Cryptography 1

Matteo ROSSI
Politecnico di Torino

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Obiettivi

- Comprensione base dei problemi che la crittografia vuole risolvere
- Conoscenza della storia della crittografia
- Comprensione del concetto di segretezza perfetta, One-Time pad e dei relativi problemi
- Comprensione del concetto di cifrario simmetrico moderno
- Comprensione del funzionamento e delle differenze tra cifrari a flusso e cifrari a blocchi
- Comprensione delle modalità di funzionamento dei cifrari a blocchi

Prerequisiti

- Encoding e conversioni:
  - ASCII
  - Decimale
  - Binario
  - Esadecimale
- Matematica di base

Argomenti

- Introduzione
- Storia della crittografia
- Segretezza perfetta e one-time pad
- Stream ciphers
- Block ciphers e modalità di funzionamento

Argomenti

- Introduzione
- Storia della crittografia
- Segretezza perfetta e one-time pad
- Stream ciphers
- Block ciphers e modalità di funzionamento

Introduzione

- Il principale scopo della crittografia è quello di comunicare segretamente in presenza di "nemici"

Introduzione

- Il principale scopo della crittografia è quello di comunicare segretamente in presenza di "nemici"
- Con la crittografia possiamo raggiungere i seguenti obiettivi:
  - Confidenzialità
  - Integrità
  - Autenticazione
  - Non ripudio

Introduzione

- Confidenzialità: nessuno esterno alla comunicazione può leggere il contenuto del messaggio

Introduzione

- Integrità: se il messaggio è stato modificato durante la trasmissione, dev'essere possibile rilevarlo

Introduzione

- Autenticazione: l'identità delle parti in una comunicazione deve poter essere verificata

Introduzione

- Non ripudio: se una persona invia un messaggio specifico, non dev'essergli possibile negare di averlo fatto

Termini chiave

- *Cifrario* (*cipher*): algoritmo per eseguire operazioni per rendere oscuro (*cifrato*) un testo in chiaro (*cifratura*) o per ripristinare un messaggio precedentemente cifrato (*decifratura*)

Termini chiave

- *Cifrario* (*cipher*): algoritmo per eseguire operazioni per rendere oscuro (*cifrato*) un testo in chiaro (*cifratura*) o per ripristinare un messaggio precedentemente cifrato (*decifratura*)
- *Chiave:* segreto utilizzato nel cifrario per cifrare o decifrare messaggi

Termini chiave

- *Cifrario* (*cipher*): algoritmo per eseguire operazioni per rendere oscuro (*cifrato*) un testo in chiaro (*cifratura*), o per ripristinare un messaggio precedentemente cifrato (*decifratura*)
- *Chiave:* segreto utilizzato nel cifrario per cifrare o decifrare messaggi
- *Cifrario simmetrico:* cifrario che utilizza la *stessa chiave* per cifrare e decifrare

Principio di Kerckhoffs/Shannon

"La sicurezza di un sistema crittografico non deve dipendere dal tenere celato l'algoritmo crittografico ma solo dal tenere celata la chiave"

Crittografia simmetrica

- Anche chiamata crittografia a chiave privata
- Unico tipo di crittografia utilizzata fino agli anni ‘70
- Presuppone che le due parti siano già in possesso di una stessa chiave condivisa

Crittografia simmetrica

- Anche chiamata crittografia a chiave privata
- Unico tipo di crittografia utilizzata fino agli anni ‘70
- Presuppone che le due parti siano già in possesso di una stessa chiave condivisa
- Nel seguito, indichiamo $Enc(chiave, messaggio)$ l’operazione di cifratura e $Dec(chiave, messaggio)$ l’operazione di decifratura, omettendo eventualmente la chiave usata

Argomenti

- Introduzione
- Storia della crittografia
- Segretezza perfetta e one-time pad
- Stream ciphers
- Block ciphers e modalità di funzionamento

Cifrari a sostituzione (?? BC)

- Primi esempi già testimoniati nell'antichità
- Idea semplice: ogni carattere viene sostituito con un altro attraverso una tabella di corrispondenze
- La tabella è la chiave del cifrario

Cifrari a sostituzione (?? BC)

- Supponiamo di voler cifrare la stringa "abc":
  - $Enc(\text{abc}) = \text{fqt}$
- Per decifrare, leggiamo la tabella al contrario:
  - $Dec(\text{fqt}) = \text{abc}$

Chiaro	Cifrato
a	f
b	q
c	t
d	r
…	…
z	g

Cifrario di Cesare

- L'esempio più celebre di cifrario a sostituzione
- Non c'è una chiave: ogni lettera subisce uno shift di 3 posizioni
- Basta conoscere la regola di cifratura per rompere il cifrario

Chiaro	Cifrato
a	d
b	e
c	f
d	g
…	…
z	c

Attacchi ai cifrari a sostituzione

- I cifrari a sostituzione sono vulnerabili ad attacchi basati sull'analisi delle frequenze:
  - In italiano il carattere più frequente è il carattere "e":
    - nel testo cifrato, il carattere più frequente corrisponderà al carattere "e" con alta probabilità
  - Il secondo carattere più frequente è il carattere "a":
    - nel testo cifrato, il secondo carattere più frequente corrisponderà al carattere "a" con alta probabilità
  - E così via…

Cifrario di Vigenère (XVI secolo)

- Idea: combinare diversi cifrari a sostituzione

Cifrario di Vigenère (XVI secolo)

- Idea: combinare diversi cifrari a sostituzione
- Il testo viene diviso in blocchi di lunghezza fissata e per ogni blocco viene applicata una sostituzione diversa per ogni carattere

Cifrario di Vigenère (XVI secolo)

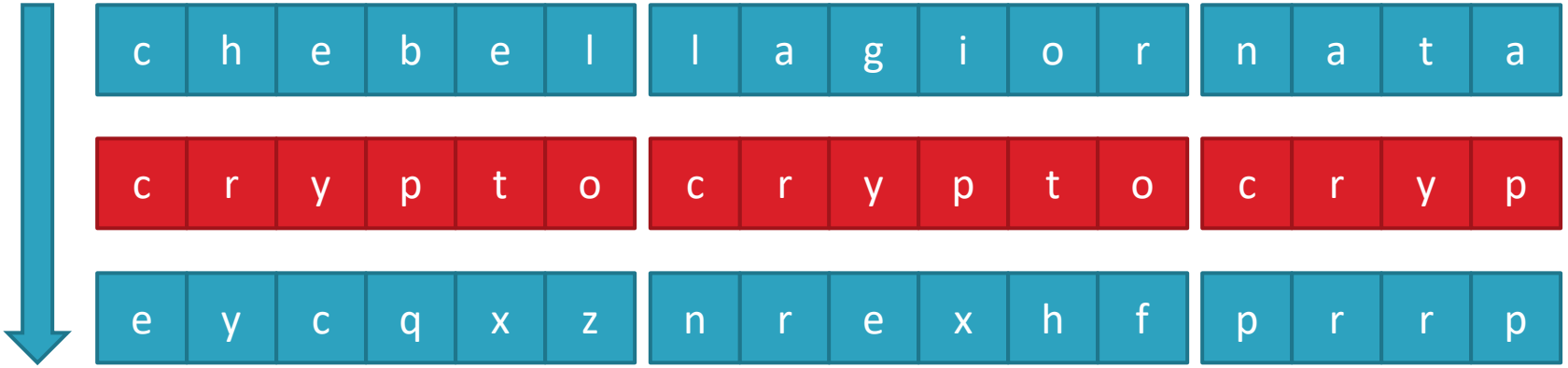
- Idea: combinare diversi cifrari a sostituzione
- Il testo viene diviso in blocchi di lunghezza fissata e per ogni blocco viene applicata una sostituzione diversa per ogni carattere
- Le sostituzioni sono shift come nel cifrario di Cesare, ma di lunghezze diverse

Cifrario di Vigenère (XVI secolo)

- Idea: combinare diversi cifrari a sostituzione
- Il testo viene diviso in blocchi di lunghezza fissata e per ogni blocco viene applicata una sostituzione diversa per ogni carattere
- Le sostituzioni sono shift come nel cifrario di Cesare, ma di lunghezze diverse
- La chiave è una parola, che raccoglie le lunghezze degli shift

Cifrario di Vigenère (XVI secolo)

- Esempio: vogliamo cifrare la frase "chebellagiornata" con la chiave "crypto":



Attacchi al cifrario di Vigenère

- La strategia è la stessa dei cifrari a sostituzione:
  - Ricaviamo la lunghezza della chiave (bruteforce o analisi statistica)
  - Risolviamo diversi cifrari a sostituzione indipendentemente

Macchine a rotori (1840-1950)

- Idea: cifrario a sostituzione con chiave ruotata a ogni cifratura

Chiaro	Cifrato
a	e
b	r
c	t
d	u
…	…
z	a

Step 1

Chiaro	Cifrato
a	a
b	e
c	r
d	t
…	…
z	v

Step 2

Chiaro	Cifrato
a	v
b	a
c	e
d	r
…	…
z	s

Step 3

Chiaro	Cifrato
a	s
b	v
c	a
d	e
…	…
z	l

Step 4

Cifrari moderni

- Data Encryption Standard (1974)
- Advanced Encryption Standard (2001)
- Salsa20 (2005)
- ChaCha20 (2008)

Argomenti

- Introduzione
- Storia della crittografia
- Segretezza perfetta e one-time pad
- Stream ciphers
- Block ciphers e modalità di funzionamento

Segretezza perfetta

- Quando possiamo dire che un cifrario è "*sicuro*"?
  - Idea informale: un cifrario è sicuro se il testo cifrato "*sembra casuale*", ovvero non dà nessuna informazione sul testo in chiaro
  - Questa proprietà è chiamata *segretezza perfetta* (perfect secrecy) e i cifrari che la raggiungono *cifrari perfetti*

Esempio: cifrari a sostituzione

- I cifrari a sostituzione non sono perfetti
- Viene mantenuta la frequenza delle lettere
- Il cifrario dà informazioni all'attaccante!

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua.

Sgktd ohlwd rgsgk lom adtm, egfltemtmwk aroholeo tsom, ltr rg towldgr mtdhgk ofeorwfm wm sazgkt tm rgsgkt daufa asojwa.

Exclusive-Or

L'*exclusive-or* (XOR) è un'operazione binaria indicata con $\oplus$, o con ^ in alcuni linguaggi di programmazione, con la seguente tabella di verità:

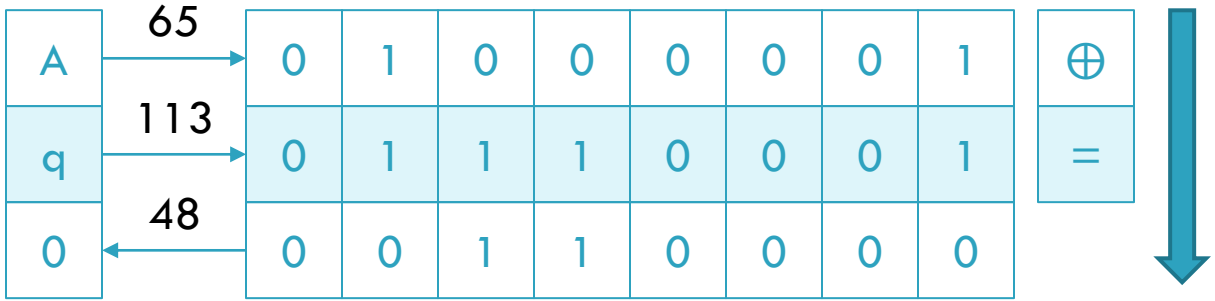
a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-Or

- In pratica, per fare lo XOR di due caratteri:
  - Convertiamo i caratteri (ASCII) in binario
  - Facciamo lo XOR verticalmente
  - Riconvertiamo il risultato

Exclusive-Or

- Esempio: "A"⊕"q" = "0"



Exclusive-Or

- Proprietà di base:
  - $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
  - $a \oplus b = b \oplus a$
  - $a \oplus a = 0$
  - $a \oplus 0 = a$
  - $a \oplus b \oplus a = b$

One-Time Pad (Vernam, 1917)

- Idea: usiamo lo XOR per cifrare messaggi
  - $Enc(k, m) = k \oplus m = c$
  - $Dec(k, c) = k \oplus c = m$
  - Questo cifrario viene chiamato *"One-Time Pad"* (OTP)
  - Nota: la chiave k dev'essere generata in maniera casuale

One-Time Pad (Vernam, 1917)

- Il One-Time Pad ha segretezza perfetta
  - Perché? Intuitivamente: ogni bit del risultato può essere 0 o 1 con la stessa probabilità!

One-Time Pad (Vernam, 1917)

- Il One-Time Pad ha sicurezza perfetta
  - Perché? Intuitivamente: ogni bit del risultato può essere 0 o 1 con la stessa probabilità!
- Assunzioni "scomode":
  - Dobbiamo avere una chiave lunga almeno quanto il testo
  - La chiave può essere utilizzata per una sola cifratura
  - Difficile utilizzarlo nella pratica

Attacchi a One-Time Pad

- Riutilizzare la chiave di un OTP non è sicuro:
  - Si possono estrarre informazioni dallo XOR dei cifrati
  - Si possono effettuare attacchi statistici:
    - https://github.com/CameronLonsdale/MTP
    - https://github.com/hellman/xortool

The bad news

Teorema: per avere segretezza perfetta la chiave dev'essere lunga almeno quanto il testo da cifrare

The bad news

Teorema: per avere segretezza perfetta la chiave dev'essere lunga almeno quanto il testo da cifrare

Nella pratica è *impossibile* ottenere segretezza perfetta

Argomenti

- Introduzione
- Storia della crittografia
- Segretezza perfetta e one-time pad
- Stream ciphers
- Block ciphers e modalità di funzionamento

Generatori Pseudocasuali

- Definizione: un generatore di numeri *pseudocasuali* (PRNG) è un algoritmo *deterministico* che genera numeri che "sembrano" casuali

Generatori Pseudocasuali

- In pratica:
  - Il generatore prende un numero "piccolo" (seed)
  - A partire da questo numero piccolo produce un flusso (stream) molto lungo di bit

Stream Ciphers

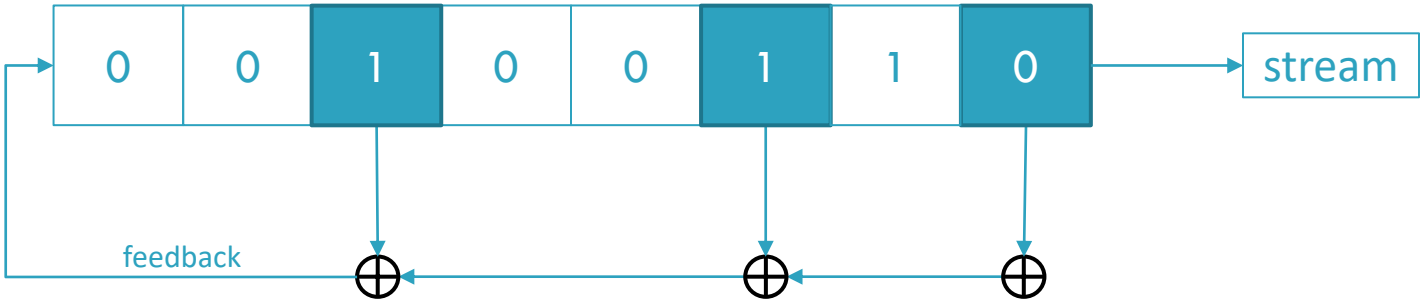
- Scopo: rendere pratico l'OTP
- Idea: usare generatori pseudocasuali per costruire la chiave dell'OTP

Stream Ciphers

- Osservazioni:
  - La sicurezza del cifrario dipende dalla sicurezza del PRNG
  - La lunghezza della chiave è la lunghezza del seed
  - In generale il seed è più corto del plaintext: non si può avere segretezza perfetta

Stream Ciphers

- Esempio: Linear Feedback Shift Register (LFSR)



Stream Ciphers

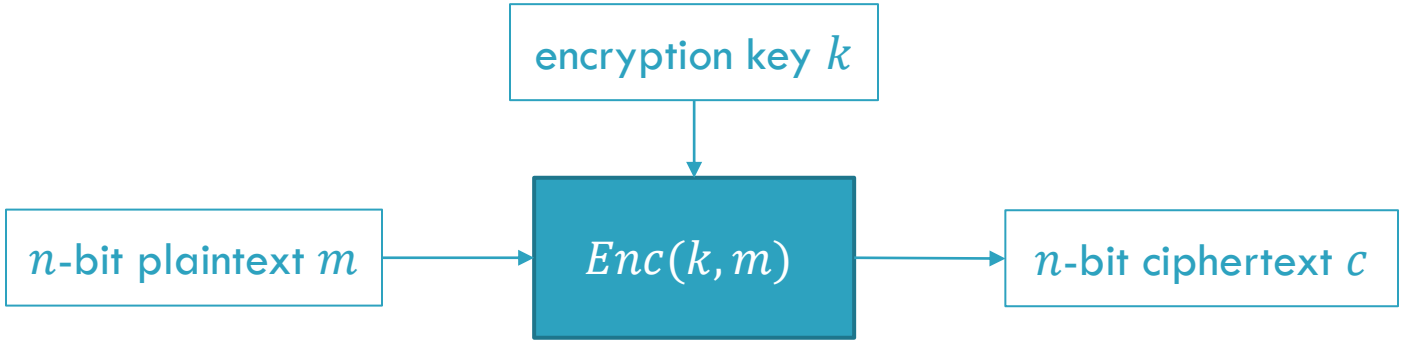
- Stream ciphers nel mondo reale:
  - RC4 (1987)
  - Salsa20 (2005)
  - ChaCha20 (2008)

Argomenti

- Introduzione
- Storia della crittografia
- Segretezza perfetta e one-time pad
- Stream ciphers
- Block ciphers e modalità di funzionamento

Block Ciphers

- Idea: cifrare blocchi di *lunghezza* (in bit) *fissata*

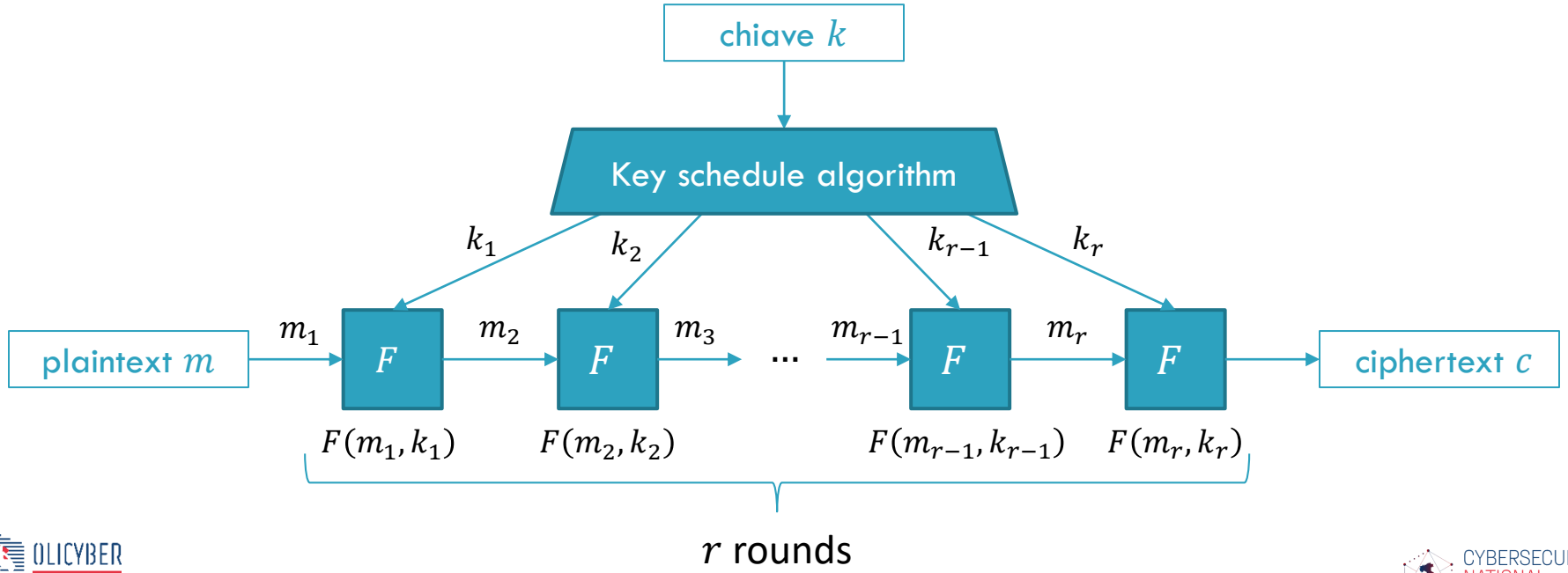


Block Ciphers

- Informalmente un block cipher è una keyed permutation:
  - È una permutazione sui possibili blocchi di n bit
    - Una funzione reversibile che associa ad ogni blocco un altro blocco in modo univoco
  - La permutazione è completamente determinata dalla chiave scelta

Block Ciphers

- Block ciphers in pratica: iterazione di funzioni semplici



Block Ciphers

- Block ciphers nel mondo reale:
  - DES (1974)
    - Blocchi da 64 bit
    - Chiavi da 56 bit
    - 16 round
  - AES (2001)
    - Blocchi da 128 bit
    - Chiavi da 128 a 256 bit
    - Da 10 a 14 round

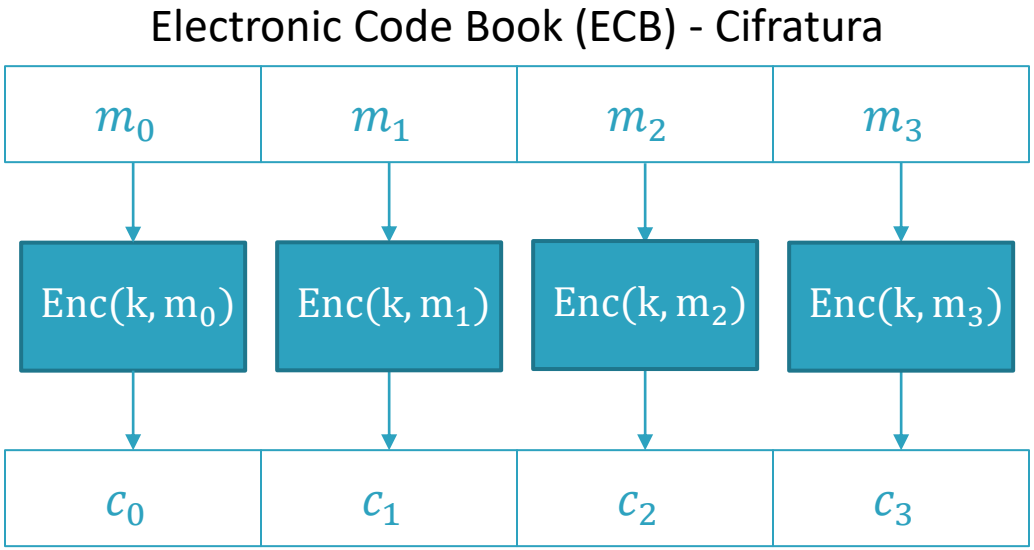
Modalità di funzionamento

- Come facciamo a cifrare dati più lunghi di n bit?
  - Idea: vogliamo far comportare un block cipher come uno stream cipher
  - I modi di farlo si chiamano *modalità di funzionamento* (modes of operation)
  - Le modalità di funzionamento sono *indipendenti dal cifrario*

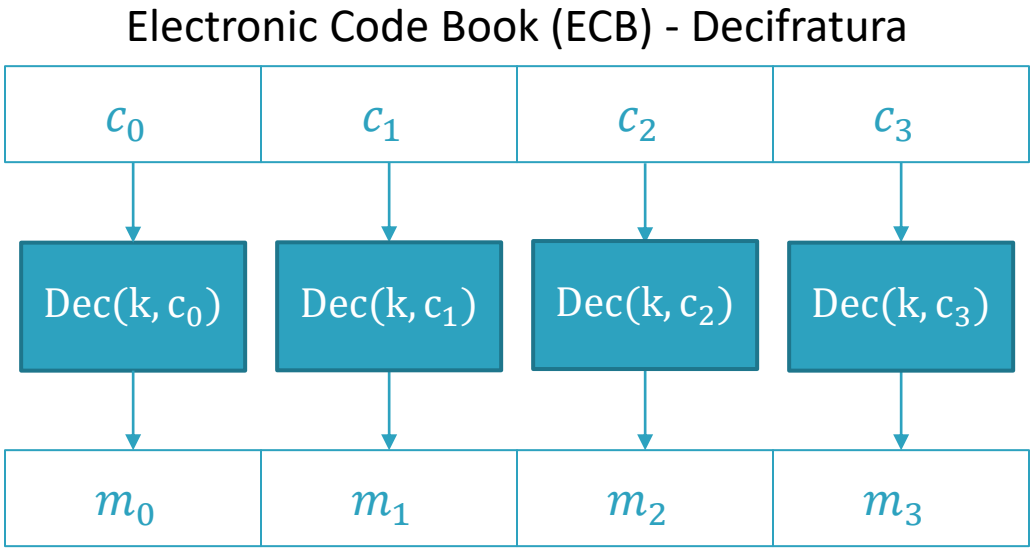
Electronic Code Book (ECB)

- Idea semplice:
  - Dividiamo in blocchi e cifriamo ogni blocco singolarmente
  - Questa modalità si chiama *"Electronic Code Book"* (ECB)
  - È la stessa idea del cifrario di Vigenère

Electronic Code Book (ECB)

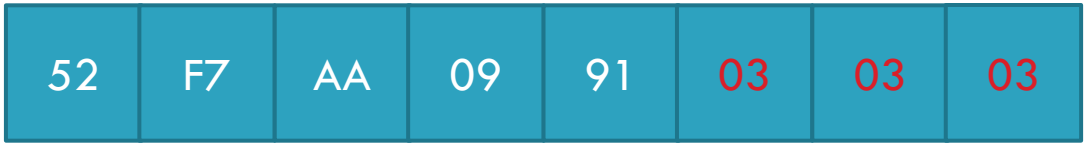


Electronic Code Book (ECB)



Padding

- Cosa succede se il numero di bit del messaggio non è multiplo della grandezza del blocco?
  - Gli standard *PKCS#5* e *PKCS#7* introducono un meccanismo di *padding*
  - Si "completa" il blocco con dei byte con valore il numero di byte mancanti



  - Nota: se la lunghezza è giusta si aggiunge comunque un intero blocco!

Problemi

- In ECB due blocchi uguali vengono cifrati allo stesso modo
- La struttura del messaggio viene mantenuta

Problemi

Immagine in chiaro

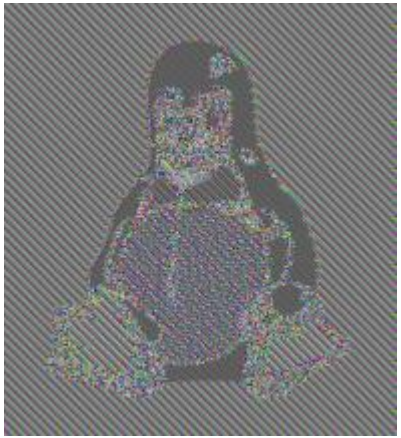


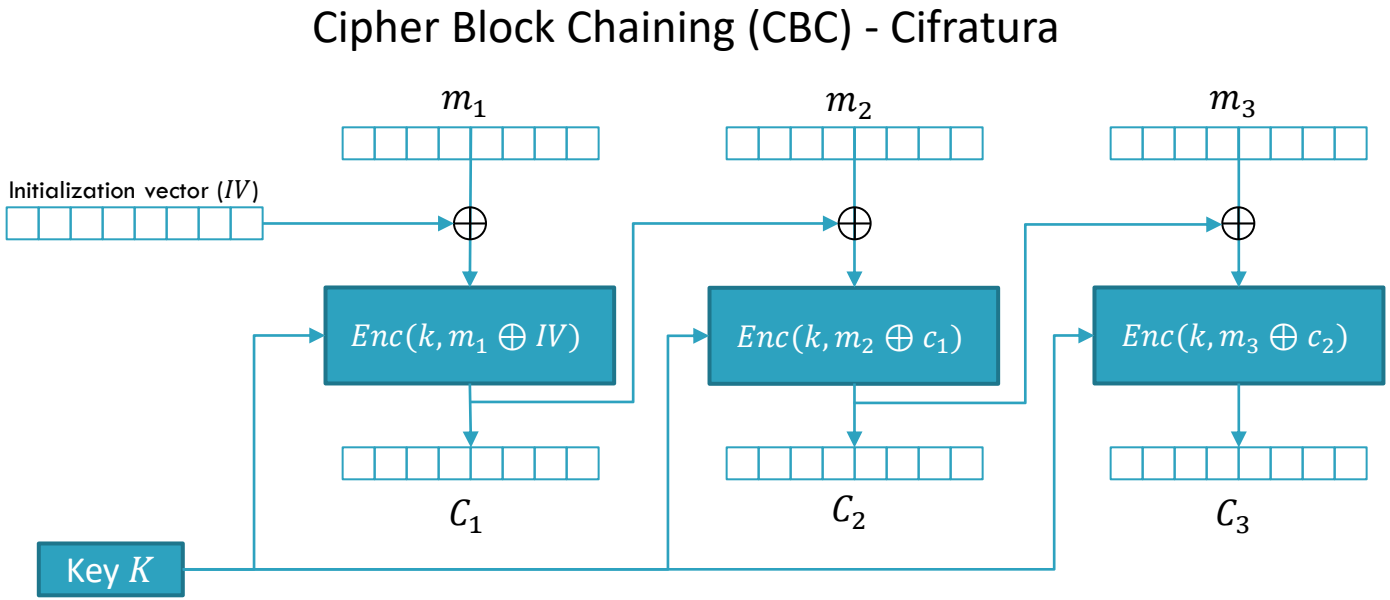
Immagine cifrata in ECB

Immagini da https://commons.wikimedia.org/

Cipher Block Chaining (CBC)

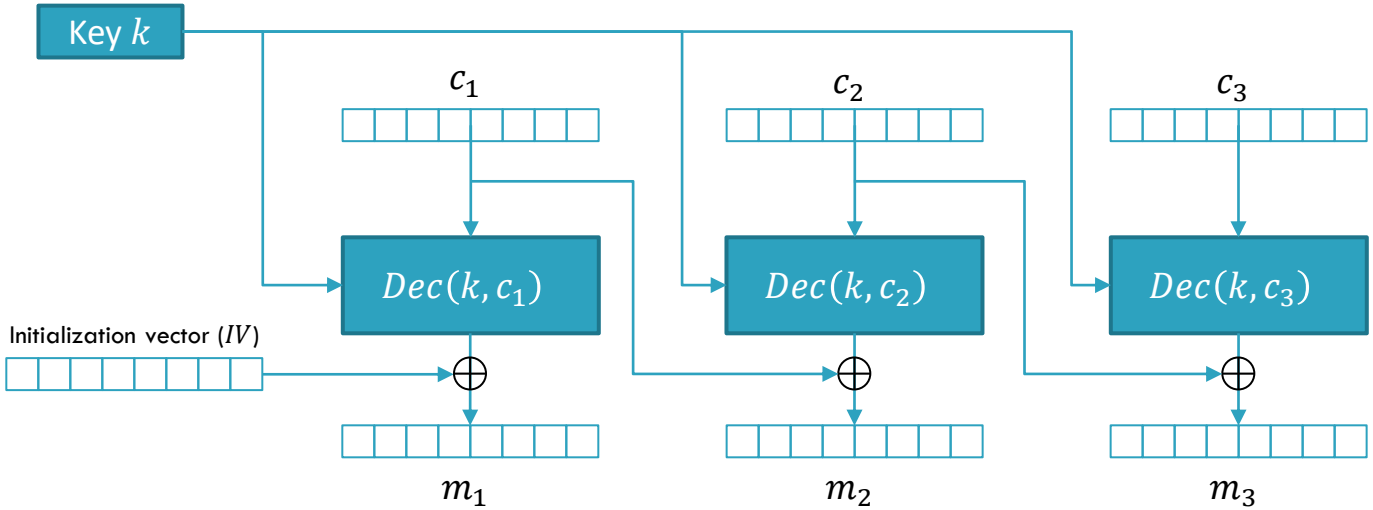
- Idea: usare il blocco cifrato precedentemente per "distruggere" la struttura del messaggio
- Questa modalità si chiama *"Cipher Block Chaining"* (CBC)
- Utilizziamo un numero random aggiuntivo (*Initialization Vector*, IV) per introdurre casualità nel primo blocco

Cipher Block Chaining (CBC)



Cipher Block Chaining (CBC)

Cipher Block Chaining (CBC) - Decifratura



Attacchi a CBC

- Gli attacchi a CBC sono spesso dovuti a errori di implementazione:
  - Scelte non random dell'IV (es. attacco BEAST a TLS)
  - Padding oracles

Counter Mode (CTR)

- Idea: usare il block cipher come PRNG per un OTP
- Questa modalità si chiama *"Counter Mode"* (CTR)
- Utilizziamo un numero random aggiuntivo chiamato *nonce* (number used once) per garantire che lo stream sia unico
- Trasformiamo di fatto un block cipher in uno stream cipher

Counter Mode (CTR)

- In pratica:
  - Scegliamo (casualmente) il nonce (es. 12345678)
  - Inizializziamo un contatore a 0
  - Generiamo lo stream cifrando la concatenazione tra nonce e contatore
    - $Enc(k, 1234567800000000)$
    - $Enc(k, 1234567800000001)$
    - $Enc(k, 1234567800000002)$
    - etc.

Modalità di funzionamento

- Esistono altre modalità di funzionamento:
  - Cipher FeedBack (CFB)
  - Output FeedBack (OFB)
  - Galois Counter Mode (GCM)
  - etc.

What's next?

- Nella prossima lezione:
  - Integrità, Autenticazione e funzioni di Hash
  - Metodi per scambiare le chiavi in maniera sicura

Cryptography 1

Matteo ROSSI
Politecnico di Torino

https://cybersecnatlab.it