EVENTO

Linux Day 2025

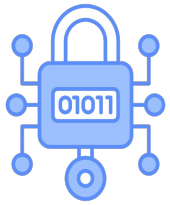
AUTORE

Pietro Terracciano

GRAZIE A

IrLUG
Irpinia Linux User Group

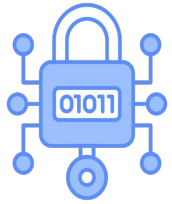
lumaca IRPINA



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro tra client-server ed end-to-end

1. Introduzione
   Cos'è la crittografia? - Privacy, autenticazione, integrità e sicurezza dei dati La macchina Enigma - Codice binario e sistemi di codifica - Come avviene la cifratura di un dato?
2. Crittografia simmetrica
   Cos'è la crittografia simmetrica? - Algoritmi di crittografia simmetrica Crittografia simmetrica client-server - Crittografia simmetrica end-to-end Cifratura simmetrica super sicura
3. Crittografia asimmetrica
   Cos'è la crittografia asimmetrica? - Algoritmi di crittografia asimmetrica Perchè si preferisce non usare la crittografia asimmetrica per l'end-to-end? Cifratura asimmetrica super sicura
4. Certificati ed Authority
   Cos'è un certificato? Cos'è un Authority?



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

1. Introduzione

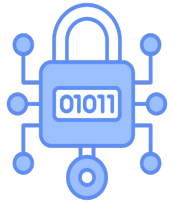
- ☐ Cos’è la crittografia?
- ☐ Privacy, autenticazione, integrità e sicurezza dei dati
- ☐ La macchina Enigma
- ☐ Codice binario e sistemi di codifica
- ☐ Come avviene la cifratura di un dato?

Crittografia
è il processo di codifica dei dati per renderli incomprensibili a utenti non autorizzati

Crittografia
utilizza diverse tipologie di algoritmi matematici altamente performanti

AUTORE

Pietro Terracciano



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

1. Introduzione

- ☑ Cos’è la crittografia?
- ☐ Privacy, autenticazione, integrità e sicurezza dei dati
- ☐ La macchina Enigma
- ☐ Codice binario e sistemi di codifica
- ☐ Come avviene la cifratura di un dato?

AUTORE

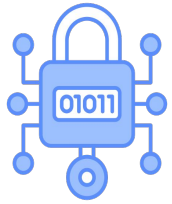
Pietro Terracciano

Privacy (segretezza)
i dati sono cifrati, anche se qualcuno li intercetta non può leggerli

Autenticazione
abbiamo effettuato l’accesso al sito della banca, il sito genera un Token che identifica in maniera univoca il nostro profilo

Integrità
i dati sono cifrati e “firmati”, anche se qualcuno prova a cambiare un bit, l’intera informazione viene scartata

Sicurezza dei dati (massiva)
tutti i vincoli precedenti aggiungendo controllo degli accessi e policy



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

1. Introduzione

- ☑ Cos’è la crittografia?
- ☑ Privacy, autenticazione, integrità e sicurezza dei dati
- ☐ La macchina Enigma
- ☐ Codice binario e sistemi di codifica
- ☐ Come avviene la cifratura di un dato?

AUTORE

Pietro Terracciano

La macchina Enigma
dispositivo elettromeccanico per la cifratura di messaggi

La macchina Enigma
utilizzato dalla Germania nazista durante la seconda guerra mondiale. Cuore del sistema di comunicazione segreto del Terzo Reich

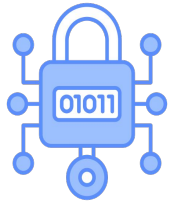
La macchina Enigma
ogni giorno venivano cambiato le impostazioni meccaniche: rotori, ordine, posizione iniziale e plugboard (chiave privata)

La macchina Enigma
offriva oltre 10^23 combinazioni

La macchina Enigma
fu trovato un metodo per la decifratura da parte di Alan Turing

INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro tra client-server ed end-to-end



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

1. Introduzione

- ☑ Cos’è la crittografia?
- ☑ Privacy, autenticazione, integrità e sicurezza dei dati
- ☑ La macchina Enigma
- ☐ Codice binario e sistemi di codifica
- ☐ Come avviene la cifratura di un dato?

AUTORE

Pietro Terracciano

Codice binario
0 - zero - spento
1 - uno - acceso

Sistemi di codifica
servono a dare un “senso” logico al nostro “Codice binario”

Sistemi di codifica
riguarda come una informazione viene rappresentata

Sistemi di codifica
ASCII, UTF8, UTF16, …

Sistemi di codifica
bisogna stare attenti alla perdita di informazioni.
Esempio se passiamo da UTF8 ad ASCII

INTRODUZIONE ALLA CRITTOGRAFIA

0 011 0000	A 100 0001	N 100 1110	a 110 0001	n 110 1110	. 010 1110
1 011 0001	B 100 0010	O 100 1111	b 110 0010	o 110 1111	, 010 1100
2 011 0010	C 100 0011	P 101 0000	c 110 0011	p 111 0000	! 010 0001
3 011 0011	D 100 0100	Q 101 0001	d 110 0100	q 111 0001	? 011 1111
4 011 0100	E 100 0101	R 101 0010	e 110 0101	r 111 0010	' 010 0111
5 011 0101	F 100 0110	S 101 0011	f 110 0110	s 111 0011	(010 1000
6 011 0110	G 100 0111	T 101 0100	g 110 0111	t 111 0100	) 010 1001
7 011 0111	H 100 1000	U 101 0101	h 110 1000	u 111 0101	- 010 1101
8 011 1000	I 100 1001	V 101 0110	i 110 1001	v 111 0110	" 010 0010
9 011 1001	J 100 1010	W 101 0111	j 110 1010	w 111 0111	space 010 0000
	K 100 1011	X 101 1000	k 110 1011	x 111 1000	
	L 100 1100	Y 101 1001	l 110 1100	y 111 1001	
	M 100 1101	Z 101 1010	m 110 1101	z 111 1010	

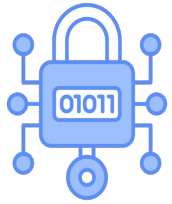
INTRODUZIONE ALLA CRITTOGRAFIA

Decimale	Binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Decimale	Binario
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

Char	Unicode	
H	00000000 01001000	English characters with 1 byte (Fill high bit with 0)
e	00000000 01100101	
l	00000000 01101100	
l	00000000 01101100	
o	00000000 01101111	
算	01111011 10010111	Chinese characters with 2 bytes
法	01101100 11010101	



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

1. Introduzione

- ✅ Cos’è la crittografia?
- ✅ Privacy, autenticazione, integrità e sicurezza dei dati
- ✅ La macchina Enigma
- ✅ Codice binario e sistemi di codifica
- ☐ Come avviene la cifratura di un dato?

AUTORE

Pietro Terracciano

Cifratura di un dato
il dato viene convertito in byte

Cifratura di un dato
viene stabilito un blocco formato da N byte

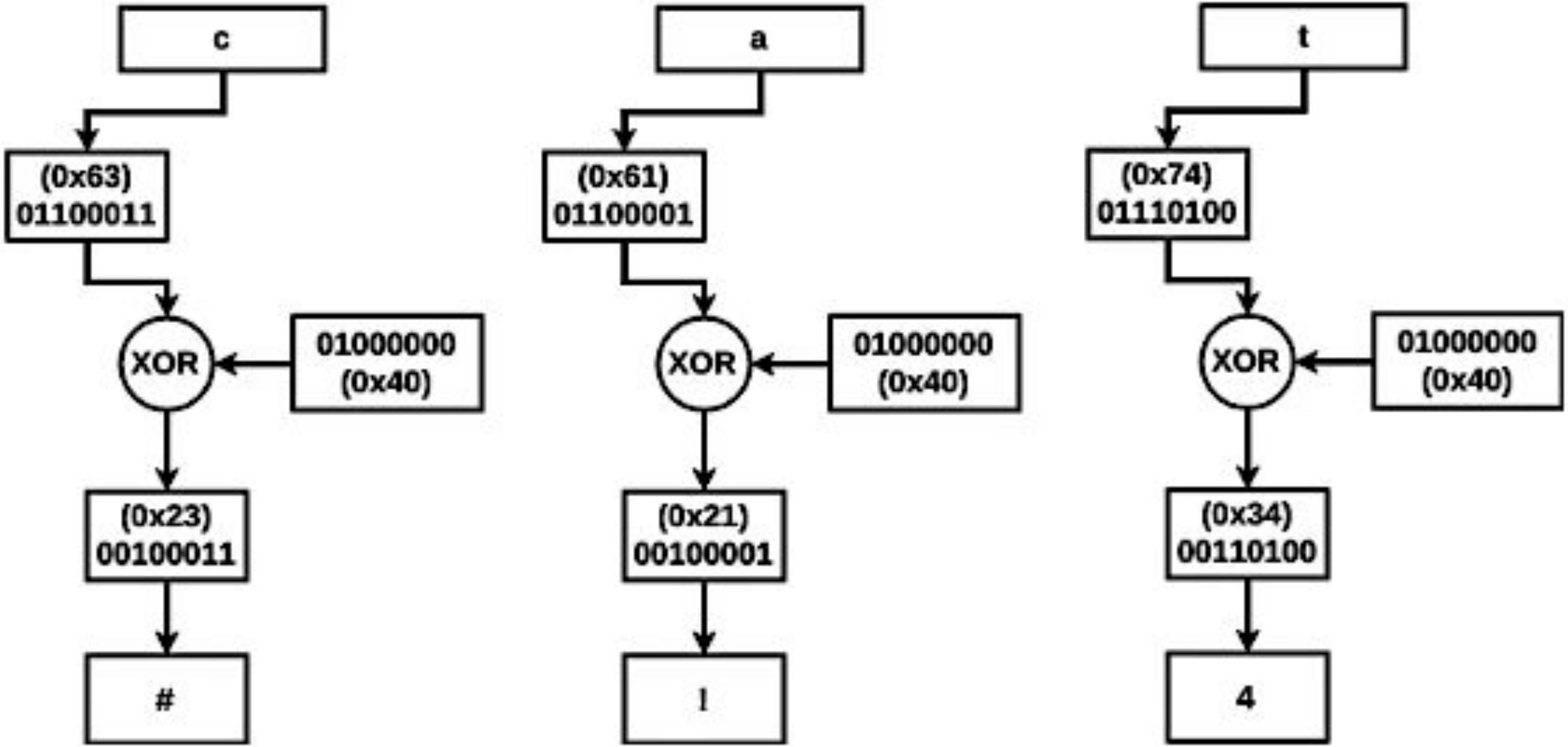
Cifratura di un dato
su ogni blocco di N byte vengono effettuate trasformazioni matematiche : sostituzioni, scambi, addizioni, rimozioni, etc.
Ottenendo un nuovo blocco di N+M byte sporco

Cifratura di un dato
l’insieme di tutti i nuovi blocchi ottenuti compongono un CypherText

INTRODUZIONE ALLA CRITTOGRAFIA

```
# Stringa di esempio
TESTO="Questo è un messaggio segreto"
echo "[+] Testo originale: $TESTO"

# Conversione in byte (hex dump)
echo -n "$TESTO" | xxd
# (-n serve a non aggiungere newline)

# Salva la stringa in un file binario
echo -n "$TESTO" > messaggio.bin

# Genera una chiave AES da 32 byte (256 bit)
openssl rand -out chiave.bin 32

# Cifra il file con AES-256-CBC
openssl enc -aes-256-cbc -salt -in messaggio.bin -out messaggio.enc -pass file:chiave.bin
echo "[+] File cifrato: messaggio.enc"

# Decifra per verifica
openssl enc -d -aes-256-cbc -in messaggio.enc -out messaggio_decifrato.bin -pass file:chiave.bin
echo "[+] File decifrato:"
cat messaggio_decifrato.bin
```

INTRODUZIONE ALLA CRITTOGRAFIA



INTRODUZIONE ALLA CRITTOGRAFIA

Original Message **abcd**

Encryption Key **69**

Encrypted Message **$'&!**

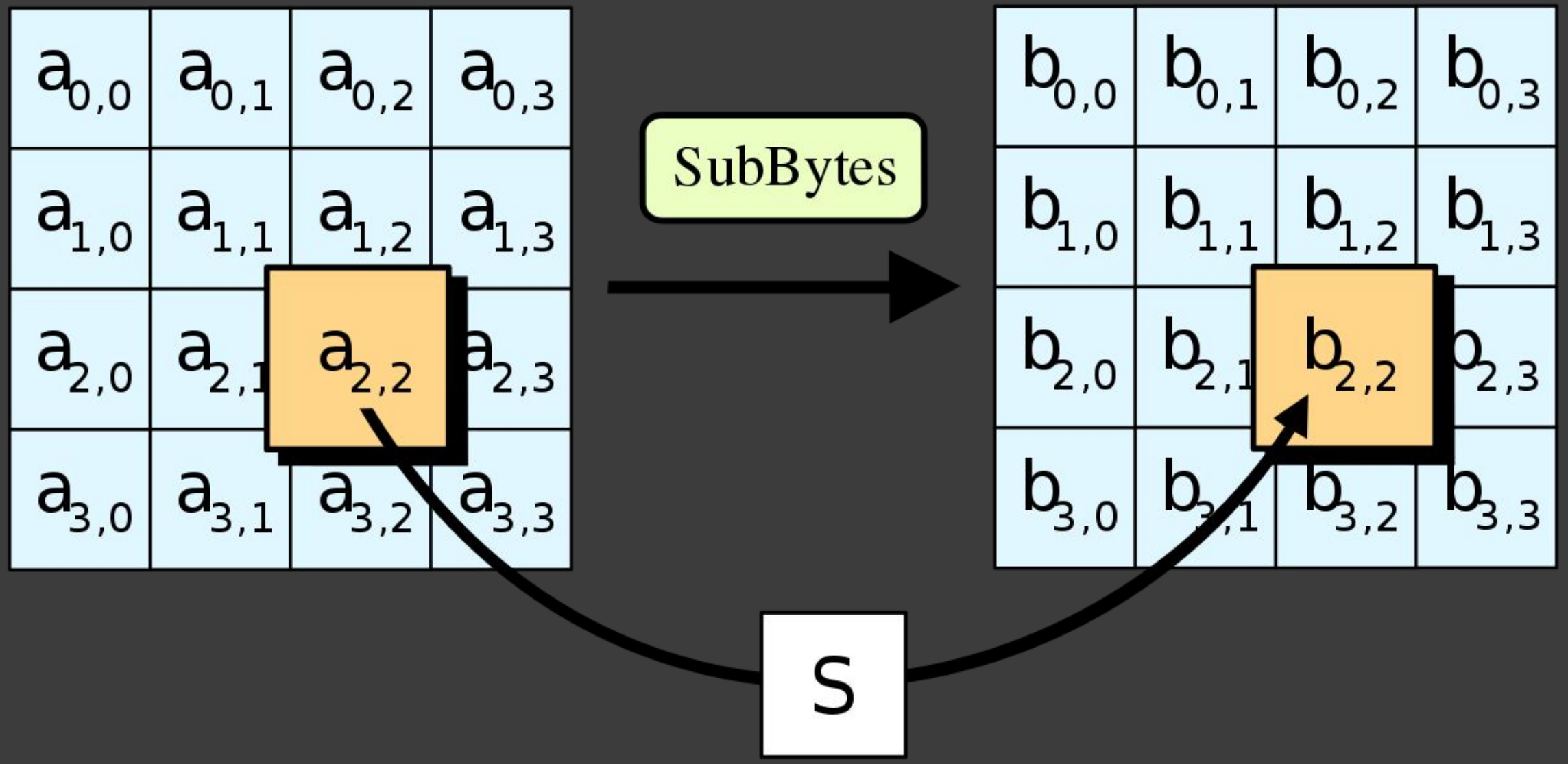
Character	Byte		Encrypted Byte	Encrypted Character
a	97	⊕ 69 →	36	$
b	98	⊕ 69 →	39	'
c	99	⊕ 69 →	38	&
d	100	⊕ 69 →	33	!

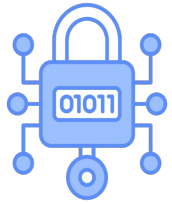
XOR

Single-byte XOR Cipher


Round 2
Round 3
Round 4
Round 5
Round 6
After SubBytes
49 45 7f 77
de db 39 02
d2 96 87 53
89 f1 1a 3b
ac ef 13 45
73 c1 b5 23
cf 11 d6 5a
7b df b5 b8
52 85 e3 f6
50 a4 11 cf
2f 5e c8 6a
28 d7 07 94
e1 e8 35 97
4f fb c8 6c
d2 fb 96 ae
9b ba 53 7c
a1 78 10 4c
63 4f e8 d5
a8 29 3d 03
fc df 23 fe
After ShiftRows
49 45 7f 77
db 39 02 de
87 53 d2 96
3b 89 f1 1a
ac ef 13 45
c1 b5 23 73
d6 5a cf 11
b8 7b df b5
52 85 e3 f6
a4 11 cf 50
c8 6a 2f 5e
94 28 d7 07
e1 e8 35 97
fb c8 6c 4f
96 ae d2 fb
7c 9b ba 53
a1 78 10 4c
4f e8 d5 63
3d 03 a8 29
fe fc df 23
After MixColumns
58 1b db 1b
4d 4b e7 6b
ca 5a ca b0
f1 ac a8 e5
75 20 53 bb
ec 0b c0 25
09 63 cf d0
93 33 7c dc
0f 60 6f 5e
d6 31 c0 b3
da 38 10 13
a9 bf 6b 01
25 bd b6 4c
d1 11 3a 4c
a9 d1 33 c0
ad 68 8e b0
4b 2c 33 37
86 4a 9d d2
8d 89 f4 18
6d 80 e8 d8
Round Key
f2 7a 59 73
c2 96 35 59
95 b9 80 f6
f2 43 7a 7f
3d 47 1e 6d
80 16 23 7a
47 fe 7e 88
7d 3e 44 3b
ef a8 b6 db
44 52 71 0b
a5 5b 25 ad
41 7f 3b 00
d4 7c ca 11
d1 83 f2 f9
c6 9d b8 15
f8 87 bc bc
6d 11 db ca
88 0b f9 00
a3 3e 86 93
7a fd 41 fd
After AddRoundKey
aa 61 82 68
8f dd d2 32
5f e3 4a 46
03 ef d2 9a
48 67 4d d6
6c 1d e3 5f
4e 9d b1 58
ee 0d 38 e7
e0 c8 d9 85
92 63 b1 b8
7f 63 35 be
e8 c0 50 01
f1 c1 7c 5d
00 92 c8 b5
6f 4c 8b d5
55 ef 32 0c
26 3d e8 fd
0e 41 64 d2
2e b7 72 8b
17 7d a9 25

INTRODUZIONE ALLA CRITTOGRAFIA





INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

2. Crittografia simmetrica

- ☐ Cos’è la crittografia simmetrica?
- ☐ Algoritmi di crittografia simmetrica
- ☐ Crittografia simmetrica client-server
- ☐ Crittografia simmetrica end-to-end
- ☐ Cifratura simmetrica super sicura

AUTORE

Pietro Terracciano

Crittografia simmetrica
è la più semplice in assoluto

Crittografia simmetrica
utilizza una Chiave definita
Chiave privata

Crittografia simmetrica
abbiamo cifratura e decifratura
con la stessa Chiave privata

Crittografia simmetrica
è come una cassaforte
condivisa : chi ha la copia della
Chiave la può aprire

Crittografia simmetrica
la Chiave è il nostro seme.
Viene aggiunto anche un IV -
Initialization Vector

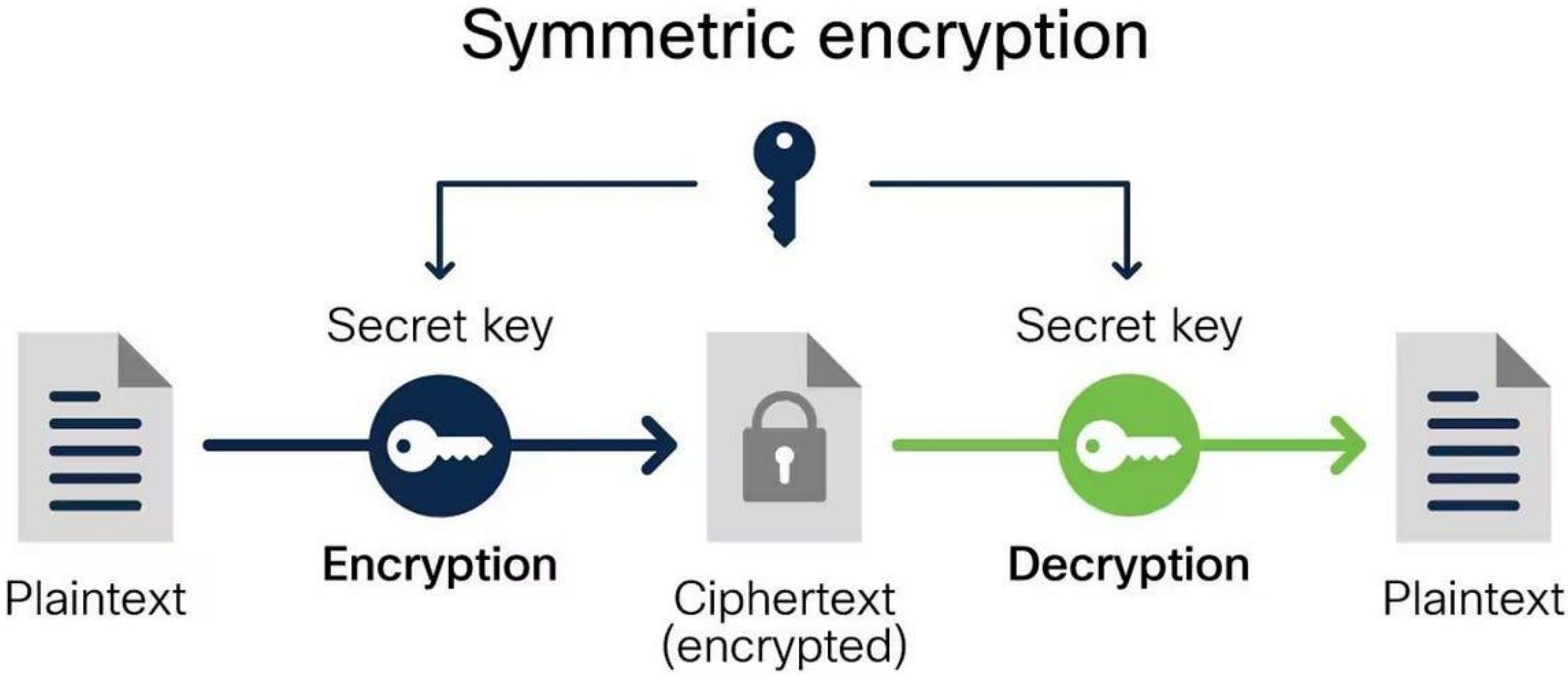
INTRODUZIONE ALLA CRITTOGRAFIA

“La crittografia simmetrica è la forma più semplice e veloce di cifratura: usa un’unica chiave segreta, condivisa tra mittente e destinatario, per trasformare il messaggio in qualcosa di illeggibile per chiunque altro.

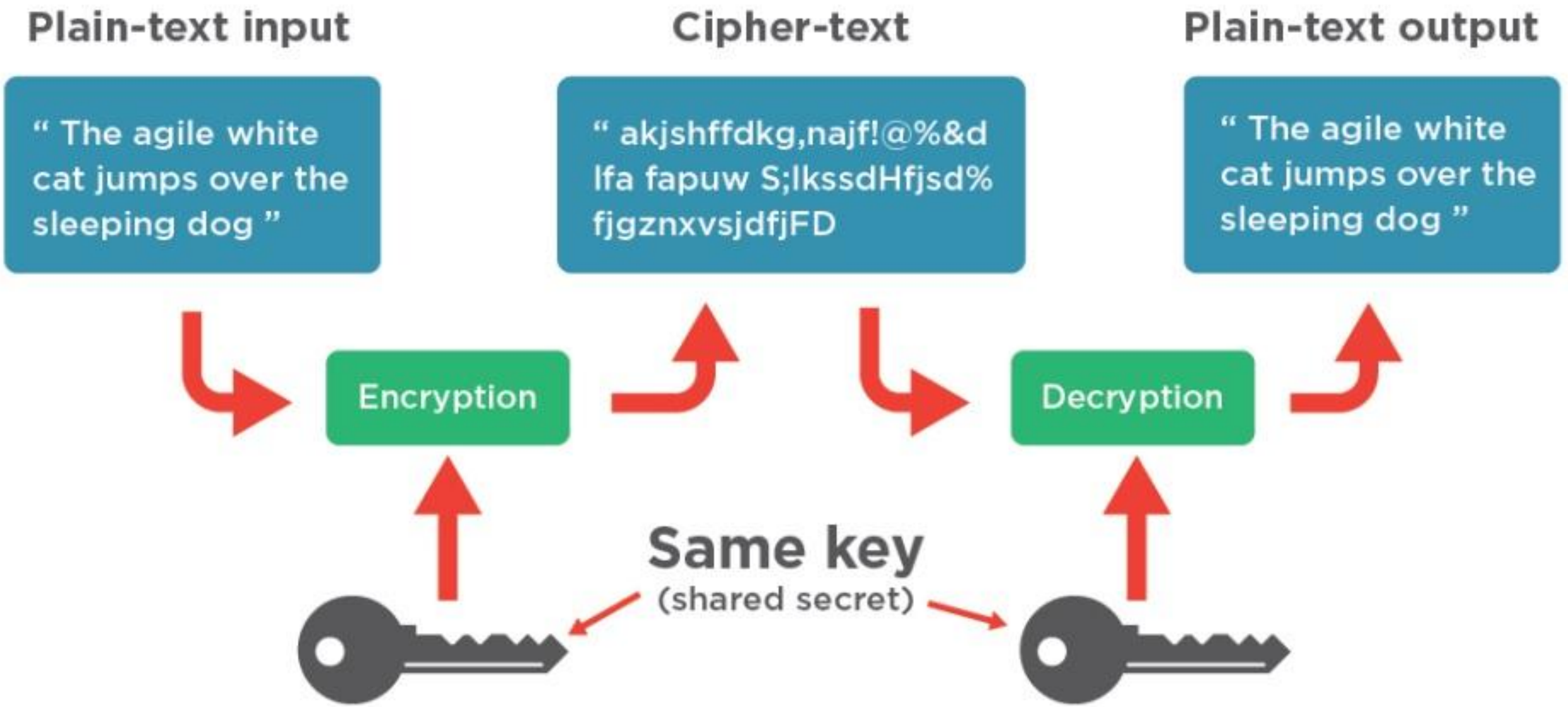
La cifratura simmetrica è come una cassaforte condivisa: chi conosce la combinazione può sia chiudere che aprire. Tutta la sicurezza dipende da quella chiave.

Con la cifratura simmetrica possiamo proteggere qualsiasi file in un attimo: basta una chiave segreta condivisa, e chi non la conosce vedrà solo dati casuali.”

INTRODUZIONE ALLA CRITTOGRAFIA



INTRODUZIONE ALLA CRITTOGRAFIA

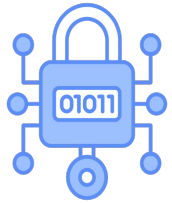


INTRODUZIONE ALLA CRITTOGRAFIA

```
Text:            hello
Pass phrase:     qwerty
-------------------------

IV:      6BE952EBC17EED10411EAA9892F19124
Key:     33A5820536F9EEB709D88AF3B40FDBB100C04327C71B5ACCF48424C8EB40C3F9
-------------------------

Cipher:          U2FsdGVkX18kH6hnY7hTQQvZbphR8WXJbOkwYSCKfY0=
  Hex: 53616C7465645F5F241FA86763B853410BD96E9851F165C96CE93061208A7D8D
Decrypt:         hello
```

INTRODUZIONE ALLA CRITTOGRAFIA

```
Please put in the text to be encrypted.
foo bar
Start: foo bar
Encrypted: K??Kz+:??*???f58
Encrypted Base64: Sz8/S3orOj8/Kj8/P2Y1OA==
From Base64 To AES Encypted Text: K??Kz+:??*???f58
Decrypted: foo bar
Decrypted From Encode and then Decode Base64 Text: ?    ??♣♦#?x?}?↕Q?
```



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

2. Crittografia simmetrica

- ☑ Cos’è la crittografia simmetrica?
- ☐ Algoritmi di crittografia simmetrica
- ☐ Crittografia simmetrica client-server
- ☐ Crittografia simmetrica end-to-end
- ☐ Cifratura simmetrica super sicura

AUTORE

Pietro Terracciano

Algoritmi di crittografia simmetrica
algoritmi matematici ad alte prestazioni

Algoritmi di crittografia simmetrica
abbiamo già detto che lavorano a blocchi

Algoritmi di crittografia simmetrica
DES - Data Encryption Standard
3DES - Triple DES
AES - Advanced Encryption Standard

Algoritmi di crittografia simmetrica
Ad oggi il più sicuro è AES che implementa l’algoritmo matematico Rijndael di due crittografi belgi Joan Daeman e Vincent Rijmen

INTRODUZIONE ALLA CRITTOGRAFIA

“AES, o Advanced Encryption Standard, è un cifratore simmetrico a blocchi che lavora su blocchi da 128 bit e supporta chiavi da 128, 192 e 256 bit.

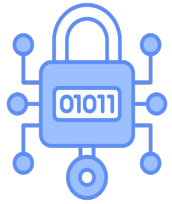
È lo standard approvato dal NIST e utilizzato in quasi tutti i protocolli moderni di sicurezza, da TLS a SSH.

Se la sicurezza informatica avesse un cuore, sarebbe AES: un algoritmo elegante, veloce e praticamente indistruttibile, che ogni giorno protegge miliardi di comunicazioni nel mondo.”

INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

```
openssl enc -aes-256-cbc -salt -in messaggio.txt -out messaggio.enc -k "passwordSegreta"
openssl enc -d -aes-256-cbc -in messaggio.enc -out messaggio_decifrato.txt -k "passwordSegreta"
```



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

2. Crittografia simmetrica

- ☑ Cos’è la crittografia simmetrica?
- ☑ Algoritmi di crittografia simmetrica
- ☐ Crittografia simmetrica client-server
- ☐ Crittografia simmetrica end-to-end
- ☐ Cifratura simmetrica super sicura

AUTORE

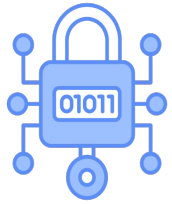
Pietro Terracciano

Crittografia simmetrica client-server
il client-server condividono la stessa Chiave privata e, quindi, va conservata in una zona super sicura

Crittografia simmetrica client-server
se utilizziamo dei linguaggi interpretati (es Bytecode), bisogna sporcare il codice con offuscatori

Crittografia simmetrica client-server
la Chiave privata deve essere scelta ad-hoc per tutti. Non va spedita in rete o salvata su file temporanei o di facile trasporto

Cifratura simmetrica client-server
se un malintenzionato ottiene la Chiave privata, legge i messaggi di tutta la rete



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

2. Crittografia simmetrica

- ☑ Cos’è la crittografia simmetrica?
- ☑ Algoritmi di crittografia simmetrica
- ☑ Crittografia simmetrica client-server
- ☐ Crittografia simmetrica end-to-end
- ☐ Cifratura simmetrica super sicura

AUTORE

Pietro Terracciano

Crittografia simmetrica end-to-end
viene stabilita una Chiave privata esclusivamente tra due endpoint (es. client-client)

Crittografia simmetrica end-to-end
per stabilire una Chiave privata tra due endpoint, spesso, si utilizzano canali poco sicuri : si usa, ad esempio, lo Scambio chiavi Diffie-Hellman

Crittografia simmetrica end-to-end
se un malintenzionato ottiene la Chiave privata di una coppia di endpoint (end-to-end), ha accesso solo ed esclusivamente ai messaggi di quella conversazione e non a tutta la rete

INTRODUZIONE ALLA CRITTOGRAFIA

“Nella cifratura simmetrica la stessa chiave serve sia per cifrare che per decifrare. È estremamente efficiente e ideale per proteggere grandi quantità di dati, ma richiede un modo sicuro per scambiare la chiave tra le parti.

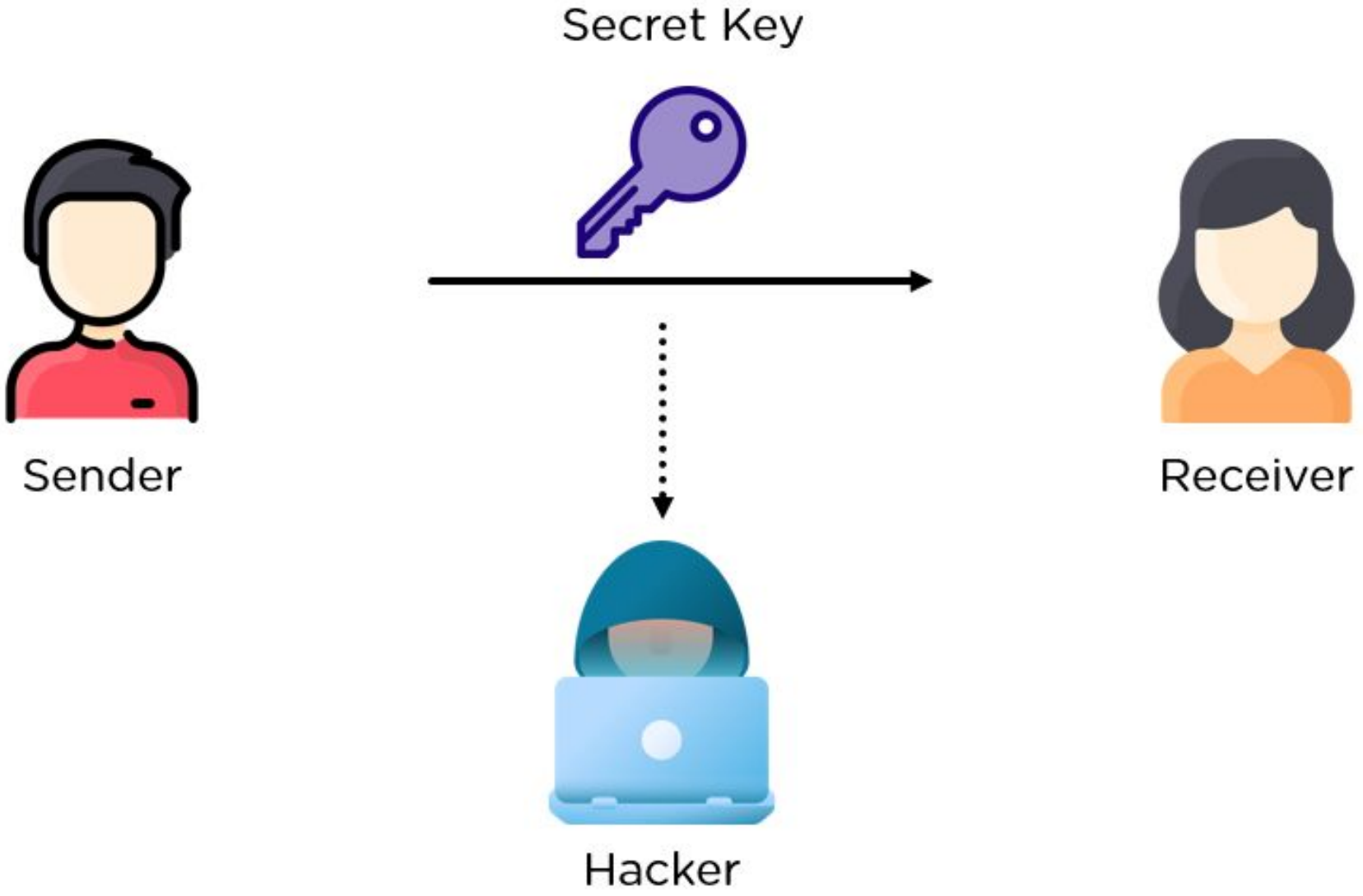
Diffie-Hellman non cifra i messaggi: costruisce il segreto.”

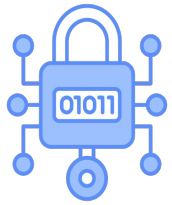
INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

```
# Parametri DH (pubblici)
openssl dhparam -out dhparam.pem 2048

# Alice e Bob generano le loro chiavi
openssl genpkey -paramfile dhparam.pem -out alice_priv.pem
openssl pkey -in alice_priv.pem -pubout -out alice_pub.pem

openssl genpkey -paramfile dhparam.pem -out bob_priv.pem
openssl pkey -in bob_priv.pem -pubout -out bob_pub.pem

# Scambio e derivazione della chiave segreta comune
openssl pkeyutl -derive -inkey alice_priv.pem -peerkey bob_pub.pem -out secret_alice.bin
openssl pkeyutl -derive -inkey bob_priv.pem -peerkey alice_pub.pem -out secret_bob.bin
```

INTRODUZIONE ALLA CRITTOGRAFIA





INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

2. Crittografia simmetrica

- ☑ Cos’è la crittografia simmetrica?
- ☑ Algoritmi di crittografia simmetrica
- ☑ Crittografia simmetrica client-server
- ☑ Crittografia simmetrica end-to-end
- ☐ Cifratura simmetrica sicura

AUTORE

Pietro Terracciano

Cifratura simmetrica sicura
per potenziare il vincolo privacy ed unicità, va aggiunto il SALT alla cifratura : la Chiave privata, da sola, genera sempre lo stesso CipherText. Evita i Brute Force

Cifratura simmetrica sicura
per aggiungere il vincolo di integrità va usata funzione hash. Scartiamo il CipherText se è stato compromesso

Cifratura simmetrica sicura
la cifratura simmetrica da sola non rispetta il vincolo di autenticazione

Cifratura simmetrica sicura
Diffie-Hellman è solo un algoritmo

Cifratura simmetrica sicura
non siamo protetti da Man-in-the-Middle. Manca una Authority

INTRODUZIONE ALLA CRITTOGRAFIA

“Con l’Hash garantiamo che il messaggio non sia stato alterato (integrità), con il SALT garantiamo che ogni cifratura sia unica (privacy).

Tuttavia non siamo in grado di rispettare il vincolo di Autenticazione ed inoltre non siamo protetti da Man-in-the-Middle. Serve una Authority”

INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

```
openssl enc -aes-256-cbc -salt -in file.txt -out file.enc -k password
```

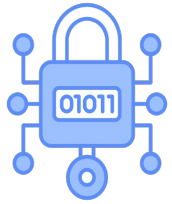
INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro tra client-server ed end-to-end

```
#!/bin/bash
# Mini demo: AES + SALT + HASH (SHA256)

echo "messaggio segreto" > messaggio.txt              # File da cifrare
openssl rand -out chiave.bin 32                       # Chiave AES (32 byte)
openssl rand -out salt.bin 16                         # SALT (16 byte)

# Cifratura con AES-256-CBC e SALT
openssl enc -aes-256-cbc -salt -S "$(xxd -p salt.bin)" \
  -in messaggio.txt -out messaggio.enc -pass file:chiave.bin

# Hash SHA256 del file cifrato
sha256sum messaggio.enc > messaggio.enc.sha256

# Verifica e decifratura (solo se hash valido)
sha256sum -c messaggio.enc.sha256 && \
openssl enc -d -aes-256-cbc -in messaggio.enc -out messaggio_decifrato.txt -pass file:chiave.bin

cat messaggio_decifrato.txt
```

INTRODUZIONE ALLA CRITTOGRAFIA

				
Password	`p4s5w3rdz`	`p4s5w3rdz`	`p4s5w3rdz`	`p4s5w3rdz`
Salt	-	-	`et52ed`	`ye5sf8`
Hash	`f4c31aa`	`f4c31aa`	`lvn49sa`	`z32i6t0`



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

3. Crittografia asimmetrica

- ☐ Cos’è la crittografia asimmetrica?
- ☐ Algoritmi di crittografia asimmetrica
- ☐ Perchè si preferisce non usare la crittografia asimmetrica per l’end-to-end?
- ☐ Cifratura asimmetrica super sicura

AUTORE

Pietro Terracciano

Crittografia asimmetrica
è molto più complessa perchè bisogna gestire una coppia di Chiavi

Crittografia asimmetrica
abbiamo una Chiave pubblica, utilizzata per cifrare ed una Chiave privata utilizzata per decifrare

Crittografia asimmetrica
la Chiave pubblica può essere spedita in rete e può essere data a chiunque perchè serve solo a cifrare

Crittografia asimmetrica
può essere utilizzata come crittografia end-to-end, come aiutante nello Scambio delle Chiavi, come firma digitale, etc.

INTRODUZIONE ALLA CRITTOGRAFIA

“Il limite della crittografia simmetrica non è nella cifratura, ma nella distribuzione della chiave. Per risolvere questo, negli anni ’70 nasce la crittografia asimmetrica: un sistema che elimina la necessità di condividere segreti in anticipo.

Diffie-Hellman è come stringere la mano a qualcuno in una stanza buia: la stretta è sicura, ma non sai chi hai davanti. Soffre di attacchi Man-in-the-Middle. La crittografia asimmetrica accende la luce: ti permette di riconoscere con chi stai davvero parlando.

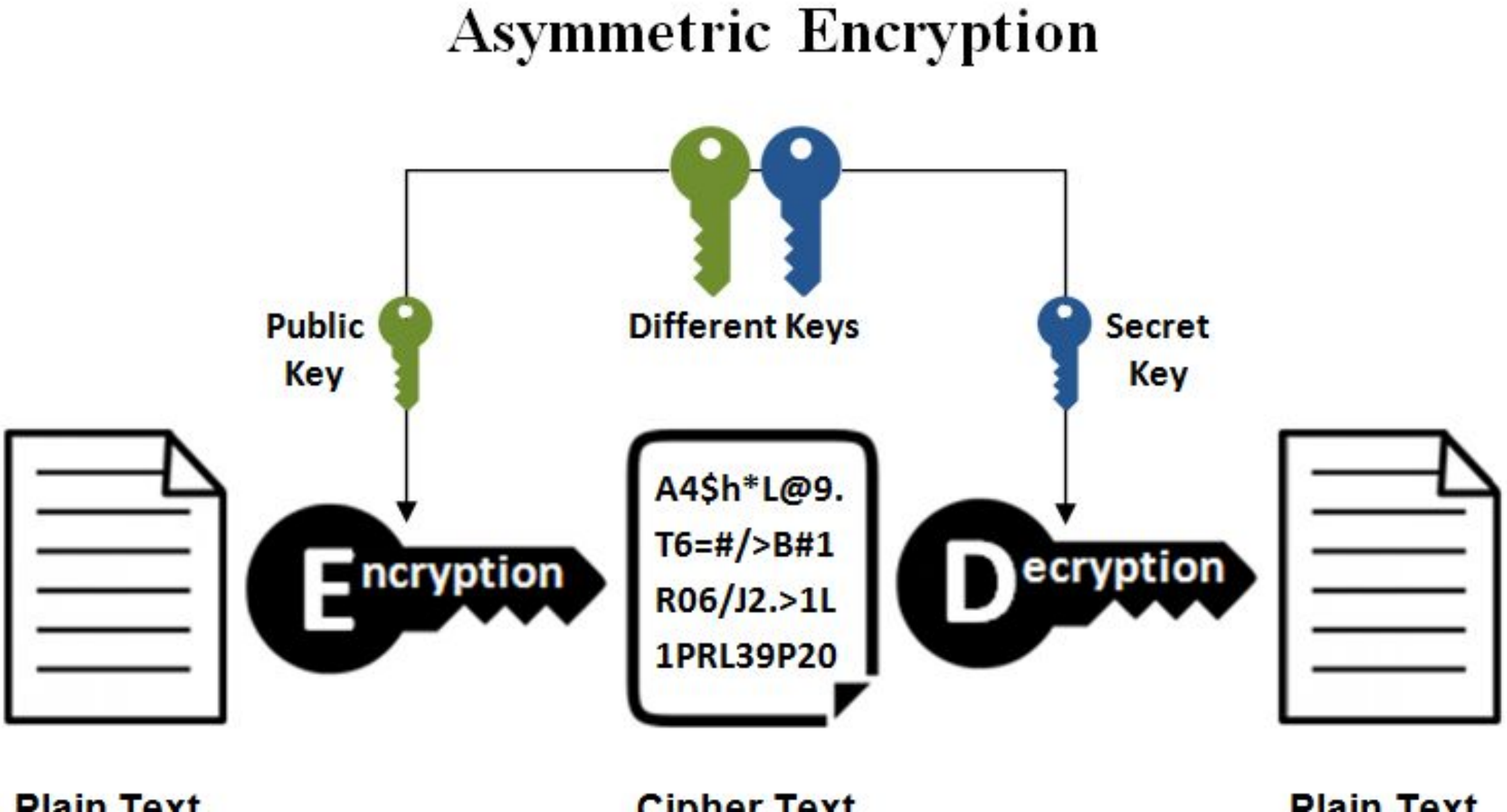
Diffie-Hellman crea la chiave, l’asimmetrica garantisce chi la sta creando”

INTRODUZIONE ALLA CRITTOGRAFIA

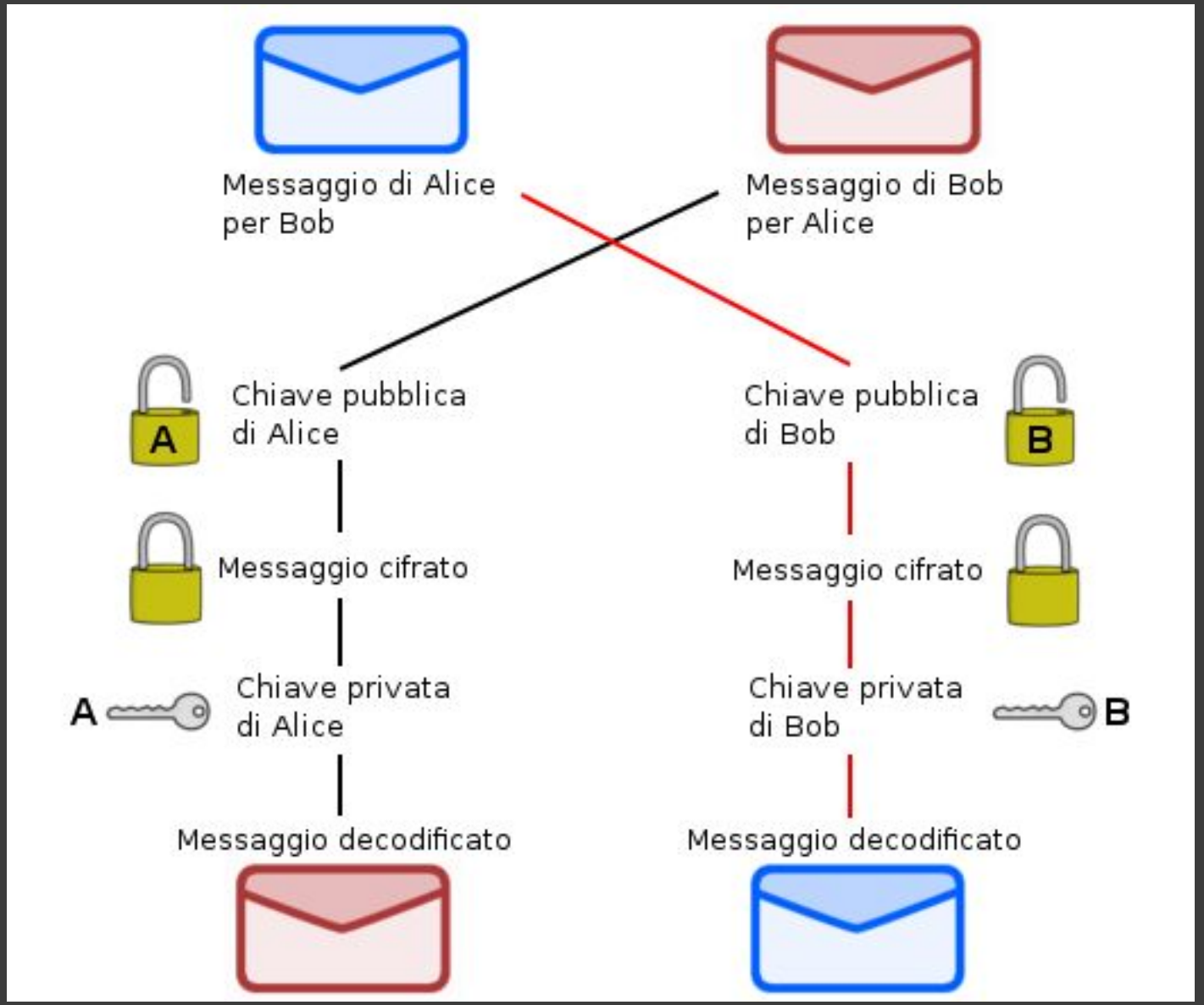
“La crittografia asimmetrica è come avere una cassetta postale: chiunque può infilare un messaggio dentro (chiave pubblica), ma solo tu puoi aprirla con la chiave privata.

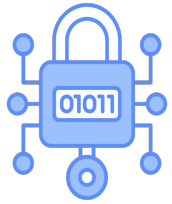
E la firma digitale è l’opposto: serve a dimostrare che sei stato tu ad aprire la cassetta.

Con la crittografia asimmetrica, possiamo fare due cose cruciali: cifrare i dati in modo che solo il destinatario possa leggerli e firmare i messaggi per garantire che provengano veramente dal mittente.”


Asymmetric Encryption
Public Key
Different Keys
Secret Key
Encryption
A4$h*L@9.
T6=#/>B#1
R06/J2.>1L
1PRL39P20
Decryption
Plain Text
Cipher Text
Plain Text

INTRODUZIONE ALLA CRITTOGRAFIA





INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

3. Crittografia asimmetrica

- ☑ Cos’è la crittografia asimmetrica?
- ☐ Algoritmi di crittografia asimmetrica
- ☐ Perchè si preferisce non usare la crittografia asimmetrica per l’end-to-end?
- ☐ Cifratura asimmetrica super sicura

Algoritmi di crittografia asimmetrica
algoritmi matematici molto più complessi

Algoritmi di crittografia asimmetrica
abbiamo già detto che lavorano a blocchi

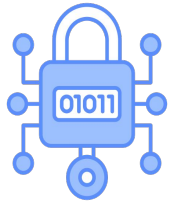
Algoritmi di crittografia simmetrica
Il più sicuro ed utilizzato è l’algoritmo RSA inventato dai crittografi Ronald Rivest, Adi Shamir e Leonard Adleman

AUTORE

Pietro Terracciano

INTRODUZIONE ALLA CRITTOGRAFIA

```
# Esempio su Linux (OpenSSL)
# Genera chiave privata RSA
openssl genpkey -algorithm RSA -out bob_priv.pem -pkeyopt rsa_keygen_bits:2048

# Estrai la chiave pubblica
openssl rsa -pubout -in bob_priv.pem -out bob_pub.pem

# Cifra con la chiave pubblica (chiunque può farlo)
openssl rsautl -encrypt -inkey bob_pub.pem -pubin -in messaggio.txt -out messaggio.enc

# Solo Bob può decifrare
openssl rsautl -decrypt -inkey bob_priv.pem -in messaggio.enc -out messaggio_decifrato.txt
```



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

3. Crittografia asimmetrica

- ☑ Cos’è la crittografia asimmetrica?
- ☑ Algoritmi di crittografia asimmetrica
- ☐ Perchè si preferisce non usare la crittografia asimmetrica per l’end-to-end?
- ☐ Cifratura asimmetrica super sicura

Perchè si preferisce non usare la crittografia asimmetrica per l’end-to-end?

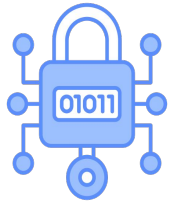
Generalmente non viene usata nella crittografia end-to-end perchè è molto lenta vista la complessità della Coppia delle Chiavi. Si preferisce usarla come supporto nello Scambio delle Chiavi, come mediatore per la firma digitale, etc.

AUTORE

Pietro Terracciano

INTRODUZIONE ALLA CRITTOGRAFIA

“La crittografia asimmetrica è sicura, ma troppo lenta e pesante per proteggere ogni singolo messaggio.
Per questo la usiamo solo per scambiare la chiave, e poi lasciamo il lavoro alla crittografia simmetrica, molto più efficiente.”



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

3. Crittografia asimmetrica

- ☑ Cos’è la crittografia asimmetrica?
- ☑ Algoritmi di crittografia asimmetrica
- ☑ Perchè si preferisce non usare la crittografia asimmetrica per l’end-to-end?
- ☐ Cifratura asimmetrica super sicura

AUTORE

Pietro Terracciano

Cifratura asimmetrica sicura per potenziare il vincolo privacy ed unicità, va aggiunto il SALT alla cifratura

Cifratura asimmetrica sicura per aggiungere il vincolo di integrità va usata crittografia hash

Cifratura asimmetrica sicura per aggiungere il vincolo di autenticazione va firmato l’hash con la Chiave privata. L’hash va firmato non cifrato

Cifratura asimmetrica sicura anche se rispettiamo privacy, integrità ed autenticazione non siamo protetti da Man-in-the-Middle. Manca una Authority

INTRODUZIONE ALLA CRITTOGRAFIA

“Con l’Hash garantiamo che il messaggio non sia stato alterato (integrità), con il SALT garantiamo che ogni cifratura sia unica (privacy), con la Firma garantiamo che il messaggio sia stato spedito esattamente da quel mittente (Autenticazione).

Tuttavia non siamo ancora protetti da Man-in-the-Middle”

INTRODUZIONE ALLA CRITTOGRAFIA

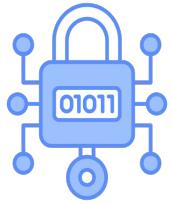
“La firma non è semplicemente ‘cifrare l’hash con la privata’ in senso pratico: è un’operazione privata progettata per essere verificata pubblicamente con i corretti schemi di padding/hash.

Criptare l’hash con la chiave pubblica non produce né la stessa cosa né l’effetto di autenticazione — serve invece usare gli schemi di firma adeguati (es. RSA-PSS, Ed25519).”

INTRODUZIONE ALLA CRITTOGRAFIA

```
# Alice genera la sua chiave privata
openssl genpkey -algorithm RSA -out alice_priv.pem -pkeyopt rsa_keygen_bits:2048

# Estrai la chiave pubblica di Alice
openssl rsa -in alice_priv.pem -pubout -out alice_pub.pem

# Crea il messaggio da firmare
echo "Questo è un messaggio firmato da Alice" > messaggio.txt

# Firma del messaggio con la chiave privata di Alice (SHA256)
openssl dgst -sha256 -sign alice_priv.pem -out firma.bin messaggio.txt
```

```
# Verifica la firma con la chiave pubblica di Alice
openssl dgst -sha256 -verify alice_pub.pem -signature firma.bin messaggio.txt
```



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

4. Certificato ed Authority

- ☐ Cos’è un certificato?
- ☐ Cos’è una Authority?

AUTORE

Pietro Terracciano

Certificato
Chiave pubblica con metadati (Nome, Cognome, Partita IVA, etc)

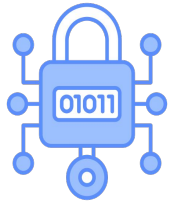
Certificato
viene spedito all’endpoint che ha richiesto il servizio di un altro endpoint (server) e viene validato sull’endpoint richiedente

Certificato
effettuo una CSR - Certificate Signing Request ad una Authority che include Chiave pubblica, e tutti i metadati necessari (Nome, Cognome, Partita IVA, etc)
Mi viene fornito un certificato firmato dalla Authority

INTRODUZIONE ALLA CRITTOGRAFIA

```
openssl genpkey -algorithm RSA -out private.key -pkeyopt rsa_keygen_bits:2048
# oppure per ECC:
openssl ecparam -genkey -name prime256v1 -noout -out private_ec.key
```

```
openssl req -new -key private.key -out request.csr \
  -subj "/C=IT/ST=Roma/L=Roma/O=MiaOrg/CN=www.esempio.com"
# Per SAN usa config file o -reqexts
```



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

4. Certificato ed Authority

☑️ Cos’è un certificato?
☐ Cos’è una Authority?

Authority
ente in grado di rilasciare un certificato firmato

Authority
se ad esempio altero il TLS per ottenere un Man-in-the-Middle. Il client chiederà alla CA la validità del certificato per il TLS originale. Abbiamo una catena di responsaibilità

AUTORE

Pietro Terracciano

INTRODUZIONE ALLA CRITTOGRAFIA

“Ogni certificato nella catena è garantito da quello che lo precede: uso la chiave pubblica del certificato superiore per verificare la firma del certificato inferiore.

È così che si costruisce la fiducia, fino ad arrivare a una root CA già fidata dal sistema.

Con l’Authority garantiamo l’Autorizzazione

Risolve il Man-in-the-Middle a patto di non vulnerabilità nella connessione (es. accettare certificati non validi, connettersi ad una rete pubblica/privata che non ha fiducia e più applicare DNS poisoning).”

INTRODUZIONE ALLA CRITTOGRAFIA

```
[leaf cert (www.example.com)]  <-- firmato da
[intermediate CA cert]         <-- firmato da
[... eventuali intermediates ...]
[root CA cert (self-signed)]   (trusted a priori nel trust store)
```

```
Tuo certificato (leaf)  ← firmato con la private key della CA intermedia
CA intermedia           ← firmata con la private key della CA root
CA root                 ← firmata con la propria private key (self-signed)
```

EVENTO

Linux Day 2025

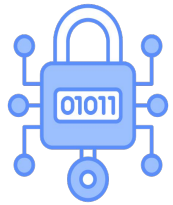
AUTORE

Pietro Terracciano

GRAZIE A

IrLUG
Irpinia Linux User Group

lumaca IRPINA



INTRODUZIONE ALLA CRITTOGRAFIA

Come implementare uno scambio dati sicuro
tra client-server ed end-to-end

GITHUB

https://github.com/pietroterracciano/

LINKEDIN

https://www.linkedin.com/in/pietroterracciano/

INSTAGRAM

https://www.instagram.com/pietroterracciano95/

EMAIL

pterracciano95@gmail.com