

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

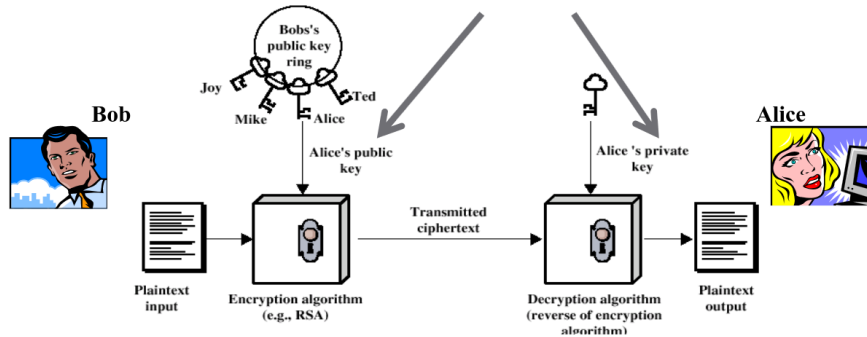
chiave pubblica e privata

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

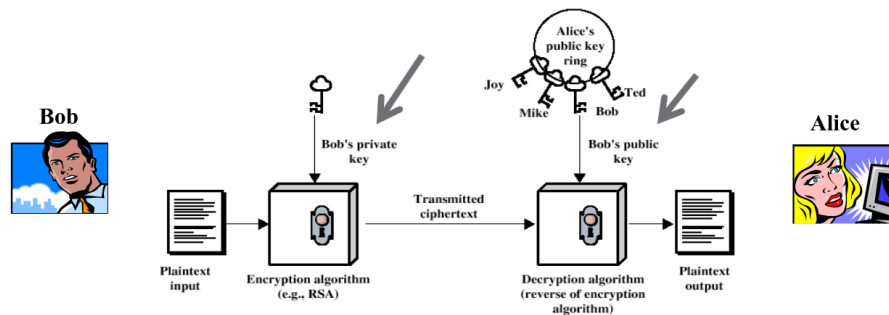
- Si vuole garantire che solo il destinatario legga il messaggio
 - Bob cifra il messaggio usando la chiave pubblica di Alice (destinatario)
 - Alice decifra il messaggio usando la propria chiave privata



autenticazione

+ Cifratura asimmetrica: autenticazione

- 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)

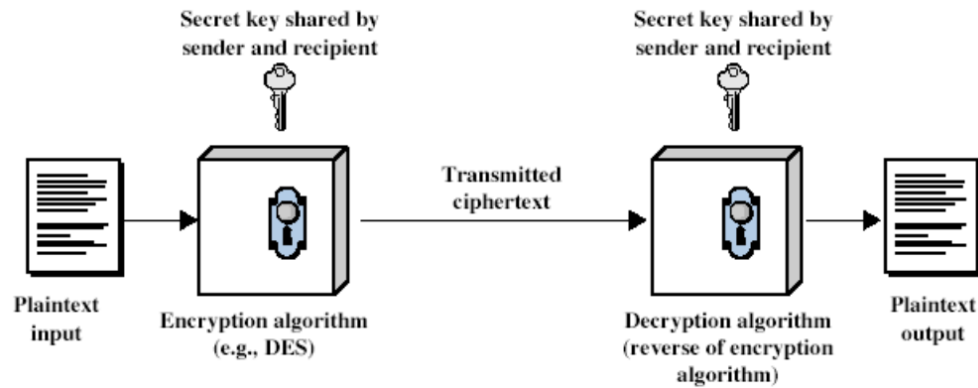
Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/ μ s	Time required at 10^6 decryptions/ μ s
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu$ s = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu$ s = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu$ s = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu$ 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

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

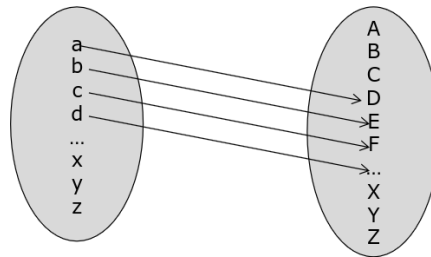
Tutti gli algoritmi di cifratura simmetrica si basano su due semplici operazioni:

- 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

M	O	N	A	R
C	H	Y	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"

M	O	N	A	R
C	H	Y	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 lettera 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

M	O	N	A	R
C	H	Y	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 x p

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

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

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

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

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	B
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	C
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	D
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	E
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	F
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	G
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	H
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	I
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	J
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	K
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	L
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	M
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	N
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	O
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	P
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	Q
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	R
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	S
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	T
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	U
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	V
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	V	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	X
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	Y

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.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- Utilizzo l'operatore XOR:

- proprietà XOR:

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

- 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 trasposizione 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 $\rightarrow$ 64 bits
- key size $\rightarrow$ 64 bits but 56 bits really 8 bits for parity checking
- number of rounds $\rightarrow$ 16
- $IP \rightarrow$ permutation look up table on plaintext (Initial Permutation)
- $FP \rightarrow$ permutation look up table on output of 16 rounds (Final Permutation)
- FP is the inverse of IP
- $F \rightarrow$ feistel function (4 stages):
 1. $E \rightarrow$ 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 $\rightarrow E \oplus keys[round]$. $keys$ 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 → 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 → 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.

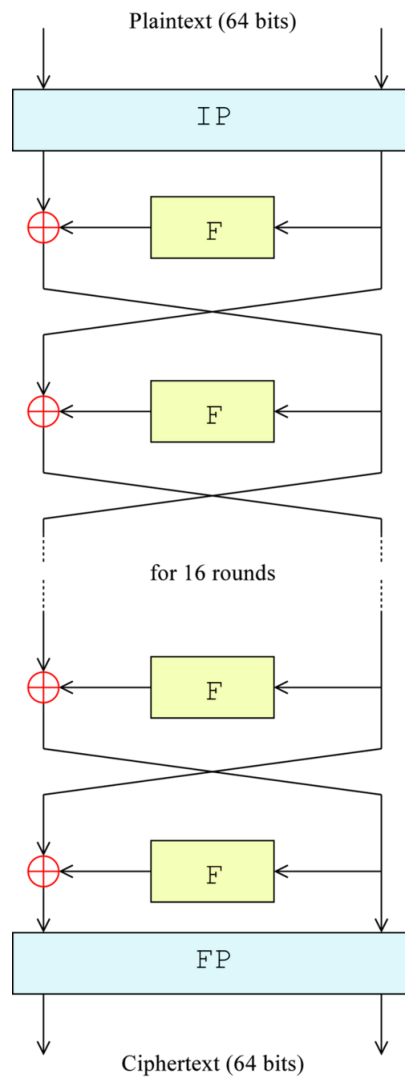


Figure 1: des overview

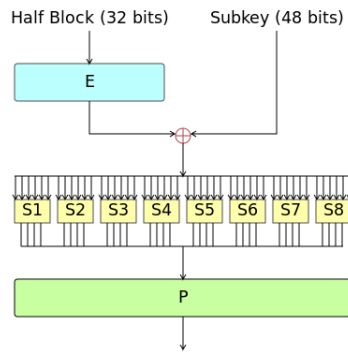


Figure 2: des feistel

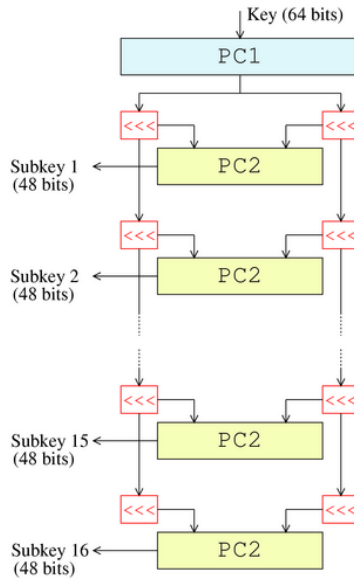


Figure 3: des feistel

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

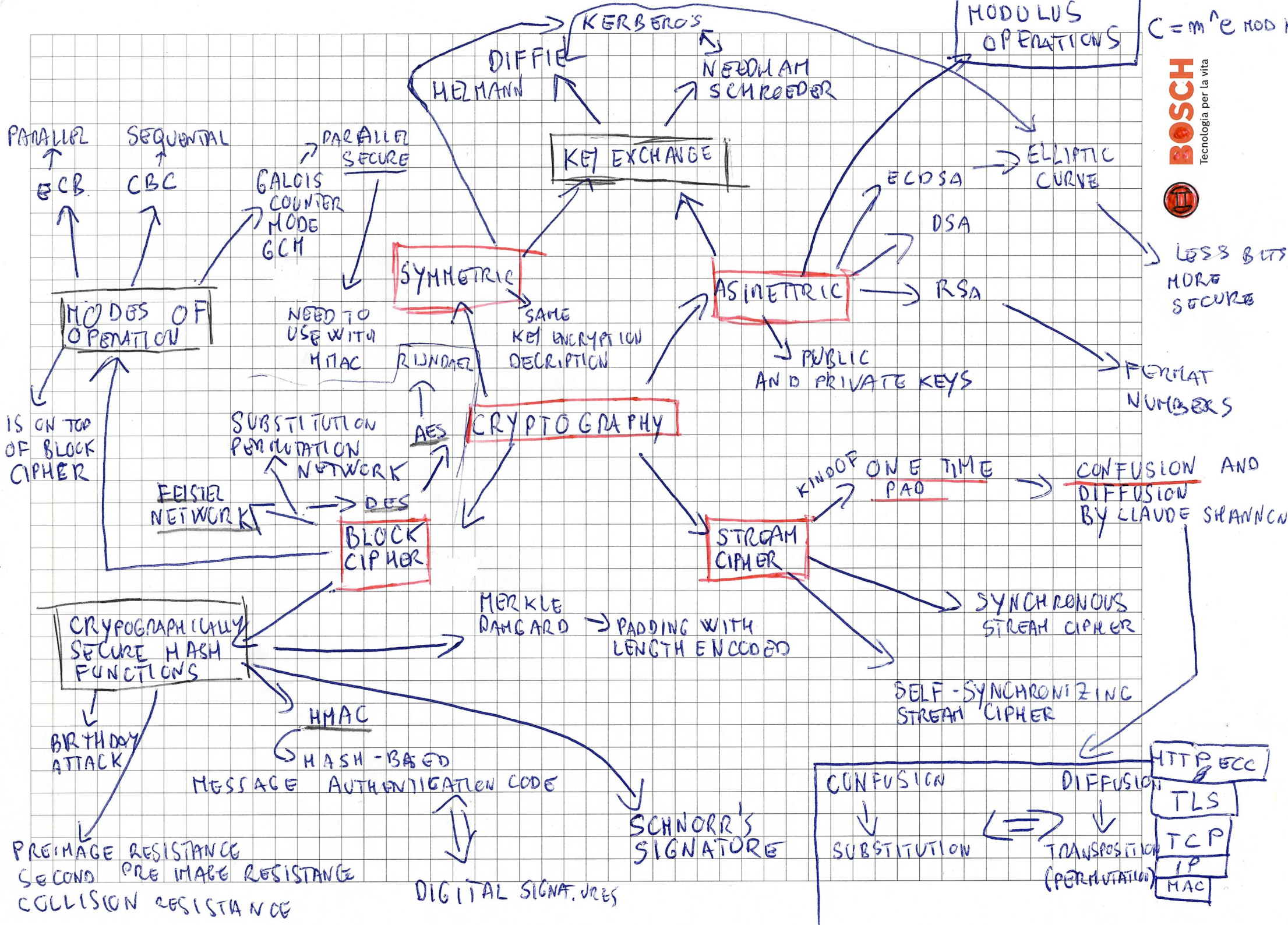
Cifratura asimmetrica: segretezza

- 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**

$$C = m^e \text{ MOD } K$$



KEY MANAGEMENT

KEY AGREEMENT



DIFFIE-HELLMAN

KEY TRANSPORT

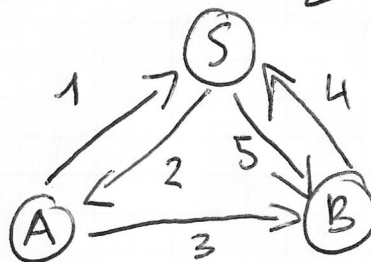


NEEDHAM-SCHROEDER

CHIAVE PUBBLICA

CHIAVE SEGRETA

VULNERABILE
A MITM
ATTACK



- ⑥ $B \rightarrow A: \{N_A, N_B\} K_{P_A}$
 ⑦ $A \rightarrow B: \{N_B\} K_{P_B}$

- ① $A \rightarrow S: ID_A, ID_B$ // A REQUESTS B'S PUBLIC KEY FROM S
 ② $S \rightarrow A: \{K_{P_B}, ID_B\} K_{R_S}$ // S RESPONDS WITH B'S PUBLIC KEY AND B'S IDENTITY, SIGNED WITH SERVER PRIVATE KEY
 ③ $A \rightarrow B: \{N_A, A\} K_{P_B}$ // A MANDA UN NONCE A B
 ④ $B \rightarrow S: ID_B, ID_A$ // B REQUESTS A'S PUBLIC KEY FROM S
 ⑤ $S \rightarrow B: \{K_{P_A}, ID_A\} K_{R_S}$ // 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 KEY DERIVED FROM USER'S PASSWORD.

① USER LOGIN AND REQUESTS A SERVICE

USER

REQUEST GRANTING TICKET

TICKET + SESSION KEY

REQUEST SERVICE GRANTING TICKET

TICKET + SESSION KEY

KERBEROS 3-DWG

AUTHENTICATION SERVER



TICKET GRANTING SERVER

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

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.

⑤ WORKSTATION SENDS TICKET AND AUTH TO SERVER.

SERVICE

① C → AS: $ID_C || ID_{TGS}$

② AS → C: $\{TICKET_{TGS}\}_{E_{KC}}$

③ C → TGS: $ID_C || ID_V || TICKET_{TGS}$

④ TGS → C: $TICKET_V$

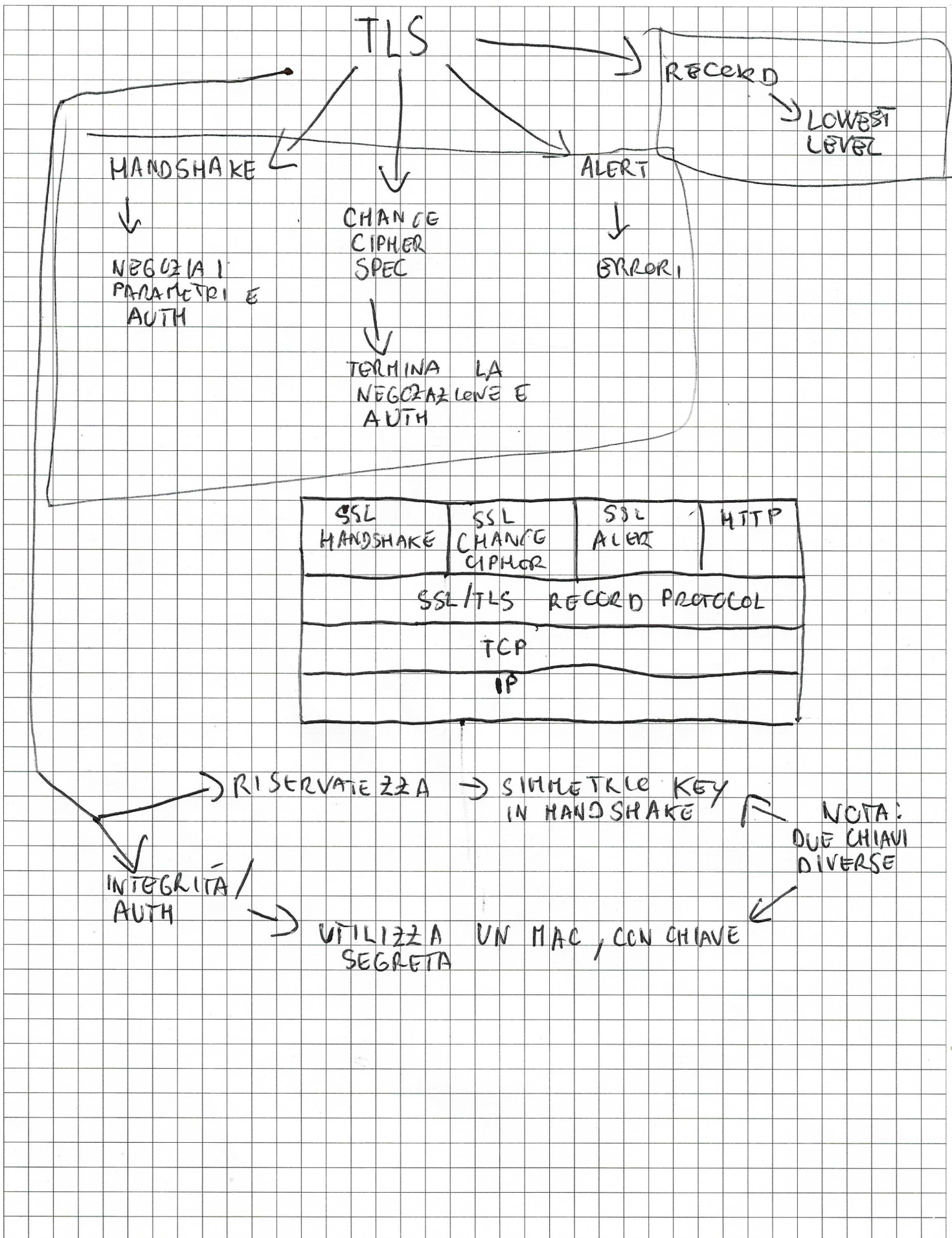
⑤ C → V: $ID_C || TICKET_V$

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

$TICKET_V = \{ID_C || AD_C || TS || LIFETIME\}_{E_{KV}}$

$K_C \rightarrow CHAVE COND VISA TPA$ CLIENT & AS
 $K_{TGS} \rightarrow CHAVE COND VISA$ TGS & AS
 $K_V \rightarrow " "$ TGS & SERVER

$TICKET_{TGS} = \{ID_C || AD_C || ID_{TGS} || TS || LIFETIME\}_{E_{KTGS}}$



TLS RECORD

PROTOCOL

APPLICATION DATA

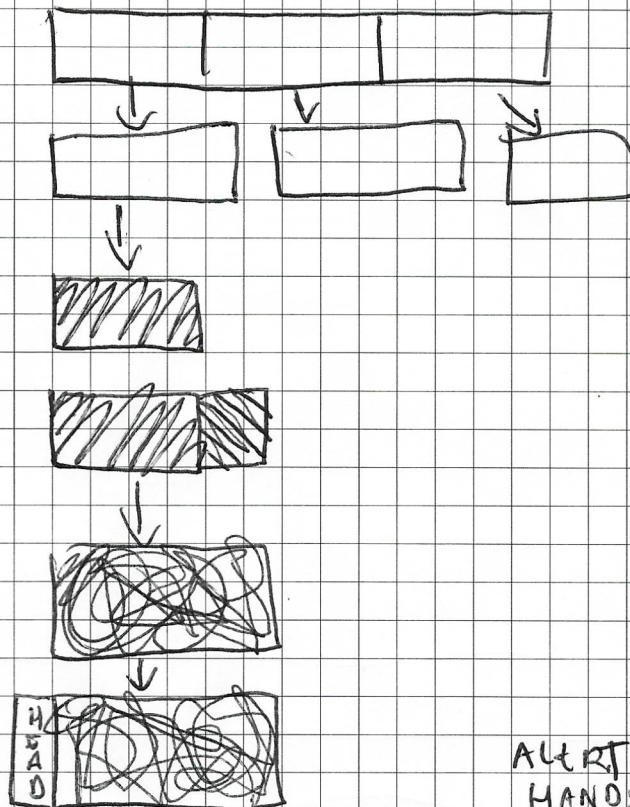
FRAGMENT

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 COMPRESSED)

1.2, 1.3

LUNGERIA
DEL FRAGMENTO

FRAGMENT

FRAGMENTO

TLS ALERT → 2 BYTE

1° BYTE

1 → WARNING
2 → FATAL

2° BYTE

SPECIFICA IL MESSAGGIO
DI ALERT

- FATAL: UNEXPECTED HEX, BAD RECORD MAC, DECOMPRESSION FAILURE, HANDSHAKE FAIL, NO CERTIFICATE, BAD CERTIFICATE,
- WARNING: CLOSE NOTIFY

TLS CHANGE CIPHER SPEC
↳ 1 MSG CON 1 BYTE A VALORE 1.
CHIUDE LA FASE DI HANDSHAKE



BOSCH
Tecnologia per la vita

TLS HANDSHAKE

CLIENT → SERVER

CLIENT HELLO

- CIPHER SUITE (CIPHERS AND HASH)

← SERVER

SERVER HELLO

- PICKS CIPHER AND HASH
- SENDS THE CERTIFICATE
 - SERVER NAME
 - CA
 - SERVER PUBLIC KEY

CLIENT
VERIFIES
CERTIFICATE
IS VALID

→

CHOICE

↙

ENCRYPTS A RANDOM NUMBER WITH THE SERVER PUBLIC'S KEY AND SENDS IT TO THE SERVER. THE NUMBER IS USED TO GENERATE A SYMMETRIC SESSION KEY

↘

DIFFIE-HELLMAN KEY EXCHANGE TO GENERATE SESSION KEY. THIS ENSURES FORWARD SECRECY.

→

CHANGE - CIPHER - SPEC

FINISHED

→

←

CHANGE - CIPHER - SPEC

←

FINISHED

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

IPSEC

OBBLIGATORIO
IN IPV6

POSSIBILE IN
IPV4

AUTH

RISERVAZIONE

GESTIONE
CHIAVI

INTEGRAZIONE
DI ESTENSIONE

AH
AUTHENTICATION
HEADER

ESP
ENCAPSULATING
SECURITY
HEADER

GATEWAY 2
GATEWAY

MODALITÀ

END 2 END TRA DUE HOST

TRASPORTO

TUNNEL

IP HEADER (GATEWAY)	IPSEC HEADER
IP HEADER	TCP/ UDP

IP HEADER	TCP/UDP	
IP HEADER	IP SEC HEADER	TCP/UDP

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

CHIAVI
SIMMETRICHE

AS
ASSOCIAZIONE DI
SICUREZZA

SPD

SECURITY
POLICY
DATABASE

SAD
SECURITY
ASSOCIATION
DATABASE

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

AZIONE
- ALLOW
- PROTECT
- DISCARD

(SPI, IP DESTINAZIONE, IDENTIFICAZIONE PROTOCOLLO)
SECURITY
PARAMETERS
INDEX

PRIMARY
KEY

AH OR
ESP

PER GENERARE AS
DUE HOST SI ACCORDANO
SUI PARAMETRI

MANUALE

AUTOMATICO → IKE → INTERNET
KEY EXCHANGE

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

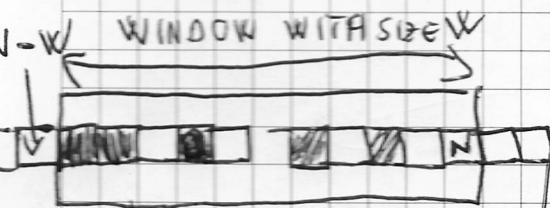
- COAKLEY: SCAMBIO DI CHIAVI BASATO
SU DH



BOSCH

Tecnologia per la vita

ADVANCING WINDOW IF VALID
PACKET TO THE RIGHT IS
RECEIVED



AUTHENTICATION HEADER

INTEGRITY
ED AUTH

SLIDING WINDOW

IP
PACKET

PART OF
IP HEADER

TRANSMIT
MAC

LE DUE PARTI DEVONO
CONDIVIDERE UNA
K

IPACCHETTI IP NON
SONO GARANTITI
IN ORDINE

ANTI
REPLAY

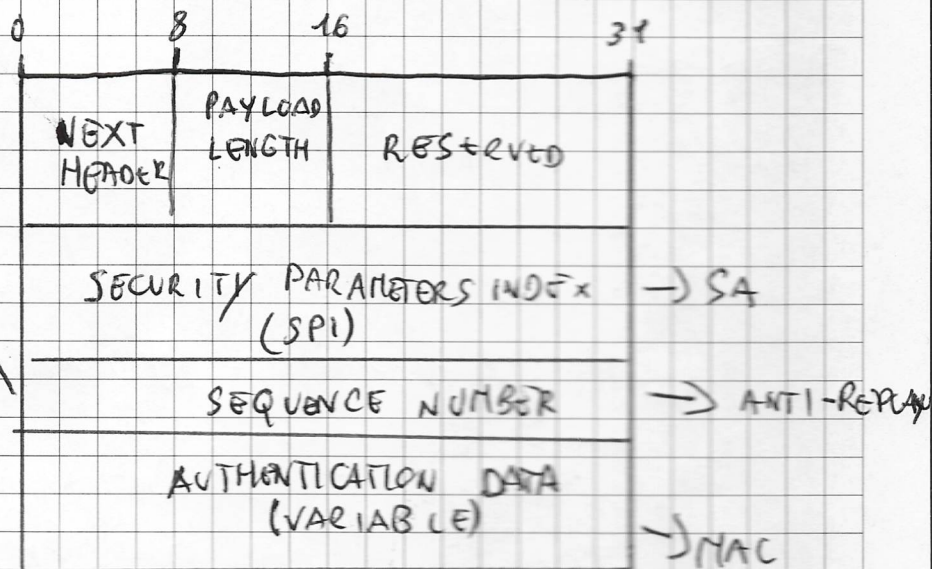
ANTI REPLAY

SA | COUNTER 32-BIT

HEADER AH COUNTERH

QUANDO COUNTER = 2^{32}

NEW SA



NO TUNNEL

IPV4 → NEW IP HDR | AH | ORIG IP HDR | TCP DATA

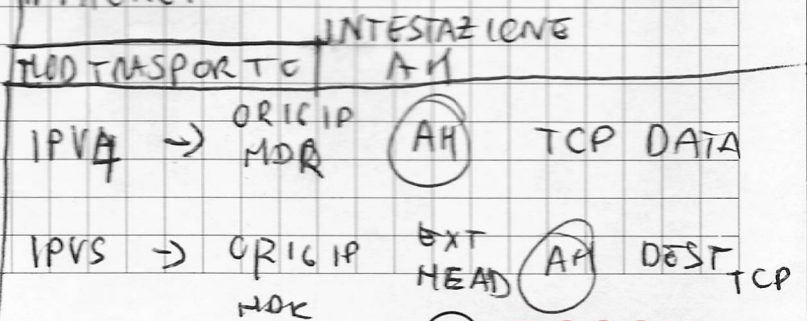
IPV6 → NEW IP HDR | EXT HEAD | AH

ORIG IP HDR | EXT HEAD | TCP DATA

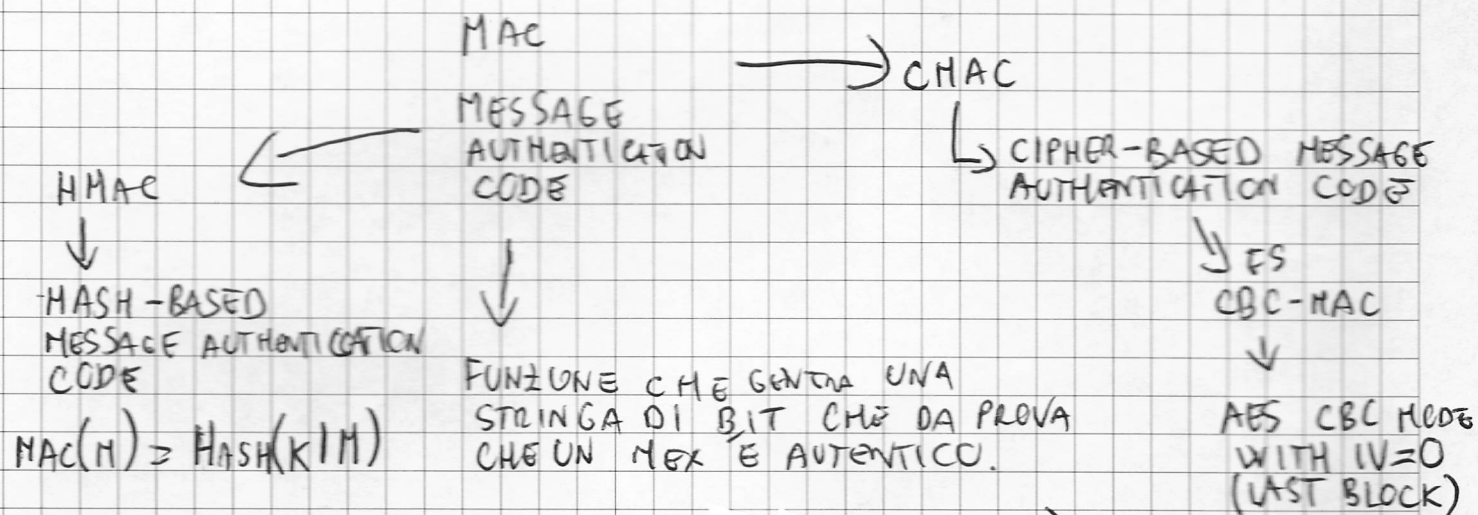
CALCOLATO SU

PAYLOAD
IP PACKET

CAMPI
INTERAZIONE
IP INMUTABILI



BOSCH DATA
Tecnologia per la vita



$MAC(M) = \{K \oplus M\}$
 ↓
 A E B CONDIVIDONO K
 A SPEDISCE A B
 $MEX = M || MAC(M) = M || \{K \oplus M\}$

B RICEVE IL MEX E COMPUTA
 $MAC(M)$. SE $MAC(M)$ È UGUALE
 AL MAC RICEVUTO DA A ALLORA
 IL MESSAGGIO È AUTENTICO E NON
 È STATO MODIFICATO





COMBINAZIONE

DI SA

TRANSPORT
ADJACENCY

ITERATED
TUNNELING

PIÙ INTESTAZIONI IPSEC
IN TRANSPORT MODE

PIÙ INTESTAZIONI IPSEC
IN TUNNEL MODE

- ESP CON OPZIONE AUTH

- TRANSPORT MODE: AUTH E CRIPTATURA AL PAYLOAD IP

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

~~NO~~ - 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 OC

- 2 SI APPLICA ESP CON TUNNEL MODE AL PACKET 1

- PRES: ~~NO~~ ON ESP AUTH: AUTH ON IP HEADER



BOSCH

Tecnologia per la vita