dal testo in chiaro:
  - cifratura simmetrica (una sola chiave)
  - cifratura asimmetrica (chiave pubblica e chiave privata)
- algoritmo di decifratura – algoritmo che trasforma il testo cifrato in testo in chiaro, prende come input la chiave

### **cifratura asimmetrica**

chiave pubblica e privata

**Un messaggio cifrato con una chiave pubblica  
può essere decifrato solo con la corrispondente chiave privata  
e/o viceversa**  
**un messaggio cifrato con una chiave privata  
può essere decifrato solo con la corrispondente chiave pubblica**

segretezza

## + Cifratura asimmetrica: segretezza

16

- Si vuole garantire che solo il destinatario legga il messaggio
  - Bob cifra il messaggio usando la chiave pubblica di Alice (destinatario)
  - Alice decifra il messaggio usando la propria chiave privata

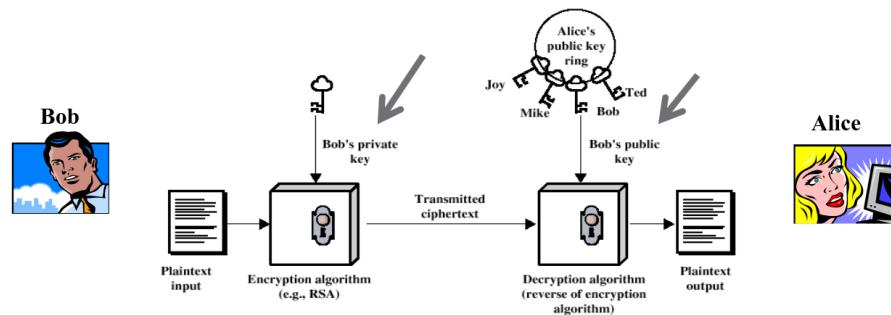


autenticazione

## + Cifratura asimmetrica: autenticazione

17

- Si vuole garantire che il messaggio sia stato creato dal mittente (BoB)
  - Bob cifra il messaggio usando la propria chiave privata
  - Alice decifra il messaggio usando la chiave pubblica di Bob



## elaborazione messaggio

**block cipher** il plaintext viene suddiviso in blocchi di grandezza fissa ed ognuno di questi blocchi viene cifrato. Esempi:

- DES
- AES
- RSA

**stream cipher** viene cifrato un flusso continuo di dati un bit alla volta.

## attacchi

**crittoanalisi** sfrutta le caratteristiche dell'algoritmo e la conoscenza di testi in chiaro/testi cifrati per tentare di individuare la chiave o testo cifrato:

- ciphertext only
- known plaintext
- chosen plaintext

**brute force** si tenta ogni possibile chiave su un messaggio cifrato finché non si riesce ad ottenere una decifratura corretta (testo in chiaro)

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

**incondizionatamente sicuro** Indipendentemente dalla potenza computazionale a disposizione non è possibile risalire alla chiave, in quanto il testo cifrato non contiene informazioni sufficienti per determinare il testo in chiaro

**Computazionalmente sicuro** Non è possibile risalire alla chiave con le risorse computazionali attualmente a disposizione in un tempo utile per l'attacco (ad esempio il tempo necessario è maggiore del tempo d'utilizzo della chiave)

## cifratura classica

## cifratura simmetrica



## requisiti

- deve essere pubblico (la robustezza non deve dipendere dalla segretezza del codice dell'algoritmo)
- deve essere forte (resistente agli attacchi noti)
- Chiave segreta conosciuta solo dal mittente e dal ricevente:
  - $C = E(P, K)$
  - $P = D(C, K)$
- Necessità di un canale sicuro per distribuire la chiave

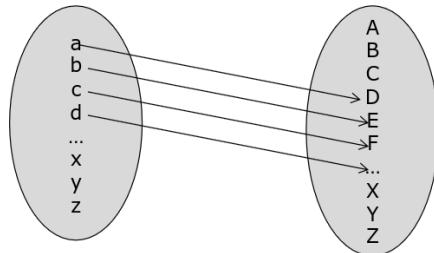
Tutti gli algoritmi di cifratura simmetrica si basano su due semplici operazioni:

- Sostituzioni → Es. BARBARA → EDUEDUD
- Trasposizioni → Es. BARBARA → ARBBARA

## sostituzione

**ceaser cipher** Sostituisce ogni lettera dell'alfabeto con la lettera che si trova a 3 posizioni di distanza

- Il cifrario di Cesare è un esempio di cifratura **monoalfabetica**
- Si utilizza un unico alfabeto (mapping) per le sostituzioni
- La chiave è la funzione sull'alfabeto che associa ad ogni lettera dell'alfabeto plaintext una lettera dell'alfabeto ciphertext.



39

anche se mappiamo ad ogni carattere del plaintext un arbitrario carattere nel ciphertext lo schema generato non è sicuro perché vulnerabile all'attacco delle frequenze.

- **Le sostituzioni monoalfabetiche non cambiano la frequenza relativa delle lettere**
- Si possono fare attacchi basati sul calcolo della frequenza delle lettere nel testo cifrato
- **Alternativa:** sostituire la stessa lettera con **più sostituti (omofoni)**, assegnati a rotazione o casualmente
  - se il numero di omofoni associati ad ogni lettera è proporzionale alla sua frequenza, allora si potrebbe celare la frequenza della singola lettera.

poligrammi playfair

- Il testo in chiaro è sostituito due lettere alla volta (digramma)
- Ogni lettera ha più sostituti (omofoni), scelti in base al digramma d'appartenenza

- Algoritmo utilizza una matrice 5X5 di lettere:
  - Si inserisce la parola chiave (senza duplicati) da sinistra a destra e dall'alto verso il basso
  - Si riempie il resto della matrice con le rimanenti lettere dell'alfabeto
- Esempio usando la parola chiave MONARCHY

<b>M</b>	<b>O</b>	<b>N</b>	<b>A</b>	<b>R</b>
<b>C</b>	<b>H</b>	<b>Y</b>	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Il testo in chiaro è sostituito due lettere alla volta:

- Se entrambe le lettere cadono nella **stessa riga**, si sostituiscono con le lettere che seguono a destra:  
esempio "ar" è cifrato come "RM"
- Se entrambe le lettere cadono nella **stessa colonna**, si sostituiscono con le lettere sottostanti:  
esempio "mu" è cifrato con "CM"

<b>M</b>	<b>O</b>	<b>N</b>	<b>A</b>	<b>R</b>
<b>C</b>	<b>H</b>	<b>Y</b>	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Il testo in chiaro è sostituito due lettere alla volta:

- Se le lettere **non** sono nella **stessa riga/colonna**, ciascuna lettere viene sostituita con la lettera che si trova sulla stessa riga e nella colonna occupata dall'altra lettera in chiaro (**vertici della sottomatrice**):  
esempio "hs" è cifrato con "BP", e "ea" con "IM" o "JM"
- La coppia di lettere doppie è tradotta separatamente aggiungendo una lettera di riempimento (ad. Esempio x):  
esempio parola ballon viene trattata come ba lx lo on

<b>M</b>	<b>O</b>	<b>N</b>	<b>A</b>	<b>R</b>
<b>C</b>	<b>H</b>	<b>Y</b>	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

- Ogni lettera ha più sostituti (omofoni), scelti in base al digramma d'appartenenza (analisi della frequenza della singola lettera è più difficile)
  - AB -> BI o BJ
  - AO -> RN
  - AP -> OS
  - AQ -> NS
  - AL -> MS
  - ....
- La sicurezza è migliorata in quanto 26 lettere ci porta alla definizione di  $26 \times 26 = 676$  digrammi (individuazione singolo digramma più difficile)
- È stato considerato per lungo tempo molto sicuro (adottato dall'esercito inglese durante la prima guerra mondiale)
- ...ma mantiene ancora diverse informazioni sulla struttura del testo

## hill cipher

- Cifrario di Hill è un cifrario basato sull'algebra lineare

- Si assegna ad ogni lettera un valore numerico:

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



- Si sostituiscono n-grammi, modellati come vettori di n elementi

- Es. n=3, plaintext p= 'cat'  $p = \begin{pmatrix} 2 \\ 0 \\ 19 \end{pmatrix}$

- La chiave K è definita come una matrice invertibile di dimensione n x n. Es  $K = \begin{pmatrix} 2 & 4 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{pmatrix}$

- La cifratura è il prodotto matriciale di  $K \times p$

$$\begin{pmatrix} 2 & 4 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 19 \end{pmatrix} = \begin{pmatrix} 2 \cdot 2 + 4 \cdot 0 + 1 \cdot 19 \\ 1 \cdot 2 + 3 \cdot 0 + 2 \cdot 19 \\ 1 \cdot 2 + 0 \cdot 0 + 0 \cdot 19 \end{pmatrix} = \begin{pmatrix} 4 + 0 + 19 \\ 2 + 0 + 38 \\ 2 + 0 + 0 \end{pmatrix} = \begin{pmatrix} 23 \\ 40 \\ 2 \end{pmatrix} \text{ mod } 26. \quad c = \begin{pmatrix} 23 \\ 14 \\ 2 \end{pmatrix}$$

- La decifratura è il prodotto matriciale di  $K^{-1} \times c$

$$\begin{pmatrix} 0 & 0 & 1 \\ 2 & -1 & -3 \\ 5 & 5 & 5 \\ -3 & 4 & 2 \\ 5 & 5 & 5 \end{pmatrix} \cdot \begin{pmatrix} 23 \\ 40 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \cdot 23 + 0 \cdot 40 + 1 \cdot 2 \\ \frac{2}{5} \cdot 23 + \frac{-1}{5} \cdot 40 + \frac{-3}{5} \cdot 2 \\ \frac{-3}{5} \cdot 23 + \frac{4}{5} \cdot 40 + \frac{2}{5} \cdot 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 19 \end{pmatrix}$$

$K^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 2 & -1 & -3 \\ 5 & 5 & 5 \\ -3 & 4 & 2 \\ 5 & 5 & 5 \end{pmatrix}$

## polialfabetico

- Idea base: si utilizzano diverse sostituzioni monoalfabetiche, in modo che ogni lettera del plaintext sia cifrata con un cifrario monoalfabetico diverso

- Ogni cifratura polialfabetica specifica:

- Un insieme di sostituzioni monoalfabetiche
- Come utilizzare la chiave per determinare quale sostituzione applicare

- Esempio di cifratura polialfabetica più noto: cifratura di Vigenère

## vigenere cipher

### ■ Vigenère cipher:

- Come insieme di sostituzioni monoalfabetiche si considerano 26 cifrari di Cesare (ottenuti con  $K=0,1\dots 25$ )
  - Regola: L'n-esima lettera della chiave indica quale alfabeto bisogna utilizzare per sostituire l'n-esima lettera del plaintext
- 
- La chiave deve essere lunga quanto il plaintext.
  - La si ottiene ripetendo la chiave finché non è della lunghezza desiderata.

Key: test

testtesttesttesttesttes

Plaintext:

meetmeafterthetogaparty

Ciphertext:

**FIWMFISYMIJMAILHZEHTKXQ**

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	
a	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
b	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	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	W	
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	

## vigenere weakness

la chiave è ripetuta, quindi si riutilizzano più volte le sostituzioni monoalfabetiche corrispondenti alle lettere ripetute della chiave (analogo a many time pad)

### one time pad

- L'unico cifrario per cui è possibile dimostrare la sua sicurezza incondizionata.

- Utilizzo l'operatore XOR:

- proprietà XOR:

- $(A \text{ XOR } B) \text{ XOR } B = A$

A	B	$A \dot{\vee} B$
0	0	0
0	1	1
1	0	1
1	1	0

- Le lettere del plaintext (bits) sono combinate in XOR con le corrispondenti lettere (bits) della chiave

- Cifratura:  $P \text{ XOR } K = C$

- Decifratura:  $C \text{ XOR } K = P$

- La chiave è lunga quanto il plaintext

- L'unico cifrario per cui è possibile dimostrare la sua sicurezza incondizionata.

- *Cifrario perfetto* - Teorema di Shannon. Claude E. Shannon, *A Mathematical Theory of Communication*, *Bell System Technical Journal* (July 1948)

- Ipotesi: **la chiave deve essere casuale e lunga quanto il messaggio da cifrare**

- La casualità della chiave garantisce che il ciphertext non mantenga informazioni sulle correlazioni interne presenti nel plaintext (i.e., frequenze di lettere, digrammi, etc.)

- Shannon dimostrò che questo tipo di cifrario è inviolabile anche da attacchi a forza bruta:

- Ad un ciphertext c possono corrispondere più plaintext m, utilizzando chiavi diverse

**drawbacks** → Generazione di chiave casuale AND Distribuzione della chiave, lunga quanto il messaggio

### trasposizione

- permutazione delle lettere nel testo in chiaro
- esse non vengono modificate
- la transposizione non cambia la frequenza delle lettere utilizzate

rail fence (staccionata)

- Il testo in chiaro viene scritto come una sequenza di diagonali e poi letto come una sequenza di righe

- **Messaggio:** meet me after the toga party

- Profondità due:

```
m e m a t r h t g p r y  
e t e f e t e o a a t
```

- Testo cifrato:

MEMATRHTGPRYETEFETEOAAT

## cifratura moderna

### DES

#### characteristics

- block size → 64 bits
- key size → 64 bits but 56 bits really 8 bits for parity checking
- number of rounds → 16
- *IP* → permutation look up table on plaintext (Initial Permutation)
- *FP* → permutation look up table on output of 16 rounds (Final Permutation)
- *FP* is the inverse of *IP*
- *F* → feistel function (4 stages):
  1. *E* → Expansion. the 32 bit half block is expanded to 48 bit padding each 4 bit block with its neighbour bits,

#### E BIT-SELECTION TABLE

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

2. Key mixing →  $E \oplus keys[round]$ . *keys* is 48 bits generated from key schedule function.
3. Substitution. the 48 bit output of key mixing is taken 6 bit at a time and run through the s-box. The 8 s-boxes are lookup table that map

6 bits to 4 bits output.

4.  $P$  Permutation  $\rightarrow$  permutation on the 32 bit output of the substitution. this is designed so that the bits of the output of previous round s-box are spread across different s-boxes in the next round.

**P**

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

- key schedule  $\rightarrow$  the key is reduced to 56 bits using PC1 (permuted choice). bits 8, 16, 24.. 64 are discarded or used for parity checking. These 56 bits are divided in two halves and left shifted according to a certain amount for each round (1 or 2). Then the two halves are concatenated back and reduced to 48 bits using PC2.



Figure 1: des overview



Figure 2: des feistel



Figure 3: des feistel

### cifratura asimmetrica

obiettivi principali:

- distribuzione delle chiavi più efficiente rispetto alla crittografia simmetrica
- garantisce che un messaggio è stato prodotto realmente dal mittente

!

## Cifratura asimmetrica: autenticazione

- Se Bob vuole inviare un messaggio per cui è possibile provare l'autenticità
  - cifra il messaggio con la sua chiave privata
- Quando Alice riceve il messaggio,
  - lo decifra con la chiave pubblica di Bob, verificando quindi che è stato creato da lui e che non sono state fatte modifiche

## Cifratura asimmetrica: segretezza

- Se Bob vuole inviare un messaggio riservato ad Alice,
  - cifra il messaggio con la chiave pubblica di Alice
- Quando Alice riceve il messaggio
  - decifra il messaggio con la sua chiave privata

## Cifratura asimmetrica: autenticazione e segretezza

- Se Bob vuole inviare un messaggio segreto e autenticabile ad Alice,
  - Bob **cifra** il messaggio una prima volta con la **propria chiave privata**, e
  - **cifra** il risultato una seconda volta con la **chiave pubblica di Alice**
- Quando Alice riceve il messaggio
  - lo **decifra** prima la **propria chiave privata** e
  - poi con la **chiave pubblica di Bob**

$$C = m^n e \bmod k$$

**BOSCH**  
Tecnologia per la vita

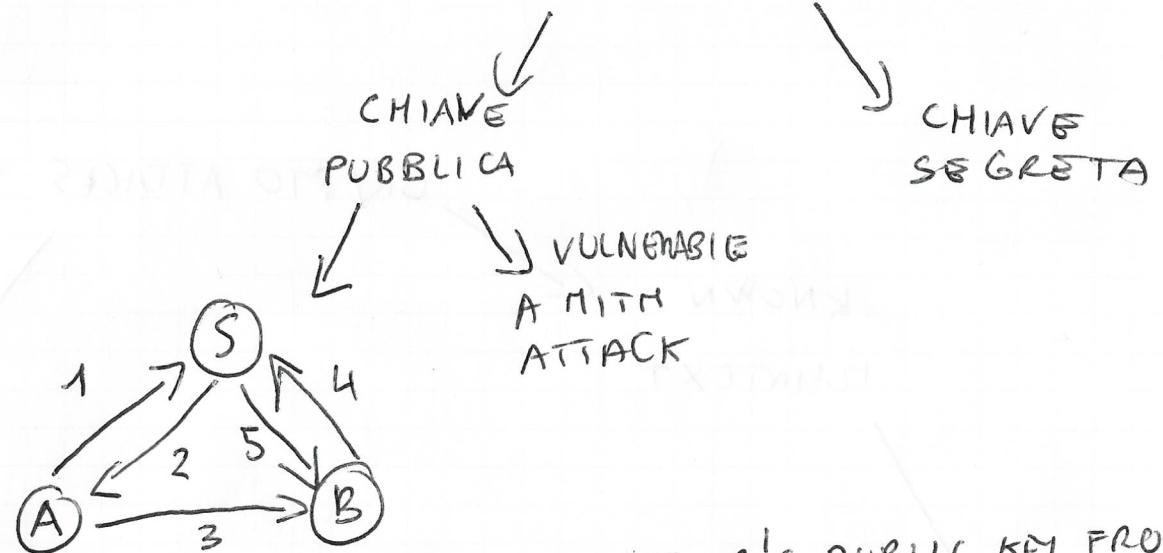


LESS BITS  
MORE  
SECURE





- ⑥ B → A:  $\{N_A, N_B\} K_{PA}$
- ⑦ A → B:  $\{N_B\} K_{PB}$



- ① A → S:  $ID_A, ID_S$  // A REQUESTS B'S PUBLIC KEY FROM S
- ② S → A:  $\{K_{PB}, ID_B\} KR_S$  // S RESPONDS WITH B'S PUBLIC KEY AND B'S IDENTITY, SIGNED WITH SERVER PRIVATE KEY
- ③ A → B:  $\{N_A, A\} K_{PB}$  // A MANDA UN NONCE A B
- ④ B → S:  $ID_B, ID_A$  // B REQUESTS A'S PUBLIC KEY FROM S
- ⑤ S → B:  $\{K_{PA}, ID_A\} KR_S$  // S RESPONDS WITH A'S PUBLIC KEY AND A'S IDENTITY, SIGNED

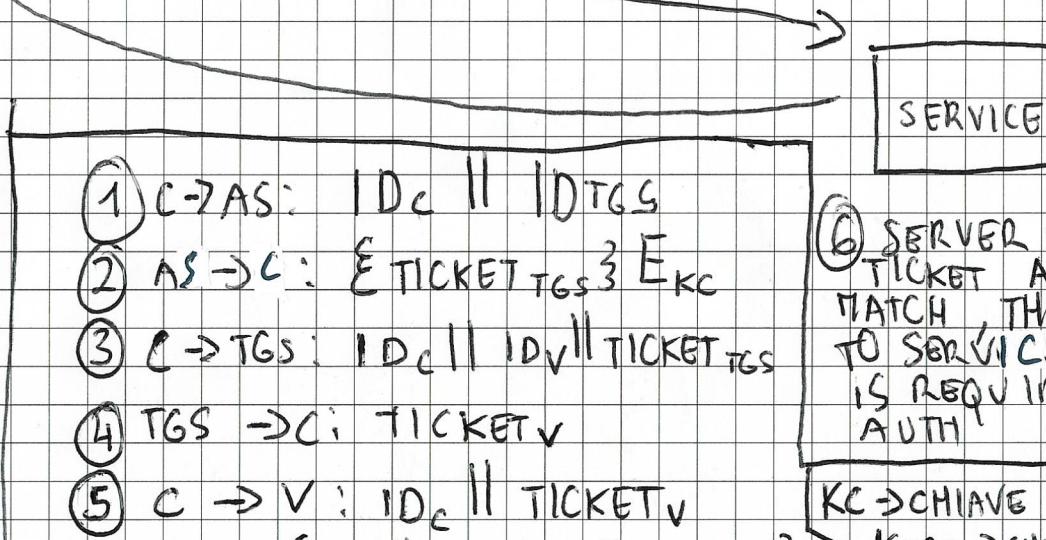
# KERBEROS

② AS VERIFIES USER'S ACCESS RIGHT IN DATABASE, CREATES TICKET GRANTING TICKET AND SESSION KEY. RESULTS ARE ENCRYPTED USING KKEY DERIVED FROM USER'S PASSWORD.



③ WORKSTATION PROMPTS USER FOR PASSWORD AND USES PASSWORD TO DECRYPT INCOMING MESSAGES THEN SENDS TICKET AND AUTH THAT CONTAINS USERNAME, NETWORK ADDRESS, AND TIME TO TGS.

④ TGS DECRYPTS TICKET AND AUTHENTICATOR, VERIFIES REQUEST AND CREATES TICKET FOR REQUESTED SERVERS

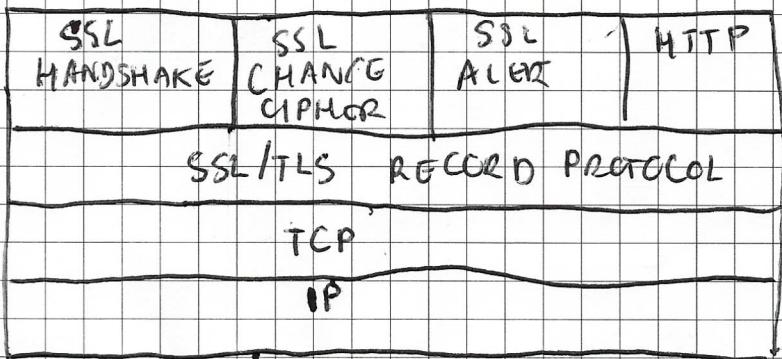


⑥ SERVER VERIFIES THAT TICKET AND AUTHENTICATOR MATCH, THEN GRANTS ACCESS TO SERVICE. IF MUTUAL AUTH IS REQUIRED SERVER RETURNS AUTH

$$TICKET_{TGS} = \{ ID_c || AD_c || ID_{TGS} || TS1 || LIFETIME || 3E_KTGS \}$$

$$TICKET_v = \{ ID_c || AD_c || TS2 || LIFETIME || 3E_KV \}$$

KC -> CHIAVE CONDIVISA TGS E AS  
 KTGS -> CHIAVE CONDIVISA TGS E AS  
 KV -> " " " "  
 TGS E SERVER



RISERVATEZZA → SIMMETRICAL KEY IN HANDSHAKE ↗  
 INTEGRITÀ / AUTH ↘  
 UTILIZZA UN MAC CON CHIAVE SEGRETA ↘

NOTA:  
DUE CHIAVI DIVERSE ↗

## TLS RECORD

PROTOCOLLO

APPLICATION DATA

FRAGMENT

COMPRESS

ADD MAC

ENCRYPT

APPEND SSL  
HEADER

ALERT  
HANDSHAKE  
CHANGE\_CIPHER\_SPEC

CHIAVE  
MAC

SEQUENCE

MAC (MAC\_WRITE\_KEY, SEQ-NUM || TLS COMPRESSED\_TYPE ||

TLS COMPRESSED\_VERSION || TLS COMPRESSED\_LENGTH || TLS (decompressed)

1.2, 1.3

LUNGHEZZA  
DEL FRAGMENTO

FRAGMENT

FRAGMENTO

TLS ALERT → 2 BYTE

1° BYTE

1 → WARNING

2 → FATAL

2° BYTE → SPECIFICA IL MESSAGGIO  
DI ALERT

- FATAL: UNEXPECTED MSG, BAD RECORD MAC, DECOMPRESSION FAILURE, HANDSHAKE FAIL, ...
- WARNING: CLOSE\_NOTIFY

NO CERTIFICATE, BAD  
CERTIFICATE,

TLS CHANGE CIPHER SPEC  
L'1 MSG CON 1 BYTE A VALORE 1.  
CHIUSA LA FASE DI HANDSHAKE

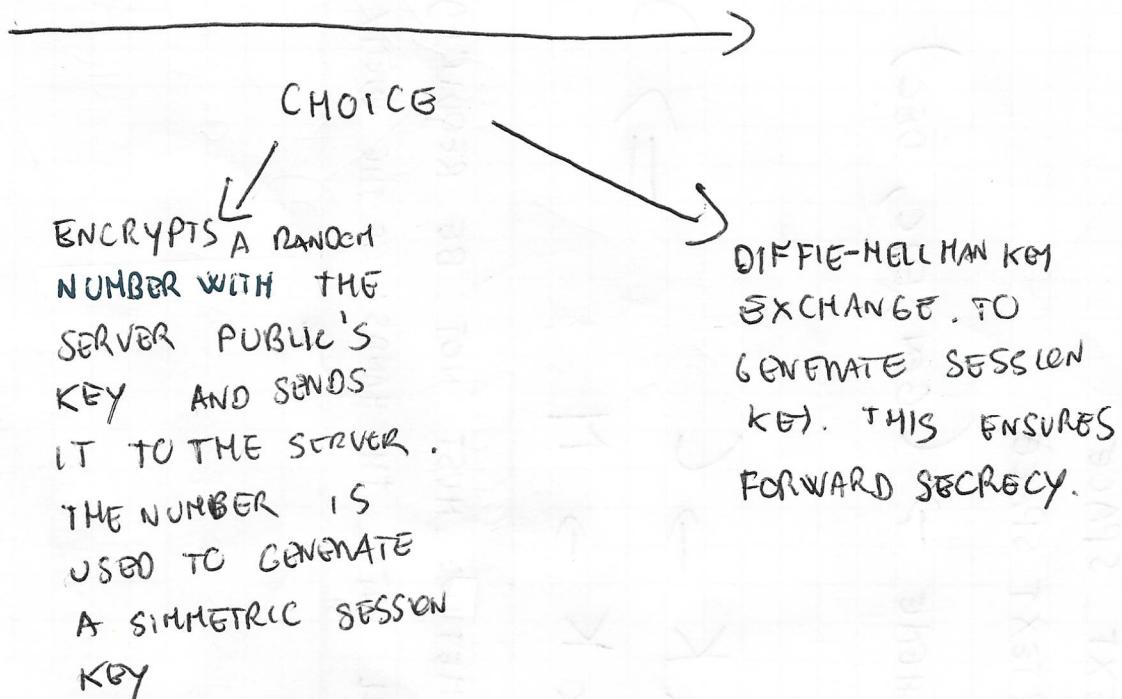


**BOSCH**  
Tecnologia per la vita

# TLS HANDSHAKE



CLIENT  
VERIFIES  
CERTIFICATE  
IS VALID



CHANGE-CIPHER-SPEC →

FINISHED

CHANGE-CIPHER-SPEC ←

FINISHED

# Configuraz. di HP Digital Filing

---

## Prerequisiti:

1. Verificare che il computer e la stampante siano connessi alla rete.
2. Per la configurazione di Scansione a e-mail OPPURE Scansione su cartella, verificare di disporre di una connessione Internet attiva.

## Opzione 1: Impostare l'archiviaz. digitale HP con il sw HP

Seguire le istruzioni relative al sistema operativo del computer:

### Windows

1. Sul computer, fare doppio clic sull'icona della stampante sul desktop.
2. Fare clic su Stampa, Scansione, Fax e selezionare il link appropriato (Config. guidata Scansione a cartella di rete oppure Config. guidata Scansione a e-mail).
3. Seguire le istruzioni a video.

### macOS

1. Fare clic sull'icona HP Utility nel Dock.  
NOTA: se l'icona non compare nel Dock, fare clic sull'icona Spotlight a destra della barra dei menu, digitare HP Utility, quindi fare clic sull'opzione HP Utility.
2. Clic su [Scansione a e-mail]/[Scansione su cartella di rete] in [Conf. scansione] e seguire le istruzioni.

---

## Opzione 2: Configurazione via Configurazione remota

L'archiviazione digitale HP può essere configurata in remoto da qualunque computer della rete utilizzando il server Web incorporato della stampante:

1. Digitare il seguente indirizzo nel browser Web del computer.  
<https://192.168.0.35>
2. Fare clic su Scansione, selezionare il link appropriato (Configurazione cartella rete o Configurazione scansione a e-mail), seguire le istruzioni.

# NETWORK LEVEL SECURITY

## PROBLEMS

LETTURA  
PACCHETTI IN  
TRANSITO

MODIFICA  
PACCHETTI IN  
TRANSITO

IP SEC  
OBLIGATORIO  
IN IPV6

POSSIBILE IN  
IPV4

AUTH

GESTIONE  
CHIAVI  
RISERVATIZZA

INTESTAZIONE  
DI ESTENSIONE

AH

AUTENTICATION  
HEADER

ESP  
ENCAPSULATING  
SECURITY  
HEADER

GATEWAY 2  
GATEWAY

MOODALITÀ

TUNNEL



TRANSPORTO



IL PACCHETTO ORIGINARIO È  
TRATTATO COME PAYLOAD DI  
UN ALTRO PACCHETTO IP.

SE C'È ESP NESSUN ROUTER È IN GRADO DI  
VEDERE IL CONTENUTO



**BOSCH**

Tecnologia per la vita

AH E ESP

UTILIZZANO  
CRYPTO

CHAVI  
SIMETRICHE

AS  
ASSOCIAZIONE DI  
SICUREZZA

SPD

SECURITY  
POLICY  
DATABASE

SELETTORI  
TRAFFICO  
- IP DEST  
- IP SRC  
- USR ID  
- PORT SRC  
- PORT DEST

AZIENDE  
- ALLOW  
- PROTECT  
- DISCARD

SECURITY  
PARAMETERS  
INDEX

PRIMARY  
KEY

SAD  
SECURITY  
ASSOCIATION  
DATABASE

( SPI, IP DESTINAZIONE, IDENTIFICATORE  
PROTOCOLLO )

AH OR  
ESP

PER GENERARE AS  
DUE HOST SI ACCORDANO  
SUI PARAMETRI

MANUALE

INTERNET  
AUTOMATICO  $\rightarrow$  IKE  $\rightarrow$  KEY EXCHANGE

- ISAKMP : INTERNET SECURITY ASSOCIATION  
KEY MANAGEMENT PROTOCOL : CRUD DI SA

- OAKLEY : SCAMBIO DI CHAVI BASATO  
SU DH



**BOSCH**  
Tecnologia per la vita

ADVANCING WINDOW IF VALID  
PACKET TO THE RIGHT IS RECEIVED



## AUTHENTICATION HEADER

INTEGRITÀ  
ED AUTH

SLIDING WINDOW

IP  
PACKET

PART OF  
IP HEADER

TRANSMISSION  
MAC

I PACCHETTI IP SONO GARANTITI  
IN ORDINE

ANTI  
REPLAY

LE DUE PARTI DIVERSE  
CONDIVIDONO UNA  
K

ANTI REPLAY

SA E COUNTER 32-BIT

HEADER AH COUNTER++

QUANDO COUNTER =  $2^{32}$

NEW SA



SECURITY PARAMETERS INDEX → SA  
(SPI)

SEQUENCE NUMBER

AUTHENTICATION DATA  
(VARIABLE)

→ MAC

NO TUNNEL

IP V4 → NEW IP  
HDR AH ORIG IP  
TCP DATA

IP V6 → NEW IP  
HDR EXT HEAD AH

CALCOLATO SU

CAMPIONI  
INTESTAZIONE  
IP IMMUTABILI

ORIG IP  
HDR EXT HEAD TCP DATA

PAYLOAD

IP PACKET

TUNNEL TRANSPORT AH INTESAZIONE

IPV4 → ORIG IP  
HDR AH TCP DATA

IPVS → ORIG IP  
HDR EXT HEAD AH DEST TCP

BOSCH DATA  
Tecnologia per la vita



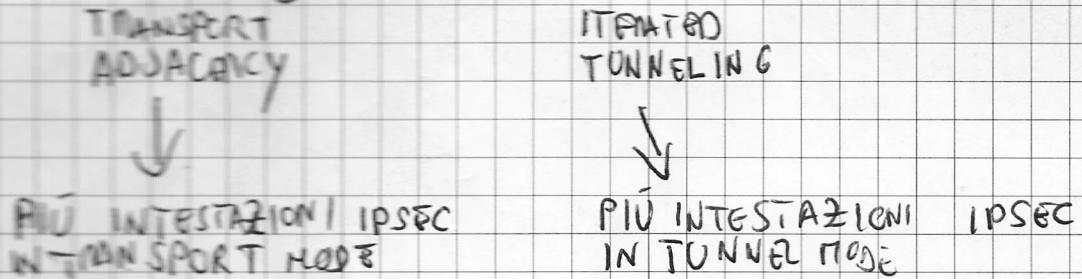


**BOSCH**

Tecnologia per la vita

# COMBINAZIONI

DI SA



- ESP CON OPZIONE AUTH

- TRANSPORT MODE: AUTH E CIFRATURA AL PAYLOAD IP

- TUNNEL MODE: AUTH E CIFRATURA ALL'INTERO IP PACKET

→ - AUTENTICAZIONE AL CIPHERTEXT

- BUNDLE TRANSPORT ADJACENCY

- 2 SA MODALITÀ TRASPORTO

- ESP SENZA AUTH AL IP PACKET ORIGINALE

- AUTHENTICATION HEADER CON TRANSPORT MODE AL PACCHETTO PRECEDENTE

- VANTAGGIO RISPETTO AD ESP CON AUTH: AUTH ON IP HEADER

-

→ BUNDLE TRANSPORT TUNNEL: DUE SA IN MODALITÀ TRASPORTO E TUNNEL

- 1 SI APPLICA AH IN TRANSPORT MODE ALL'IP PACKET OG

- 2 SI APPLICA ESP CON TUNNEL MODE AL PACKET 1

- PROS: DUE ON ESP AUTH: AUTH ON IP HEADER