

Cifratura Simmetrica con OpenSSL

Alfredo De Santis

Dipartimento di Informatica
Università di Salerno

ads@unisa.it



Maggio 2020

Outline

- Concetti Preliminari
- Cifratura Simmetrica in OpenSSL

Outline

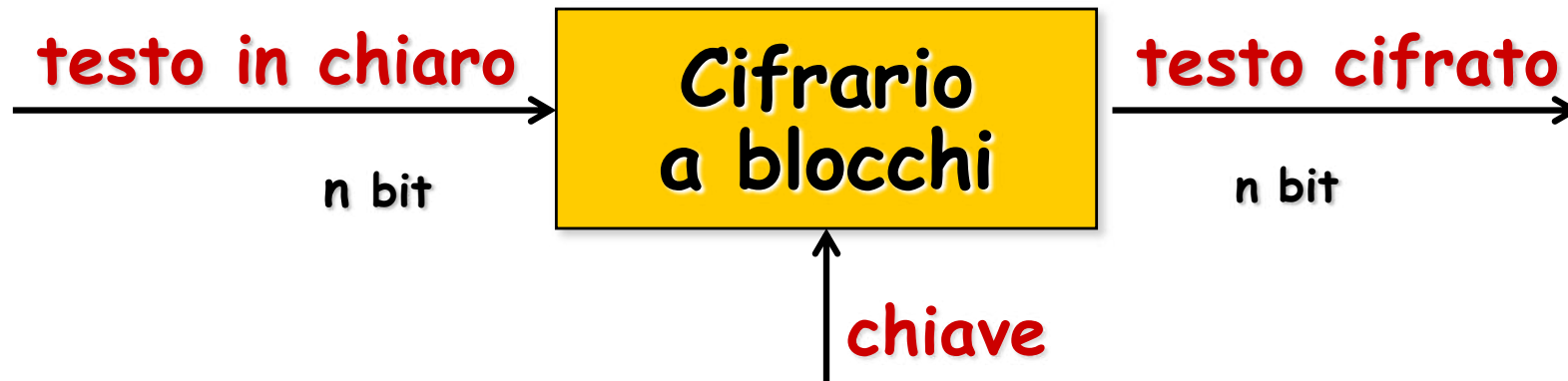
- Concetti Preliminari
- Cifratura Simmetrica in OpenSSL

Cifrari Simmetrici

- Crittosistemi a chiave privata (o segreta)
 - Alice e Bob conoscono la **stessa** chiave **K**
- Due tipologie di crittosistemi a chiave privata
 - Cifrari a Blocchi
 - Messaggi divisi in blocchi e poi cifrati
 - Stream Cipher
 - Messaggi cifrati carattere per carattere



Cifrari a Blocchi



➤ Alcuni Esempi

- Data Encryption Standard (DES)
- DES triplo
- Blowfish
- Advanced Encryption Standard (AES)

Outline

- Concetti Preliminari
- Cifratura Simmetrica in OpenSSL

Cifratura Simmetrica in OpenSSL

Cifrari e Modalità Operative Supportate

- OpenSSL fornisce numerosi cifrari simmetrici
- Molti cifrari supportano varie modalità operative
 - ECB, CBC, CFB, OFB e CTR
- La modalità operativa di default è CBC, se nessun'altra è esplicitamente specificata

Cifratura Simmetrica in OpenSSL

Cifrari e Modalità Operative Supportate

- Per conoscere i cifrari supportati da OpenSSL
 - `openssl list --cipher-commands`

```
aes-128-cbc      aes-128-ecb      aes-192-cbc      aes-192-ecb
aes-256-cbc      aes-256-ecb      aria-128-cbc      aria-128-cfb
aria-128-cfb1    aria-128-cfb8    aria-128-ctr      aria-128-ecb
aria-128-ofb     aria-192-cbc     aria-192-cfb      aria-192-cfb1
aria-192-cfb8    aria-192-ctr     aria-192-ecb      aria-192-ofb
aria-256-cbc     aria-256-cfb     aria-256-cfb1     aria-256-cfb8
aria-256-ctr     aria-256-ecb     aria-256-ofb      base64
bf              bf-cbc          bf-cfb           bf-ecb
bf-ofb          camellia-128-cbc camellia-128-ecb  camellia-192-cbc
camellia-192-ecb camellia-256-cbc camellia-256-ecb  cast
cast-cbc        cast5-cbc       cast5-cfb        cast5-ecb
cast5-ofb       des             des-cbc          des-cfb
des-ecb         des-ede         des-ede-cbc      des-ede-cfb
des-ede-ofb     des-ede3        des-ede3-cbc     des-ede3-cfb
des-ede3-ofb    des-ofb         des3             desx
rc2             rc2-40-cbc      rc2-64-cbc       rc2-cbc
rc2-cfb        rc2-ecb         rc2-ofb          rc4
rc4-40         seed            seed-cbc         seed-cfb
seed-ecb       seed-ofb        sm4-cbc          sm4-cfb
sm4-ctr        sm4-ecb         sm4-ofb
```


Cifratura Simmetrica in OpenSSL

Struttura di un Ciphername

- Un ciphername è composto da al più 3 parti separate da un trattino '-'
 - Nome del Cifrario
 - Lunghezza della Chiave (in bit)
 - Modalità Operativa
- Esempio
 - aes-256-cbc
- N.B. è obbligatorio solo il nome del cifrario

Cifratura Simmetrica in OpenSSL

Caratteristiche Generali

- Dati letti dallo standard input e scritti sullo standard output
 - Possono anche essere specificati file di input ed output
- Solo un singolo file alla volta può essere cifrato o decifrato
- Ciascun cifrario richiede una chiave per effettuare la cifratura o la decifratura
 - Tale chiave deve essere nota solo al mittente ed ai destinatari dei dati cifrati

Cifratura Simmetrica in OpenSSL

Il Comando enc

- Il comando `enc` (encrypt/encode) permette di accedere ai cifrari simmetrici forniti da OpenSSL

```
openssl enc args
```

`args` sono i
parametri del
comando

CITRATURA SIMMETRICA IN OpenSSL

Opzioni principali del comando enc

`openssl enc args`

➤ **args**

- `-ciphername`
 - Tipo di cifrario, lunghezza della chiave e modalità operativa
 - Usare il comando `openssl list-cipher-commands` per ottenere la lista completa
- `-in filename`
 - File di input
- `-out filename`
 - File di output
- `-e` or `-d`
 - Specifica se si tratta di cifratura o decifratura
- `-K key`
 - Chiave usata dal cifrario per cifrare o decifrare. Se non viene specificata, OpenSSL deriverà questa chiave da una password
- `-pass arg`
 - Sorgente della password. I valori possibili per `arg` sono `pass:password` o `pass:filename`, dove `password` è la password e `filename` è il file contenente la password. Se non si usa questo parametro viene mostrato un prompt per inserire la password
- `-base64`
 - Applica la codifica Base64 prima o dopo le operazioni crittografiche

CITRATURA SIMMETRICA IN OpenSSL

Opzioni principali del comando enc

openssl enc args

➤ **args**

- **-ciphername**
 - Tipo di cifrario, l
 - Usare il comando
- **-in filename**
 - File di input
- **-out filename**
 - File di output
- **-e or -d**
 - Specifica se si tratta di cifratura o decifratura
- **-K key**
 - Chiave usata dal cifrario per cifrare o decifrare. Se non viene specificata, OpenSSL deriverà questa chiave da una password
- **-pass arg**
 - Sorgente della password. I valori possibili per arg sono pass:password o pass:filename, dove password è la password e filename è il file contenente la password. Se non si usa questo parametro viene mostrato un prompt per inserire la password
- **-base64**
 - Applica la codifica Base64 prima o dopo le operazioni crittografiche

Per ottenere la lista completa delle opzioni del comando enc è possibile utilizzare `man enc`

iva
ottenere la lista completa

Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Motivazioni ed Applicazioni

- Permette di memorizzare o trasferire un flusso arbitrario di dati binari mediante caratteri stampabili
 - Chiavi Crittografiche
 - Certificati
 - Allegati e-mail
 - Etc.
- Base64 è un sistema di decodifica/codifica per dati binari, che utilizza 64 simboli stampabili
 - A-Z, a-z, 0-9, + e /

Crittatura Simmetrica in OpenSSL

Codifica in Base64

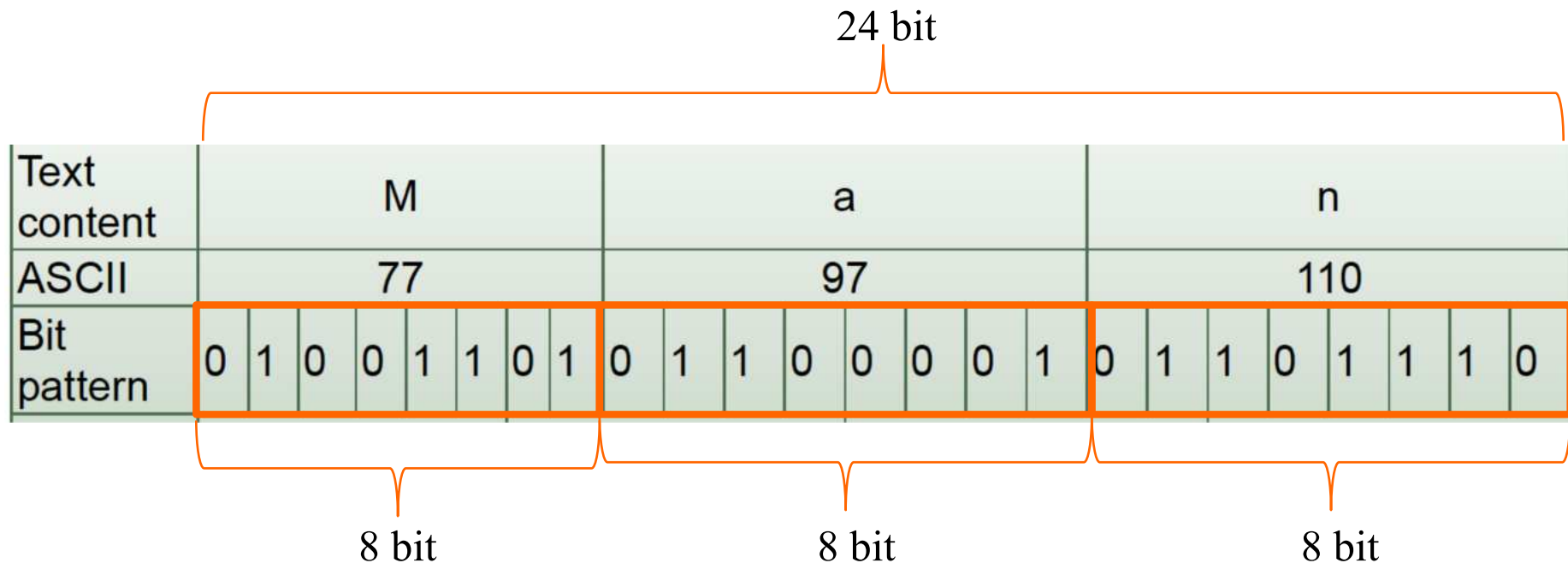
valore	codifica	valore	codifica	valore	codifica	valore	codifica
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Tabella di conversione Base64

Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Esempio 1

Il file in input è processato a blocchi da 24 bit

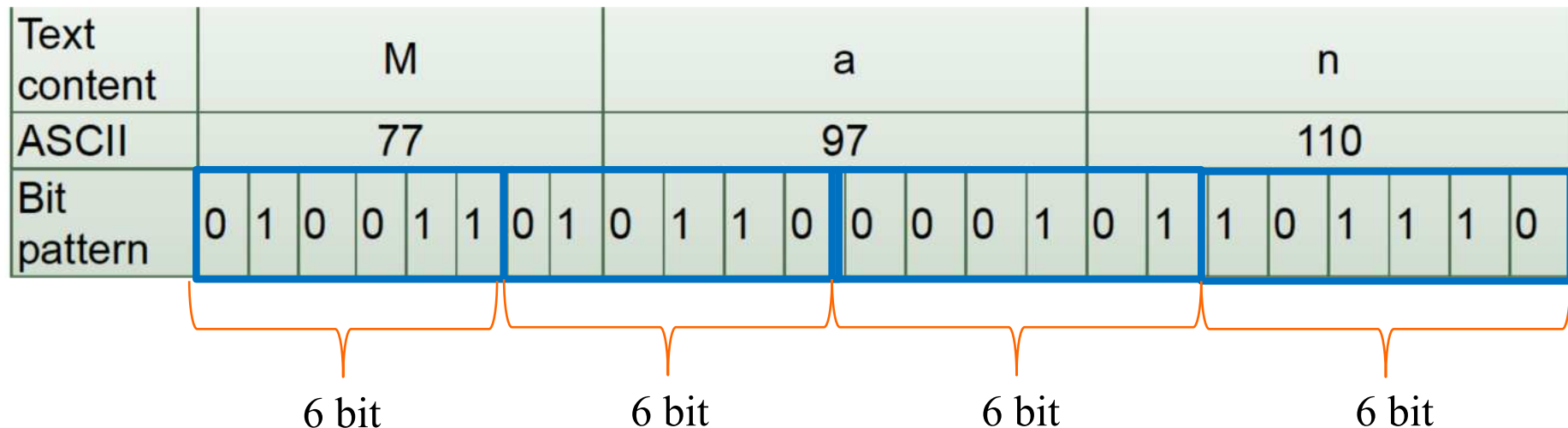


Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Esempio 1

Il file in input è processato a blocchi da 24 bit

➤ Ciascun blocco è suddiviso in gruppi di 6 bit (a partire da sinistra)




Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Esempio 1

Il file in input è processato a blocchi da 24 bit

- Ciascun blocco è suddiviso in gruppi di 6 bit (a partire da sinistra)
- Viene considerato il valore decimale di ciascun gruppo di bit, tale valore rappresenterà un indice nella tabella di codifica Base64 (valori da 0 a 63)

Text content	M								a								n															
ASCII	77								97								110															
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0								
Index	19								22								5								46							



Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Esempio 1

Il file in input è processato a blocchi da 24 bit

- Ciascun blocco è suddiviso in gruppi di 6 bit (a partire da sinistra)
- Viene considerato il valore decimale di ciascun gruppo di bit, tale valore rappresenterà un indice nella tabella di codifica Base64 (valori da 0 a 63)
- Ogni indice viene convertito in caratteri ASCII, secondo la tabella di conversione Base64

Text content	M								a								n							
ASCII	77								97								110							
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19						22						5						46					
Base64-encoded	T						W						F						u					

Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Esempio 1

- Il file in input è processato a blocchi da 24 bit
- Se il numero totale di bit da processare non è un multiplo di 24 allora viene utilizzato il padding
 - Vengono inseriti bit nulli (0) alla fine
 - Nella codifica viene inserito il simbolo '=' per ogni gruppo di 6 bit che manca per creare un blocco da 24 bit
 - Questo garantisce che l'output codificato in Base64 sia multiplo di 4 byte

<https://www.base64encode.org/>

Decode

Encode

Other

Encode to Base64 format

Simply use the form below

Man

> ENCODE <

UTF-8 You may also select output charset.

☒ Live mode ON

Encodes while you type or paste (strict format).

UPLOAD FILE

Encodes an entire file (max. 10MB).

TWFu

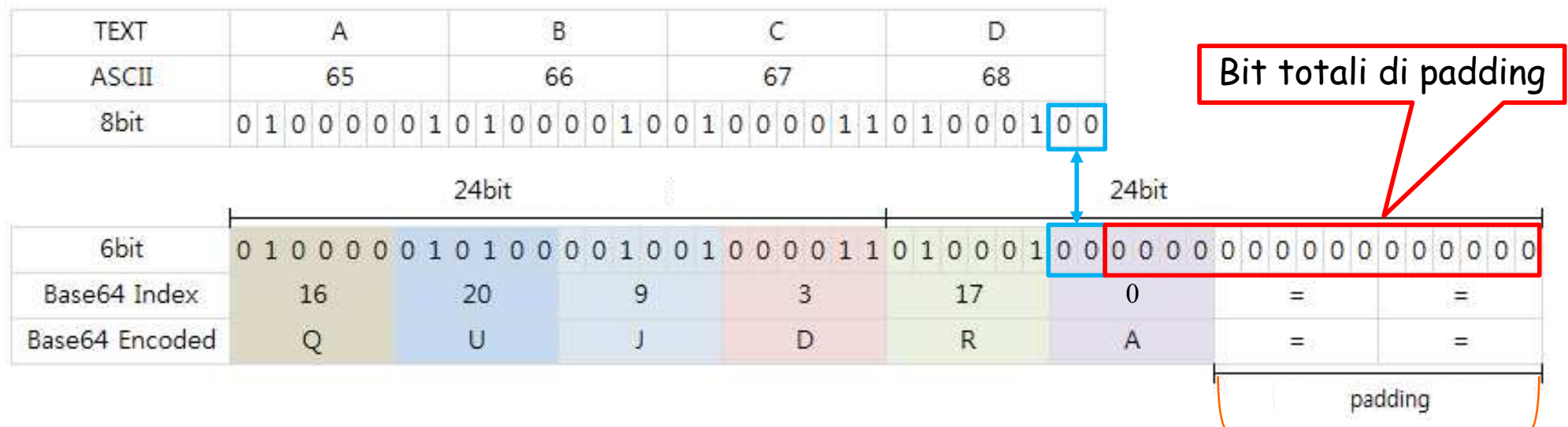
Codifica in Base64 - Esempio 2

TEXT	A	B	C	D
ASCII	65	66	67	68
8bit	<div style="border: 1px solid blue; padding: 2px; display: inline-block;"> 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 </div> 32 bit			

	24bit																								Encode																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
6bit	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Esempio 2



2 blocchi di padding,
che corrisponderanno
a due simboli '=' nella
codifica finale

<https://www.base64encode.org/>

Decode

Encode

Oth

Encode to Base64 format

Simply use the form below

Man

> ENCODE <

UTF-8

You may also select output charset.

Live mode ON

Encodes while you type or paste (strict format).

UPLOAD FILE

Encodes an entire file (max. 10MB).

TWFu

Encode to Base64 format

Simply use the form below

ABCD

> ENCODE <

UTF-8

You may also select output charset.

Live mode ON

Encodes while you type or paste (strict format).

UPLOAD FILE

Encodes an entire file (max. 10MB).

QUJDRA==

Encode to Base64 format

Simply use the form below

Ma

> ENCODE <

UTF-8

You may also select output charset.

Live mode ON

Encodes while you type or paste (strict format).

UPLOAD FILE

Encodes an entire file (max. 10MB).

TWE=

Encode to Base64 format

Simply use the form below

M

> ENCODE <

UTF-8

You may also select output charset.

Live mode ON

Encodes while you type or paste (strict format).

UPLOAD FILE

Encodes an entire file (max. 10MB).

TQ==

Crittatura Simmetrica in OpenSSL

Codifica in Base64 - Sintassi

- Per codificare un file in Base64 può essere usata la seguente sintassi

```
openssl enc -base64 -in input-file -out output-file
```

- Per decodificare un file codificato in Base64 può essere usata la seguente sintassi

```
openssl enc -base64 -d -in input-file -out output-file
```

Cifratura Simmetrica in OpenSSL

Esempio di Cifratura

FileInChiaro.rtf

Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia e aspra e forte che nel pensier rinova la paura! Tant'  amara che poco   pi  morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.



Cifratura

```
openssl enc -aes-256-cbc -in FileInChiaro.rtf -out FileCifrato.rtf -e
-pass pass:P1pp0B4ud0
```

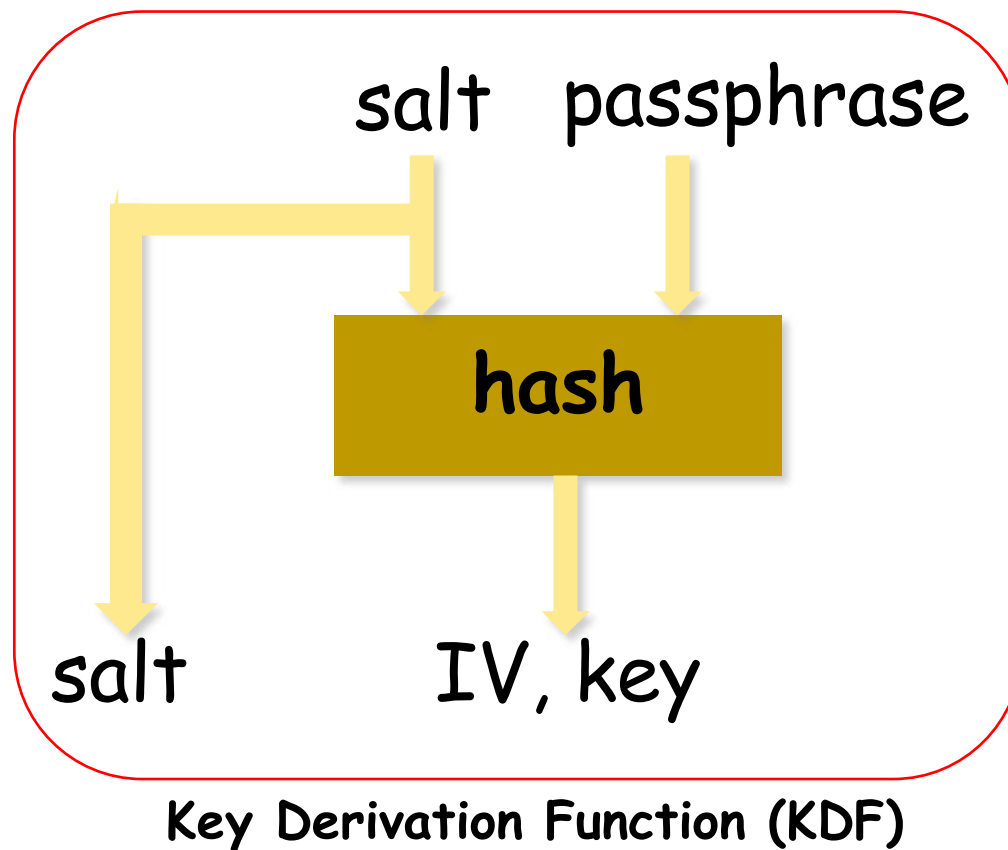


FileCifrato.rtf

Salted__M äio (ä~Ä<84>NwU&xEBj<8f>GvA6CzI<8f>iT<95><8c>^Z-){ÄÜQà~W^K'HÜP
 <95>:<9d>1PUë¹²³yK<89>Z+NÜG+ÜEW;puLlM<92>+Agç<90>b<82>-ð1E<85>NU\¹G<83>ð1ÄÖ¹²
 }¹²VYXq1qôB<8>Kmhh<84>Bö<9d>üðD84Ea¶³⁹e-i<93>⁹5~¹²VYGP6R¹L<85>EÄ88>JU
 @&º0<89><93>S<90>ú¹[ÐÆÛ<9d>±ØEªx<8d>Pðw×ú¹ADÉYRSùää<99>·-ÆñI\¹<85>¹M<8e>
 @,+0ïÄu(hi<9c>\¹Hs,¹Lý¹F<9c>gP9KA)¹h é¹H<86>Yaòx1?;¹S£Ç<90>y¹
 +.i<88>ZT=b<9d>2Ý<92>q<8b>-1¼ætQà¹W¹Ue<90>Ü_EKÑ¹X¹Q+Ø¹\$ûöI¹@t¹Yáô`T0<9c>=é¹x¹@E
 LB¹U<84>9e¹-(¹m¹Ü<8>¹æ<8a>G¹Sn, <98>
 LÈ¹BEj)-¹FB/¹!ÜB¹¶¹¹UO×¹¶èX7±¹H<87><85>S§¹C¹[¹⁹5~¹[öápGc¹Yéd<89><9f><90>Iï<8c>
 <87>¹{S¹I=K¹P<93>¹Jø¹Ä¹¹FDJ¹<94>ñ¹L¹¹Y<96>¹i.öY¹<96>¹B<9b>,¹LùÄ1f7T0Ñh;¹i¹Vowb¹Q
 ¹Höp<99>äd<97>¹Sð¹¹ScfööNy¹EiE¹YÄUöB~Ew¹M,¹L@¹P¹J¹¹ö¹Y¹MI¹¹Iä¹C<8d><8e>F<9
 5><92>®[¹qmi¹K¹Sö<9c><89>¹Ä<8c><97>ærq+<90>öf{¹X;¹1>¹Tc¹G6â¹]áh<8a>yY¹A*¹[¹[cöD#
 ®<8b>ö¹³y6y¹Q¹;¹L65m¹<86>¹i?U,NÈ¹¹ULY¹Mz¹¹500yD¹A¹Y¹SJ7<92>éo,u<8f>>ýÄ¹¹Lñx¹D¹
 HV¹¹q¹M<91>¹-®ðhöv
 R9a>æc=ÚÁ¹[rä]¹Ä¹½¹@Üüáy¹Pæ¹ùA<89>¹BIGPf<97>Üwö3ð¹ö~<83>¹¾¹Qúh<83>E4=¹0h®<96>¹i<8d
 >¹<8c><80>¹yÙe<97>¹Nð,¹95;¹¹Ü¹_PSZ<99>,¹a=Ü¹Q¹¹äq¹Ü?²Zö2Üßsq¹LG<83>¹çÄ¹¹B¹R¹®au<
 93>¹<88>a¹Y²w¹Vtè¹LCF²¹ÜPÉZZ¹CuLaëfs¹Zm3ð¹ÄFø<93>¹ä-Ø¹i<89>¹T&E¹[¹Hi¹C¹U¹<8c>¹i
 ¹¹yÜ¹¹ä<81><8e><99>-ä1<9a>on:CPÄ¹¿¹[2si¹h<8a>YU<82>Ø¹K¹¡YÇ<99>ØyÜ¹ iÜ¹P70<9d>

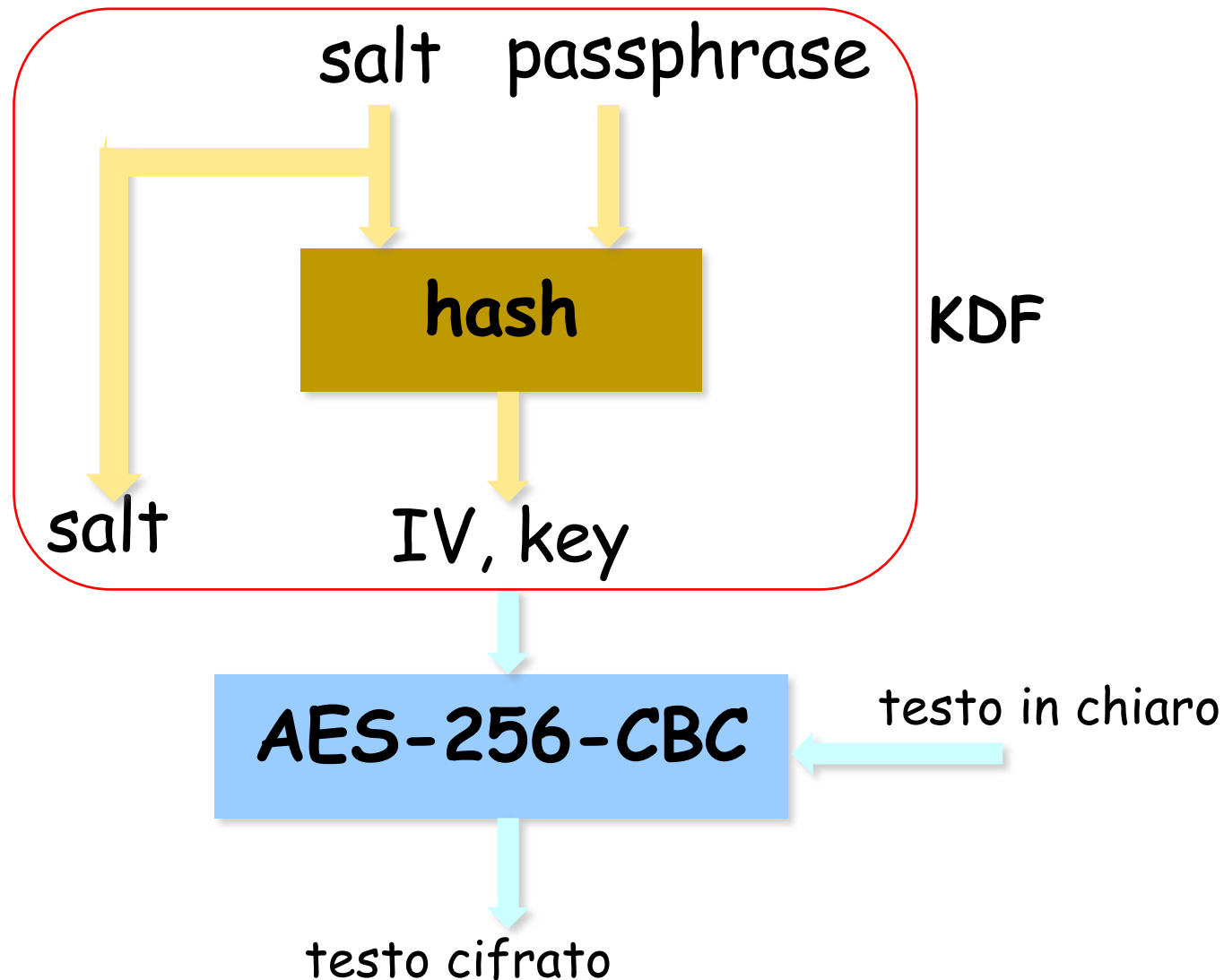
Cifratura con Passphrase

Derivazione Chiave



Cifratura con Passphrase

Derivazione Chiave e Cifratura



Cifratura Simmetrica in OpenSSL

Generare Chiavi da Password

- PBKDF2 (Password-Based Key Derivation Function 2)
 - Evoluzione di PBKDF1
 - Che generava chiavi solo a 160 bit
 - Pubblicato come RFC 2898
 - Usa HMAC più volte ed un salt per derivare una chiave di cifratura

Cifratura Simmetrica in OpenSSL

Esempio 1

FileInChiaro.rtf Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la
diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia
e aspra e forte che nel pensier rinova la paura! Tant'  amara che poco   pi 
morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.



Cifratura e Codifica in Base64 dell'output

```
openssl enc -aes-256-cbc -in FileInChiaro.rtf -out FileCifratoB64.rtf -e  
-base64 -pass pass:P1pp0B4ud0
```



FileCifratoB64.rtf

```
U2FsdGVKX18LBDbd1sxWx4xZGaTIB7dSwmJ1xAcgIqkpsRI3jhY3mLdnBzJ0mck3d2ZDhtGh8siGPmuuA2NF0dl1YdSf/zAK1Bo+90odha4e9r7  
f4qtUNfHmNIWJeQabMmcd50omOpIXYDhCROXiG490nTxSMcO+tvGNxTIJrqHUDqQ8xVBMnb0dafpTU+229dV/du6pwLCh9IKXDQIAvsPSc  
HbOoZyBDxmvgcI6kbPrEqL2M+/OfLaNumZWPNd4QwpFYHZKHb+UPXOp1BhGQ77M8GcVBcGeGarVE/BeAKOcyo78UhQ9BvNclNyw+3VlquY  
VVRQDZO8d6TLdLe6UUgSxblucELY19hRiIhb+HdgwjJKpdx+xBobksl69n9uTwEO8TzUsKkLog3L+ODAizZdXsidJX0+ZT76ijW7zkwufFUI8  
zYY1C6+RRqSNqiUB5VIXzUDIFxkOQ4JyN2mziKSQPv840o0+C4ShdWPP5eCXIoRA3Ar7bx4AO3AyyzP0nW1kHbCESddSFYho0+S0q4q6y0s  
4CvKabjch1yre+HZo+GpwtqCZK0Yxy11Isc2Ft2MmRm6X//QfKwdYhiMa0HhPpmGMjZgWP2dg/+HmIbKR8Mw41mAc3NoogBy4YbXYjopUdFrt  
adso6UnQmpr5uHDiks9KdxY7ZogNy2G5K8avSkaMj1sPHT9pBS23SrV9FYorg9AVZIIgYPuVeX9S+NS+P4/NzCeYXe/4fO6Uz8W/nLGgpEJ1G  
Ywu1qR9HtdwVvajvQnaqq0HV9sQcoLyvcaZUTrPLJ3+bFu3SvA8ha+TPQEzwIdXqxePf6QE/opdGJzmALfk9uIOu5awdDBYUzXiB8LMk0YiInD  
ZrQ5BLfnMzK5P80SH4zrhZ+rdHCAQ/QMyTJeb7BIF+HLEIfOu1Z/czUQQcmEd+A8bUD7WrfJHJ/UUZhlCtt/sCUArfXThzRNoIueRLPXJl  
KdIilbK8VyW/Ei7fCycrmPtQB4/xmstCSMtQT+POAKa6xHda1WsEPcPeOx04KmsxPtFFLbVP7W+wExgWZLgrzImN6WQvJo8h4mJ8zsoIeiz  
MHgSjq+oEAN1W7ou8TNZUyvURtg==
```

Cifratura Simmetrica in OpenSSL

Esempio 1

FileInChiaro.rtf Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la
diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia
e aspra e forte che nel pensier rinova la paura! Tant'  amara che poco   pi 
morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.



Cifratura e Codifica in Base64 dell'output

```
openssl enc -aes-256-cbc -in FileInChiaro.rtf -out FileCifratoB64.rtf -e  
-base64 -pass pass:P1pp0B4ud0
```

Eseguendo tale comando si ottiene il seguente warning
***** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.**

Per maggiori informazioni

➤ <https://courses.csail.mit.edu/6.857/2018/project/Ainane-Barrett-Johnson-Vivar-OpenSSL.pdf>

Cifratura Simmetrica in OpenSSL

Esempio 2

FileCifratoB64.rtf

```
U2FsdGVkX18LBDbd1sxWx4xZGaTIB7dSwmJ1xAcgLqkpsRI3jhY3mLdnBzJOMck3d2ZDhtGh8siGPmuuA2NF0dl1YdSf/zAK1B
o+90odha4e9r7f4qtUNfHmNIWJeQabMmcd50omOpIXYDhCROXiG490nTxSMcOtvGNxTIJrqHUDqQ8xVBMnb0dafpTU+22
9dV/du6pwLCh9IKXDQIAvsPSchB0oZyBDxmvgcI6kbPrEqL2M+/OfLaNumZWPnd4QwpFYHZKHb+UPX0p1BhGQ77M8GcVB
cGeGarVE/BeAKOcyo78UhQ9BvNclNyw+3VlquYVVBRQDZO8dGTLdLe6UUgSxblucELY19hRiIhb+HdgwjJKpdx+xHobksl69n9u
TwEO8TzUsKkLOg3LtODAizZdXsidJX0+ZT76ijW7zkwufFUI8zYY1C6+RRqSNqiUB5VIXzUDIFxkOQ4JyN2mziKSQPv840o0+
C4ShdWpP5eCXIoRA3Ar7bx4AO3AyzpP0nW1kHbCEsddSFYho0+S0q4qGy0s4CvKbjch1yre+HZo+GpwtqCZK0Yxy11Isc2Ft2
MmRm6X//QfKwdYhiMa0HhPpmGMjZgWP2dg/+HmIbKR8Mw41mAc3NoogBy4YbXYjopUdFrtadsoGUnQmpr5uHDiks9KdxY7
ZogNy2G5K8avSkaMj1sPHT9pBS23SrV9FYorg9AVZIIgYPuVeX9S+NS+P4/NzCeYXe/4fO6Uz8W/nLGgpEJ1GYwu1qR9Htdw
VvajvQnaqq0HV9sQcoLyvcaZUTrPLJ3+bFu3SvA8ha+TPQEzwIdXqxePfGQE/opdGJzmALfk9uIOu5awdDBYUzXiB8LMk0YiIn
DZrQ5BLfnMzK5P80SH4zrhZ+rdHCAQ/QMyTJeb7BIF+HLEIfOu1Z/czUQQcmEd+A8bUD7WrfJHJ/UUZhlCtt/sCUArfXTh
zRNoIueRlPXJIKdIilbK8VyW/Ei7fCycrmPtQB4/xmstCSMT+QTe+POAKa6xHda1WsEPcPeOx04KmsxPtFFLbVP7W+wExgWZLg
rzImN6WQvJo8h4mJ8zsoIeizMHgSjq+oEAN1W7ou8TNZUyvURtg==
```



Decifratura e Decodifica Base64

```
openssl enc -aes-256-cbc -in FileCifrato.rtf -out FileDecifrato.rtf
-d -base64 -pass pass:P1pp0B4ud0
```



FileDecifrato.rtf

Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia e aspra e forte che nel pensier rinnova la paura! Tant'  amara che poco   pi  morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.

Cifratura Simmetrica in OpenSSL

Esempio 3

FileInChiaro.rtf

Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia e aspra e forte che nel pensier rinova la paura! Tant'  amara che poco   pi  morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.



Cifratura con password letta da file e Codifica in Base64

```
openssl enc -aes-256-cbc -in FileInChiaro.rtf -out FileCifratoB64.rtf  
-e -base64 -pass pass:password.txt
```



FileCifratoB64.rtf

```
U2FsdGVkX18hL9rC2FLmVrrlpI8xl6/d02wFVPnlwMvNU96sAh5kILpsTmKr/sf0N5fiH7OwijixKscWQXacxwB058ow1cS6YmE  
L88mpwXHZVTYUr5A0jbU+t9UgAVQN5YIrw7gq3Xohl+0xgVOMY4YRq3N/kU++S3yUxt73iM6x/iXGd1LSPoMf3mP84ZXLou  
OUYDXxogGr9iR9GdxKsM/dkAR7TWsWhqSqY0ATHGTAof46qemlnhSxuWz1Sq6GskOKBDB1f5sCTxaRqZG3/u88pVmbRRU  
J0J0R7zWfBtkGRrat8a00U6re7jiMaHC6PpMt3MWD6EQYmUiEqIjKkP+VpYy5egwZdBBCUuvu8+b2M2n16wuRyfstNrYkD9s  
cYowwoXYs26yNKQQSD7fMRZ/nmt2evFIPn4NM7B95lgbck40aDHqAY6l/+G/kJQfPCNQd5X2wRGL1mVTrBUvJ/BI8Zuockf  
fjDAf9tj6efUhdgJbCa0KaJHdbysjGa2dDHlc1rOYu8slvDvufkhCG8UcAfOmQ73M6b0G01285lZNQRYuvw0zOf9vw4PnWb6w  
VxWHNMMyPn4F3eJrtv5C2pN9RQON/VPfByWEN/E8n1dtUTpF/VCS06MHk5hynVIgTpaKWlcerCMcEgU3ucW6NRmgI5Db  
oguGJReFxPAkGGcumgi3PwpMd6wVi8vR42rn6uyztffKpmwf9MpQDxsSBn5Fk7oRsv5aP6aCWGg2n0U3Ah7chbDBpljmdcLjcUrr  
VuEKqa/uN8vH72LLwBHLGKfSjQweu4R4nTTzXdkPgDQuOf9y+xRt2gRzx/FIRY8d3kISMvzSiXSE1Gc2KW02ATvxrlB+fq0Jeq  
2Pou3zGEVUchyBIE2VNfONMjuc3O7/qvJjK46rrBRNiC3sBQq1eU28XZPHPWBSL2BObAROG6UqBi2yX+QBp9DH0fZ6uiKcai  
OFmKS9nPSsePmJO18qXhsBnUVDt+AcedtVv/9tKlaBrXt1Ff41HFxKURqwwJ2tFb2xVXkXUUYxBAYhpixnMNq6tKgScR/F5RzM  
GdbAY2V+jOg0Q4pggjvOH1Vjan3N9gV60PqdH51IAhY80HXpg==
```

Cifratura Simmetrica in OpenSSL

Esempio 3

FileInChiaro.rtf

Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia e aspra e forte che nel pensier rinova la paura! Tant'  amara che poco   pi  morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.



Cifratura con password letta da file e Codifica in Base64

```
openssl enc -aes-256-cbc -in FileInChiaro.rtf -out FileCifratoB64.rtf  
-e -base64 -pass pass:password.txt
```

Osservazione:

- Il file contenente la password per la cifratura (password.txt nell'esempio) deve trovarsi nella current working directory
- Altrimenti verr  usata come password la stringa che segue il parametro `-pass pass:`
 - Nell'esempio «password.txt»

Cifratura Simmetrica in OpenSSL

Esempio 4

FileCifratoB64.rtf

```
U2FsdGVkX18hL9rC2FLmVrrlpixl6/d02wFVPnlwMvNU96sAh5kILpsTmKr/sf0N5fiH7OwijixKscWQXacxwB058ow1cS6YmE
L88mpwXHZVTYUr5A0jbU++9UgAVQN5YIrw7gq3Xohl+0xgVOMY4YRq3N/kU++S3yUxt73iM6x/iXGd1LSPoMf3mP84ZXLOU
OUYDXxogGr9iR9GdxKsM/dkAR7TWsWhqSqY0ATHGTAof46qemlnhSxuWz1Sg6GskOKBDB1f5sCTxaRqZG3/u88pVmbRRU
J0J0R7zWfB+kGRrat8a00U6re7jiMaHC6PpM+3MWD6EQYmUiEqIjKkP+VpYy5egwZdBBCUuvu8+b2M2n16wuRyfstNrYkD9s
cYowwoXYs26yNKQQSD7fMRZ/nmt2evFIPn4NM7B95lgbck40aDHqAY6l/+G/kJQfPCNQd5X2wRGL1mVTrBUvJ/BI8Zuockf
fjDAf9+j6efUhdgJbCa0KaJHdbysjGa2dDHlc1rOYu8slvDvufkhCG8UcAfOmQ73M6b0G01285lZNQRYuvw0zOf9vw4PnWb6w
VxWHNMMYpN4F3eJrtv5C2pN9RQON/VPfByWEN/E8n1dtUTpF/VCS06MHk5hynVIgTpaKWlcerCMcEgU3ucW6NRmgI5Db
oguGJReFXpAkGGcumgi3PwpMd6wVi8vR42rn6uyztffKpmwf9MpQDxsSBn5Fk7oRsv5aP6aCWGg2n0U3Ah7chbDBpljmdcLjcUrr
VuEKqa/uN8vH72LLwBHLgKfSjQweu4R4nTTzXdkPgDQuOf9y+xRt2gRzx/FIRY8d3klSMvzSiXSE1Gc2KW02ATvxrlB+fq0Jeq
2Pou3zGEVUchyBIE2VNfONMjuc3O7/qvJjK46rrBRNiC3sBQq1eU28XZPHWPBSEL2BOBAROG6UqBi2yX+QBp9DH0fZ6uiKcai
OFmKS9nPSSePmJO18qXhsBnUVDt+AcedtVv/9tklaBrXt1Ff41HFxKURqvwJ2tFb2xVXkXUUYxBAYhpxnMNq6tkgScR/F5RzM
GdbAY2V+jOg0Q4pggjvOH1Vjan3N9gV60PqdH51IAhY80HXpg==
```



Decifratura del file codificato in Base 64, con password letta da file

```
openssl enc -aes-256-cbc -in FileCifratoB64.rtf -out FileDecifrato.rtf
-d -base64 -pass pass:password.txt
```



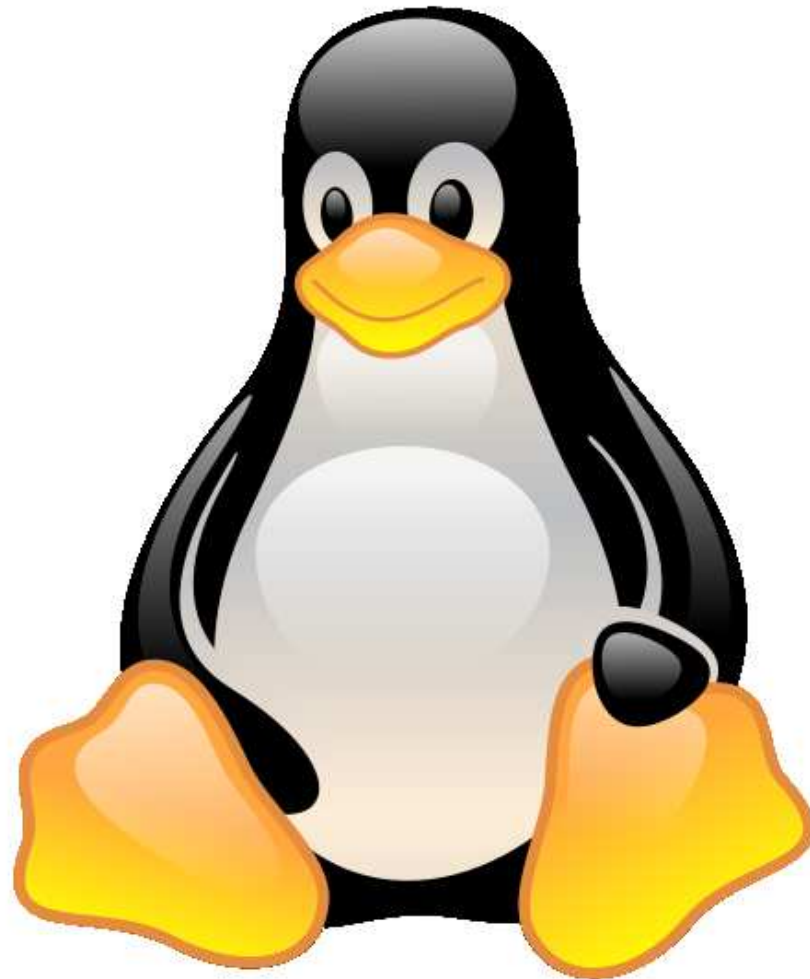
FileDecifrato.rtf

Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ch  la diritta via era smarrita. Ahi quanto a dir qual era   cosa dura esta selva selvaggia e aspra e forte che nel pensier rinova la paura! Tant'  amara che poco   pi  morte; ma per trattar del ben ch'i' vi trovai, dir  de l'altre cose ch'i' v'ho scorte.

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

tux.bmp



Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

- Il file `tux.bmp` contiene un'immagine bitmap
- Cifriamo tale file usando AES con le modalità ECB e CBC, usando una chiave a 128 bit

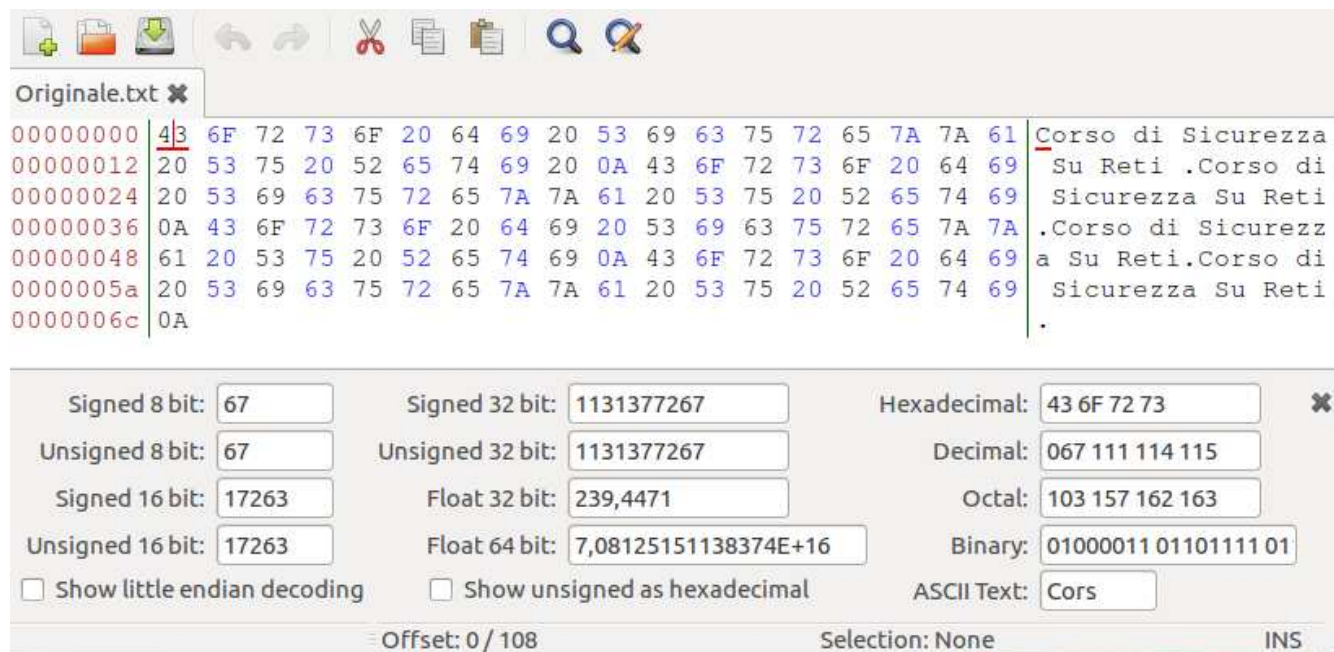
```
openssl enc -aes-128-ecb -in tux.bmp -out tux_aes_128_ecb.bmp  
-e -pass pass:P1pp0B4ud0
```

```
openssl enc -aes-128-cbc -in tux.bmp -out tux_aes_128_cbc.bmp  
-e -pass pass:P1pp0B4ud0
```

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

- Visualizziamo i contenuti dei file ottenuti dalla cifratura
 - N.B. Per visualizzare il contenuto di un'immagine bitmap cifrata è necessario ripristinare il relativo header, contenuto nei primi 54 byte dell'immagine originaria (`tux.bmp`)
 - Per farlo useremo l'editor esadecimale Bless



Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

- Siano
 - `tux_aes_128_ecb.bmp` il file contenente la cifratura di `tux.bmp` con AES a 128 bit in modalità ECB
 - `tux_aes_128_cbc.bmp` il file contenente la cifratura di `tux.bmp` con AES a 128 bit in modalità CBC
- Usando Bless ripristiniamo l'header bitmap dell'immagine cifrata
 - Apriamo sia il file `tux.bmp` che il file `tux_aes_128_ecb.bmp`
 - Copiamo i primi 54 byte del file `tux.bmp` in quelli corrispondenti del file `tux_aes_128_ecb.bmp`
 - La stessa operazione deve essere fatta per il file `tux_aes_128_cbc.bmp`

Esempio 5 - Cifratura di Immagini Bitmap

tux.bmp x tux_aes_128_ecb.bmp x

00000000	42 4D D6 BB 0D 00 00 00 00 00 36 00 00 00 28 00 00 00	BM.....6... (....
00000012	F4 01 00 00 58 02 00 00 01 00 18 00 00 00 00 00 A0 BBX.....
00000024	0D 00 13 0B 00 00 13 0B 00 00 00 00 00 00 00 00 00 00
00000036	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000048	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Signed 8 bit: -1 Signed 32 bit: -1 Hexadecimal: FF FF FF FF x
 Unsigned 8 bit: 255 Unsigned 32 bit: 4294967295 Decimal: 255 255 255 255
 Signed 16 bit: -1 Float 32 bit: NaN Octal: 377 377 377 377
 Unsigned 16 bit: 65535 Float 64 bit: NaN Binary: 11111111 11111111 11
☐ Show little endian decoding ☐ Show unsigned as hexadecimal ASCII Text: ???

Offset: 54 / 900053 Selection: 0 to 53 (54 bytes) INS

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

The screenshot shows a hex editor interface. At the top, a toolbar contains icons for file operations and editing. Below the toolbar, a list of files is visible, with 'tux' selected. The main area displays a hex dump of a file. A blue rectangular selection box highlights a portion of the data, specifically the BMP file header. The selected area contains the following hex values: 00 00 28 00 00 00, 00 00 00 00 A0 BB, 00 00 00 00 00 00, FF FF FF FF FF FF, and FF FF FF FF FF FF. To the right of the hex dump, the corresponding ASCII representation is shown, including 'BM...', '...', 'X...', and '...'. Below the hex dump, there are several input fields for data conversion: Unsigned 8 bit (255), Signed 16 bit (-1), Unsigned 16 bit (65535), Unsigned 32 bit (7295), Float 32 bit (NaN), Float 64 bit (NaN), Hexadecimal (FF FF FF FF), Decimal (255 255 255 255), Octal (377 377 377 377), Binary (11111111 11111111 11), and ASCII Text (????). At the bottom, the status bar shows 'Offset: 54 / 900053' and 'Selection: 0 to 53 (54 bytes)'. A blue callout box points to the selection area, containing the following text:

➤ **N.B.** Assicurarsi che la selezione avvenga utilizzando l'indicizzazione in formato decimale

➤ Ciò può essere ottenuto cliccando sull'area indicata dal rettangolo in azzurro

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

The screenshot displays a hex editor interface with two tabs: `tux.bmp` and `tux_aes_128_ecb.bmp`. The `tux.bmp` tab is active, showing the raw bytes of the image file. A green box highlights the first 54 bytes of the file, which correspond to the BMP header. A green arrow points from this box to a text label that reads "Header bitmap di 54 byte". The hex data is displayed in columns, with the first 54 bytes (00000000 to 00000048) highlighted in orange. The rest of the file is filled with `FF` bytes, indicating encryption. The bottom of the editor shows various data type interpretations (Signed 8 bit, Unsigned 8 bit, Signed 16 bit, Unsigned 16 bit, Signed 32 bit, Unsigned 32 bit, Float 32 bit, Float 64 bit, Decimal, Octal, Binary, ASCII Text) and a status bar indicating the current offset (54 / 900053) and the selected range (0 to 53 (54 bytes)).

Offset	Hex	ASCII
00000000	42 4D D6 BB 0D 00 00 00 00 00 00 36 00 00 00 28 00 00 00	BM.....6....(...
00000012	F4 01 00 00 58 02 00 00 01 00 18 00 00 00 00 00 A0 BB	...X.....
00000024	0D 00 13 0B 00 00 13 0B 00 00 00 00 00 00 00 00 00 00
00000036	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000048	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Header bitmap di 54 byte

Offset: 54 / 900053 Selection: 0 to 53 (54 bytes) INS

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

The screenshot shows a hex editor window with the file `tux.bmp` open. A selection of 54 bytes is highlighted in orange. A context menu is open over the selection, showing options: `Perform Operation`, `Taglia`, `Copia` (highlighted), `Incolla`, and `Elimina`.

The hex data is as follows:

Offset	Hex	ASCII
00000000	42 4D D6 BB 0D 00 00 00 00 00 00 36 00 00 00 28 00 00 00	BM.....6... (...
00000012	F4 01 00 00 58 02 00 00 00 00 00 00 00 00 A0 BBX.....
00000024	0D 00 13 0B 00 00 1
00000036	FF FF FF FF FF FF FF
00000048	FF FF FF FF FF FF FF
0000005a	FF FF FF FF FF FF FF
0000006c	FF FF FF FF FF FF FF
0000007e	FF FF FF FF FF FF FF

The bottom panel shows the conversion of the selected bytes (hex `FF FF FF FF`) to various formats:

Format	Value
Signed 8 bit:	-1
Unsigned 8 bit:	255
Signed 16 bit:	-1
Unsigned 16 bit:	65535
Signed 32 bit:	-1
Unsigned 32 bit:	4294967295
Float 32 bit:	NaN
Float 64 bit:	NaN
Hexadecimal:	FF FF FF FF
Decimal:	255 255 255 255
Octal:	377 377 377 377
Binary:	11111111 11111111 11
ASCII Text:	????

Offset: 54 / 900053 Selection: 0 to 53 (54 bytes) INS

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

tux.bmp ✕ tux_aes_128_ecb.bmp ✕

00000000	53 61 6C 74 65 64 5F 5F 63 FF 6D 44 62 F7 E1 77 DB 46	Salted__c.mDb..w.F
00000012	BD D1 82 B4 2A 4F C2 3D 6F 03 95 EF B6 9E 7C 91 DC FC*O.=o..... ...
00000024	76 05 34 7A FC 02 62 52 73 9A 7E 5B 04 C0 3D 1F FA 25	v.4z..bRs.~[..=..%
00000036	FA B4 BF EB 15 0D 5D BF 6F 5A 55 18 4A F3 23 A4 9E E4].oZU.J.#...
00000048	D5 CA 25 FA A3 B7 5A B1 A8 BE EC E9 9C C2 A1 0D 7B A5	..%...Z.....{.

Signed 8 bit:	-6	Signed 32 bit:	-88817685	Hexadecimal:	FA B4 BF EB ✕
Unsigned 8 bit:	250	Unsigned 32 bit:	4206149611	Decimal:	250 180 191 235
Signed 16 bit:	-1356	Float 32 bit:	-4,69253E+35	Octal:	372 264 277 353
Unsigned 16 bit:	64180	Float 64 bit:	-1,20527836817238E+283	Binary:	11111010 10110100 10
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text:	????

Offset: 54 / 900079 Selection: 0 to 53 (54 bytes) INS

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

The screenshot shows a hex editor interface with the following components:

- File List:** `tux.bmp` and `tux_aes_128_ecb.bmp` are visible in the top bar.
- Hex View:** The main area displays hexadecimal data. A selection of 54 bytes (offset 54 to 900079) is highlighted in orange. The data includes the ASCII string "Salted__c.mDb..w.F" and other binary data.
- Context Menu:** A menu is open over the selection, showing options: "Perform Operation", "Taglia", "Copia", "Incolla", and "Elimina".
- Data Representation Panel:** The bottom section shows various interpretations of the selected data:
 - Signed 8 bit: `-6`
 - Unsigned 8 bit: `250`
 - Signed 16 bit: `-1356`
 - Unsigned 16 bit: `64180`
 - Signed 32 bit: `-88817085`
 - Unsigned 32 bit: `4206149611`
 - Float 32 bit: `-4,69253E+35`
 - Float 64 bit: `-1,20527836817238E+283`
 - Hexadecimal: `FA B4 BF EB`
 - Decimal: `250 180 191 235`
 - Octal: `372 264 277 353`
 - Binary: `11111010 10110100 10`
 - ASCII Text: `????`
- Footer:** The status bar shows "Offset: 54 / 900079", "Selection: 0 to 53 (54 bytes)", and "INS".

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

The screenshot shows a hex editor with two tabs: `tux.bmp` and `tux_aes_128_ecb.bmp*`. The active tab displays the hex dump of the encrypted file. A green box highlights the first 54 bytes (from offset 00000000 to 00000048). A green arrow points from a text box labeled "Header bitmap di 54 byte" to this highlighted area.

Offset	Hex	ASCII
00000000	42 4D D6 BB 0D 00 00 00 00 00 00 36 00 00 00 28 00 00 00	BM.....6... (...
00000012	F4 01 00 00 58 02 00 00 01 00 18 00 00 00 00 00 A0 BBX.....
00000024	0D 00 13 0B 00 00 13 0B 00 00 00 00 00 00 00 00 00 00
00000036	FA B4 BF EB 15 0D 5D BF 6F 5A 55 18 4A F3 23 A4 9E E4].oZU.J.#...
00000048	D5 CA 25 FA A3 B7 5A B1 A8 B1 F9 9C C2 A1 0D 7B A5	..%...Z.....{.

Below the hex dump, various data interpretation fields are shown:

- Signed 8 bit: -6
- Unsigned 8 bit: 250
- Signed 16 bit: -1356
- Unsigned 16 bit: 64180
- Signed 32 bit: -8
- Unsigned 32 bit: 4206149611
- Float 32 bit: -4,69253E+35
- Float 64 bit: -1,20527836817238E+283
- Decimal: 250 180 191 235
- Octal: 372 264 277 353
- Binary: 11111010 10110100 10
- ASCII Text: ????

At the bottom, the status bar shows: Offset: 54 / 900079, Selection: None, and INS.

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

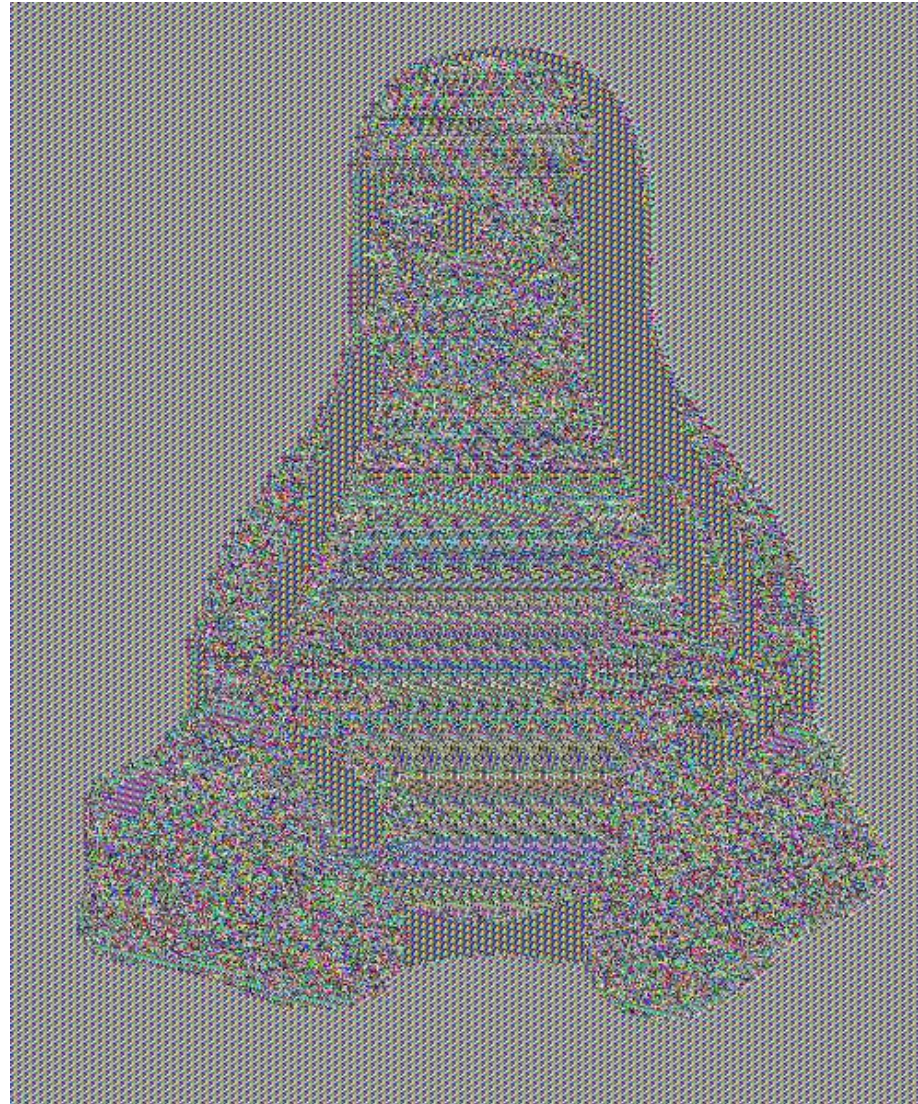
The screenshot shows a hex editor interface with the following components:

- Toolbar:** Includes icons for file operations (add, delete, save, open, copy, paste, find, replace) and a 'Save File' button that is highlighted with a red box.
- File List:** Shows two files: 'tux.bmp' and 'tux_aes_128_ecb.bmp*'. The second file is selected.
- Hex Data:** Displays a hex dump of the file. The first few bytes are '42 4D D6 BB 0D 00 00 00 00 00 36 00 00 00 28 00 00 00', which correspond to the BMP header. The ASCII column shows 'BM.....6....(....X.....]..ZU.J.#...%...Z.....{.'. The word 'tux' is visible in the ASCII column starting at offset 0x36.
- Conversion Panel:** Shows various data representations for the selected hex value 'FA B4 BF EB':
 - Signed 8 bit: -6
 - Unsigned 8 bit: 250
 - Signed 16 bit: -1356
 - Unsigned 16 bit: 64180
 - Signed 32 bit: -88817685
 - Unsigned 32 bit: 4206149611
 - Float 32 bit: -4,69253E+35
 - Float 64 bit: -1,20527836817238E+283
 - Hexadecimal: FA B4 BF EB
 - Decimal: 250 180 191 235
 - Octal: 372 264 277 353
 - Binary: 11111010 10110100 10
 - ASCII Text: ????
- Footer:** Shows 'Offset: 54 / 900079' and 'Selection: None'.

Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

tux_aes_128_ecb.bmp



Cifratura Simmetrica in OpenSSL

Esempio 5 - Cifratura di Immagini Bitmap

tux_aes_128_cbc.bmp



Cifratura Simmetrica in OpenSSL

Comando speed

- Il comando `speed` permette di effettuare test prestazionali per gli algoritmi di
 - Cifratura Simmetrica
 - Message Digest

Sintassi generale del comando

```
openssl speed [-help] [-engine id] [-elapsed] [-evp algo] [-  
decrypt] [algorithm...]
```



Cifratura Simmetrica in OpenSSL

Comando speed

- Il comando `speed` permette di effettuare test prestazionali per gli algoritmi di
 - Cifratura Simmetrica
 - Message Digest

Sintassi generale del comando

```
openssl speed [-help] [-engine id] [-elapsed] [-evp algo] [-  
decrypt] [algorithm...]
```

Per ottenere la lista completa
delle opzioni del comando
`speed` è possibile utilizzare
`man speed`



Cifratura Simmetrica in OpenSSL

Comando speed

- Se al comando speed viene passato come parametro il nome di un algoritmo, esso valuta le prestazioni di tale algoritmo
- Altrimenti valuta tutti gli algoritmi supportati dalla versione corrente di OpenSSL
- N.B. I risultati variano in base alle caratteristiche hardware dell'ambiente su cui tale comando è eseguito

Cifratura Simmetrica in OpenSSL

Comando speed - Esempio 1

```
$ openssl speed des
Doing des cbc for 3s on 16 size blocks: 15246862 des cbc's in 2.99s
Doing des cbc for 3s on 64 size blocks: 3873573 des cbc's in 3.00s
Doing des cbc for 3s on 256 size blocks: 969465 des cbc's in 3.00s
Doing des cbc for 3s on 1024 size blocks: 235959 des cbc's in 3.00s
Doing des cbc for 3s on 8192 size blocks: 28920 des cbc's in 3.00s
Doing des cbc for 3s on 16384 size blocks: 14870 des cbc's in 3.01s
Doing des ede3 for 3s on 16 size blocks: 5749558 des ede3's in 3.00s
Doing des ede3 for 3s on 64 size blocks: 1442742 des ede3's in 3.01s
Doing des ede3 for 3s on 256 size blocks: 360843 des ede3's in 2.99s
Doing des ede3 for 3s on 1024 size blocks: 90662 des ede3's in 2.99s
Doing des ede3 for 3s on 8192 size blocks: 11398 des ede3's in 3.00s
Doing des ede3 for 3s on 16384 size blocks: 5699 des ede3's in 3.01s
OpenSSL 1.1.0g  2 Nov 2017
```

The 'numbers' are in 1000s of bytes per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes	16384 bytes
des cbc	81588.56k	82636.22k	82727.68k	80540.67k	78970.88k	80940.23k
des ede3	30664.31k	30676.24k	30894.92k	31049.46k	31124.14k	31020.74k

Cifratura Simmetrica in OpenSSL

Comando speed - Esempio 1

```
$ openssl speed des
Doing des cbc for 3s on 16 size blocks: 15246862 des cbc's in 2.99s
Doing des cbc for 3s on 64 size blocks: 3872572 des cbc's in 3.00s
Doing des cbc for 3s on 256 size blocks: 90662 des cbc's in 3.00s
Doing des cbc for 3s on 1024 size blocks: 11398 des cbc's in 3.00s
Doing des cbc for 3s on 8192 size blocks: 5699 des cbc's in 3.01s
Doing des cbc for 3s on 16384 size blocks: 5699 des cbc's in 3.01s
Doing des ede3 for 3s on 16 size blocks: 81588.56k des ede3's in 3.00s
Doing des ede3 for 3s on 64 size blocks: 82636.22k des ede3's in 3.01s
Doing des ede3 for 3s on 256 size blocks: 82727.68k des ede3's in 2.99s
Doing des ede3 for 3s on 1024 size blocks: 80540.67k des ede3's in 2.99s
Doing des ede3 for 3s on 8192 size blocks: 78970.88k des ede3's in 3.00s
Doing des ede3 for 3s on 16384 size blocks: 80940.23k des ede3's in 3.01s
OpenSSL 1.1.0g  2 Nov 2017
```

Per ottenere la lista completa
delle opzioni del comando
speed è possibile utilizzare
man speed

The 'numbers' are in 1000s of bytes per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes	16384 bytes
des cbc	81588.56k	82636.22k	82727.68k	80540.67k	78970.88k	80940.23k
des ede3	30664.31k	30676.24k	30894.92k	31049.46k	31124.14k	31020.74k

Cifratura Simmetrica in OpenSSL

Comando speed - Esempio 1

```
$ openssl speed des
```

```
Doing des cbc for 3s on 16 size blocks: 15246862 des cbc's in 2.99s
```

```
Doing des cbc for 3s on 64 size blocks: 3873572 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 256 size blocks: 969465 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 1024 size blocks: 235955 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 8192 size blocks: 28920 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 16384 size blocks: 14870 des cbc's in 3.00s
```

```
Doing des ede3 for 3s on 16 size blocks: 574955 des ede3's in 2.99s
```

```
Doing des ede3 for 3s on 64 size blocks: 144274 des ede3's in 2.99s
```

```
Doing des ede3 for 3s on 256 size blocks: 360843 des ede3's in 2.99s
```

```
Doing des ede3 for 3s on 1024 size blocks: 90662 des ede3's in 2.99s
```

```
Doing des ede3 for 3s on 8192 size blocks: 11398 des ede3's in 3.00s
```

```
Doing des ede3 for 3s on 16384 size blocks: 5699 des ede3's in 3.01s
```

```
OpenSSL 1.1.0g  2 Nov 2017
```

The 'numbers' are in 1000s of bytes per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes	16384 bytes
des cbc	81588.56k	82636.22k	82727.68k	80540.67k	78970.88k	80940.23k
des ede3	30664.31k	30676.24k	30894.92k	31049.46k	31124.14k	31020.74k

In 2.99 secondi OpenSSL effettua 15246862 cifrature des-cbc su blocchi in input di 16 byte

Cifratura Simmetrica in OpenSSL

Comando speed - Esempio 1

```
$ openssl speed des
```

```
Doing des cbc for 3s on 16 size blocks: 15246862 des cbc's in 2.99s
```

```
Doing des cbc for 3s on 64 size blocks: 3873573 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 256 size blocks: 969465 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 1024 size blocks: 235959 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 8192 size blocks: 29496 des cbc's in 3.00s
```

```
Doing des cbc for 3s on 16384 size blocks: 14748 des cbc's in 3.00s
```

```
Doing des ede3 for 3s on 16 size blocks: 15246862 des ede3's in 2.99s
```

```
Doing des ede3 for 3s on 64 size blocks: 3873573 des ede3's in 3.00s
```

```
Doing des ede3 for 3s on 256 size blocks: 969465 des ede3's in 3.00s
```

```
Doing des ede3 for 3s on 1024 size blocks: 235959 des ede3's in 3.00s
```

```
Doing des ede3 for 3s on 8192 size blocks: 29496 des ede3's in 3.00s
```

```
Doing des ede3 for 3s on 16384 size blocks: 14748 des ede3's in 3.00s
```

```
OpenSSL 1.1.0g  2 Nov 2017
```

➤ Qui sono riportate le velocità di cifratura in Byte al secondo

➤ Se in 2.99 secondi sono effettuate 15246862 cifrature des-cbc di blocchi in input di 16 byte, la velocità riportata nella tabella sarà pari a $(15246862 \times 16) \div 2.99 \approx 81588.56$

The 'numbers' are in 1000s of blocks per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes	16384 bytes
des cbc	81588.56k	82636.22k	82727.68k	80540.67k	78970.88k	80940.23k
des ede3	30664.31k	30676.24k	30894.92k	31049.46k	31124.14k	31020.74k

OpenSSL

Comando speed - Esempio 2

```
$ openssl speed aes
Doing aes-128 cbc for 3s on 16 size blocks: 27738716 aes-128 cbc's in 2.99s
Doing aes-128 cbc for 3s on 64 size blocks: 7672032 aes-128 cbc's in 2.98s
Doing aes-128 cbc for 3s on 256 size blocks: 1939616 aes-128 cbc's in 2.97s
Doing aes-128 cbc for 3s on 1024 size blocks: 1071954 aes-128 cbc's in 2.99s
Doing aes-128 cbc for 3s on 8192 size blocks: 133691 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 16384 size blocks: 68005 aes-128 cbc's in 3.00s
Doing aes-192 cbc for 3s on 16 size blocks: 23469929 aes-192 cbc's in 2.99s
Doing aes-192 cbc for 3s on 64 size blocks: 6449570 aes-192 cbc's in 2.99s
Doing aes-192 cbc for 3s on 256 size blocks: 1636439 aes-192 cbc's in 2.99s
Doing aes-192 cbc for 3s on 1024 size blocks: 917697 aes-192 cbc's in 3.00s
Doing aes-192 cbc for 3s on 8192 size blocks: 114689 aes-192 cbc's in 3.00s
Doing aes-192 cbc for 3s on 16384 size blocks: 57357 aes-192 cbc's in 2.99s
Doing aes-256 cbc for 3s on 16 size blocks: 20485571 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 64 size blocks: 5486029 aes-256 cbc's in 2.99s
Doing aes-256 cbc for 3s on 256 size blocks: 1395591 aes-256 cbc's in 2.99s
Doing aes-256 cbc for 3s on 1024 size blocks: 730877 aes-256 cbc's in 2.99s
Doing aes-256 cbc for 3s on 8192 size blocks: 93914 aes-256 cbc's in 2.95s
Doing aes-256 cbc for 3s on 16384 size blocks: 48430 aes-256 cbc's in 2.98s
OpenSSL 1.1.0g  2 Nov 2017
```

The 'numbers' are in 1000s of bytes per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes	16384 bytes
aes-128 cbc	148434.60k	164768.47k	167185.76k	367117.36k	365065.56k	371397.97k
aes-192 cbc	125591.59k	138051.00k	140109.83k	313240.58k	313177.43k	314293.34k
aes-256 cbc	109256.38k	117426.71k	119488.73k	250307.04k	260794.40k	266267.49k

OpenSSL Stream Cipher RC4

- L'unico Stream Cipher supportato da OpenSSL è RC4
- OpenSSL fornisce 2 varianti di RC4
 - 40 bit
 - Accessibile mediante il ciphername `-rc4-40`
 - 128 bit
 - Accessibile mediante il ciphername `-rc4`
- Su alcune versioni di OpenSSL è anche fornita la variante a 64 bit
 - Accessibile mediante il ciphername `-rc4-64`

Crittografia Simmetrica in OpenSSL

Stream Cipher RC4 - Esempio

- Mediante il seguente comando è possibile cifrare il file **file.txt** usando RC4 con chiave a 128 bit

```
openssl enc -rc4 -e -in file.txt -out file_rc4_encrypted
```

OpenSSL

RC4 - Esempio 1

Mediante il seguente comando è possibile cifrare il file **file.txt** usando RC4 con chiave a 128 bit

```
openssl enc -rc4 -e -in file.txt -out file_rc4_encrypted
```

È possibile utilizzare le stesse opzioni viste per i cifrari a blocchi.

➤ Ad es., è possibile cifrare **file.txt** specificando la password da cui derivare la chiave, senza salt e stampando il valore della chiave effettivamente utilizzata

```
openssl enc -rc4 -k P1pp0B4ud0 -nosalt -p -e -in  
file.txt -out file_rc4_encrypted
```



```
key=6565760B203EECCDF87EB0BF58289A79
```

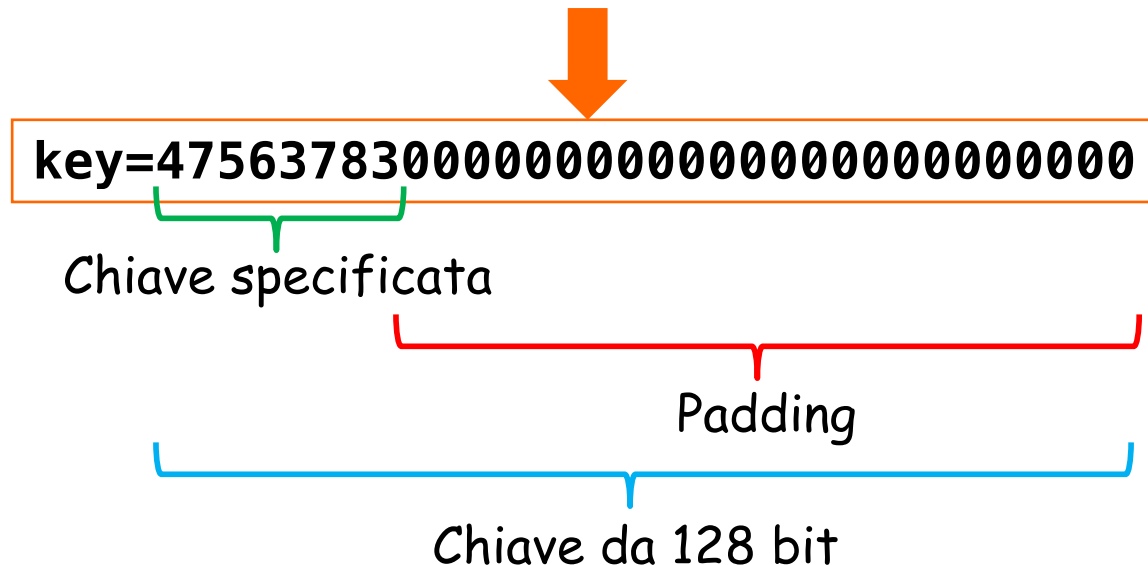
Chiave da 128 bit

הנהגה ופיקוד

È possibile utilizzare le stesse opzioni viste per i cifrari a blocchi.

- Ad es., è possibile cifrare file.txt specificando la chiave da utilizzare per la cifratura

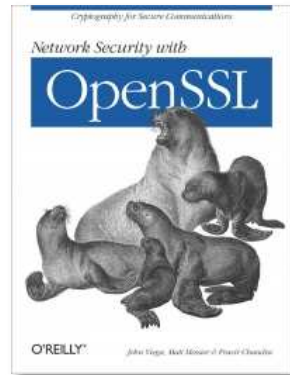
```
openssl enc -rc4 -K 47563783 -nosalt -p -e -in file.txt  
-out file_rc4_encrypted
```



Bibliografia

- **Network Security with OpenSSL**
Pravir Chandra, Matt Messier and John Viega (2002),
O'Reilly

- Cap. 2.1 e 2.3



- **Documentazione su OpenSSL**
 - <https://www.openssl.org/docs/>

Bibliografia

- Presentazioni Lezioni Corso di Sicurezza, Prof. De Santis
 - Cifrari a Blocchi
 - Advanced Encryption Standard
 - Stream Ciphers