

La crittografia nel contesto della sicurezza

Cybersecurity

<date>
Location



Università
degli Studi
della Campania
Luigi Vanvitelli

Cosa è la Crittografia

Crittografia: “scrittura cifrata o convenzionale che può essere compresa solo da chi ne conosce la chiave” (Dizionario Garzanti)

- L'idea alla base dell'utilizzo della crittografia nel contesto della sicurezza è quello di rendere inaccessibile il dato a terzi

↳ Per stabilire se qualcuno è quello che dichiara di essere

Termini:

- **Crittografia (Cryptography)**: Lo studio dei principi e dei metodi per criptare dati
- **Crittoanalisi (Cryptanalysis o Codebreaking)** – Lo studio dei principi e dei metodi per decifrare del testo cifrato senza conoscere la/le chiave/i
- **Crittologia (Cryptology)** – Il campo che include sia Crittografia che Crittoanalisi

Soluz. crittografica perfetta è una p[er] cui non esiste tecniche di crittoanalisi per follarla. Stessa cosa per il cifrario
Entrambi i lavoratori devono conoscere entrambi

Terminologia Introduttiva

- **Plaintext:**
 - the original text
- **Ciphertext:**
 - coded text
- **Cipher:**
 - algorithm for transforming plaintext to ciphertext
- **Key:** *info usata nella trasformaz.*
 - info used in cipher known only to sender/receiver
- **encipher (encrypt):** *Processo di trasformaz.*
 - converting plaintext to ciphertext
- **decipher (decrypt):**
 - recovering ciphertext from plaintext

Cifrario di Cesare

- Primo esempio di utilizzo della crittografia, risalente a Giulio Cesare (!)
- Approccio estremamente semplice: sostituire ogni lettera con quella 3 posizioni più avanti rispetto all'ordine alfabetico
- Esempio:

meet me after the toga party

PHHW PH DIWHU WKH WRJD SDUWB



Non difficile decifrare l'originale: ci sono 26 chiavi possibili; un attacco a forza
bruta è semplicissimo da implementare

Caesar Cipher

- La trasformazione può essere sintetizzata come:

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

- Assegnando ad ogni carattere un valore numerico

a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

- La cifratura di Cesare è:

$$c = E(p) = (p + k) \bmod 26$$

cifra il plain text + un valore K

$$p = D(c) = (c - k) \bmod 26$$

Cifratura di tipo shift

Attacco a Forza Bruta

Dimensione Chiave	Numero Alternative	Tempo ^{, 10⁶ tentativi} (10 ⁶ decryption/μs)
32 chiave di 32 bit	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	6.4×10^6 years

Mi sistema è sempre utilizzabile a forza bruta, ma computazionalmente non ha senso in alcuna cas. Aspetto oggettivo vs soggettivo.

Monoalphabetic Cipher

- Punto debole del Cifrario di Cesare
 - la Chiave (k) ha solo 26 alternative
- **Approccio Monoalfabetico:**
 - Scambiare le lettere in modo arbitrario
 - La chiave è composta da 26 lettere, ognuna con 26 valori
- Chiave:

Plain: abcdefghijklmnopqrstuvwxyz
Cipher: DVQKFIBJWPESCXHTMYAUOLRGZN *generated randomly*
- Esempio di Trasformazione:

Plaintext: if we wish to replace letters
Ciphertext: WIRFRWAJUHYFTSDVFSUUUFYA

Attacco a forza bruta: 26! possibili chiavi

Punto debole: non tutte le combinazioni obbligatorie sono valide!

Attacco a Forza Bruta

Dimensione Chiave	Numero Alternative	Tempo (10^6 decryption/ μ s)
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	6.4×10^6 years



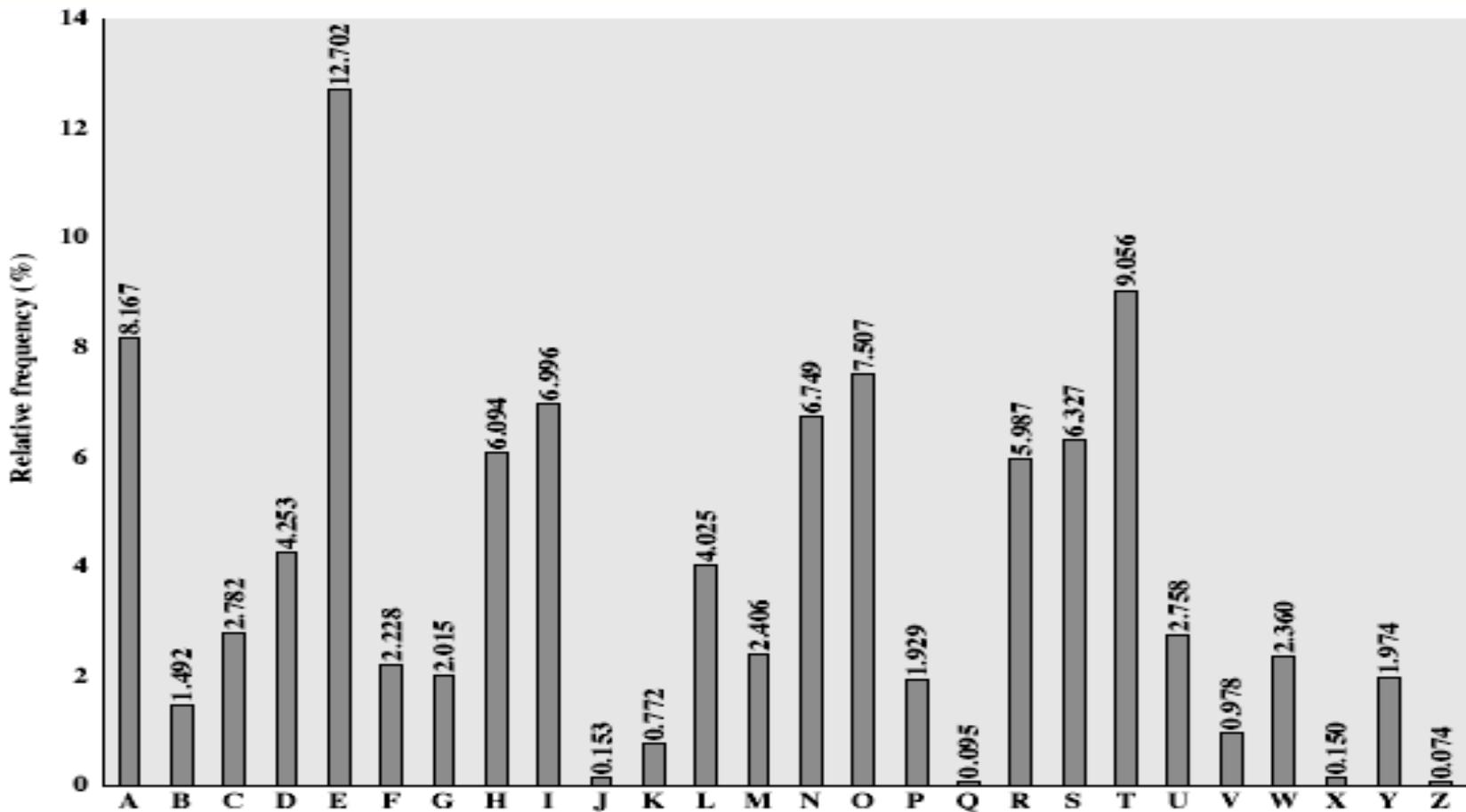
Il Cifrario Monoalfabetico è sicuro?

Esempio di Crittoanalisi

- Il Linguaggio umano è ridondante:
 - Non tutte le lettere vengono usate con la stessa frequenza
 - Alcune sequenze di lettere sono più comuni di altre
- Ad esempio, in inglese:
 - la lettera E è la più utilizzata, seguita da T,R,N,I,O,A,S
 - Le lettere Z,J,K,Q,X sono rare

Probabilmente la più frequente è la lettera e.

Frequenza di utilizzo delle lettere (inglese)



Prendo libro wfront e faccio la stessa confrontazione. Funziona al limite, per la legge dei grandi numeri

Analisi in frequenza

Raccogli (molti) testi cifrati

Conta frequenza di ciascun carattere
nel testo

Sostituisci il carattere con quello la
cui frequenza corrisponde alla
frequenza nella lingua utilizzata

Questo se ho accesso al messaggio

Attacchi basati su crittoanalisi

- **Basati solo su ciphertext**
 - Si conosce solo l'algoritmo e/o il testo cifrato
- **Plaintext noto**: attacco monoalfabetico : se manda ABCD... ho la chiave
 - know/suspect plaintext & ciphertext
- **chosen plaintext**
 - select plaintext and obtain ciphertext
- **chosen ciphertext**
 - select ciphertext and obtain plaintext
- **chosen text**
 - select plaintext or ciphertext to en/decrypt

Cifrari Polialfabetici

- Aumentare la sicurezza del cifrario utilizzando più alfabeti. *Tanti cifrari monalfabetici per una lettura*
 - Attacchi brute force più complessi
 - Attacchi in crittoanalisi più complicati perché si “appiattisce” la distribuzione in frequenza
- Si usa una chiave per selezionare l’alfabeto da utilizzare, e si cambia l’alfabeto ad ogni nuova esecuzione.

Cifrario di Vigenère

- Il più semplice dei cifrari polialfabetici
- Si basa sul cifrario di cesare e la chiave è composta da n chiavi con un valore 1..26
 - $K = k_1 \ k_2 \ \dots \ k_d$
- ith lettera specifica l'ith alfabeto da usare
- Per ogni lettera si applica il cifrario di cesare e poi si fa scorrere la chiave in modo circolare.

Esempio

- Password: deceptive
- key: deceptivedeceptivedeceptive
- plaintext:
 - wearediscoveredsaveyourself
- ciphertext:
 - ZICVTWQNGRZGVTWAVZHCQYGLMGJ

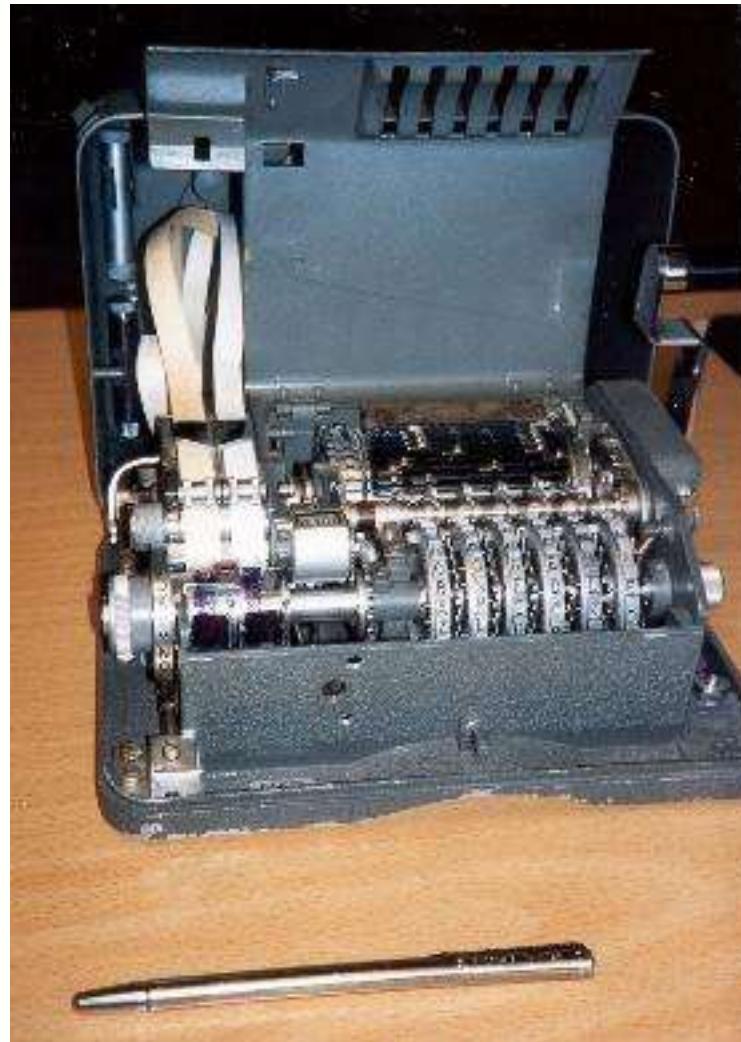
Sicurezza per il cifrario di Vigenere

- Esistono molteplici cyphertext per ogni plaintext
(a seconda dell'utilizzo della chiave)
- La frequenza di utilizzo delle lettere viene quindi nascosta, ma non completamente persa.
- Approccio di criptoanalisi:
 - Prova a decifrare usando approccio monoalfabetico.
 - Se non va, assumi polialfabetico con dimensione 2
 - Incrementa la dimensione ad ogni fallimento...

Rotor Machines

- Prima delle moderne tecniche di cifrature, le rotor machine erano gli strumenti di cifratura più potenti, ampiamente utilizzate durante la seconda guerra mondiale
 - German Enigma, Allied Hagelin, Japanese Purple
- Implementano un complesso e variabile sistema di cifratura a sostituzione
- Utilizzano una serie di cilindri, ognuno in grado di effettuare una sostituzione, poi ruotato per il carattere successivo: con 3 cilindri $26^3 = 17576$ alphabets

Hagelin Rotor Machine



One-Time Pad

- Per ogni messaggio:
 - Genera una chiave che non verrà mai più riutilizzata
 - La chiave è lunga quanto il messaggio
 - Cifra con il cifrario di cesare ogni lettera con la corrispondente della chiave.
- Meccanismo di cifratura perfetto
 - E' assolutamente impossibile da rompere, poichè non esiste alcuna associazione tra ciphertext e plaintext

Se ho solo ciphertext, per qualsiasi plaintext Esiste una chiave che mi ritorna quel plaintext.

Come passo la chiave? Punto debole. Viene generata? Punto debole.

Punto debole: la Confidenzialità, Sono sicuro al 100% che non sono un grande di rottura del plaintext.
NOTA: non riusco a nascondere che il messaggio esiste. Il cuore del problema è garantire confidenzialità sul traffico.

Crittografia non è sufficiente alla garantire confidenzialità sul traffico.

One-Time Pad

- Per ogni messaggio:
 - Genera una chiave che non verrà mai più riutilizzata
 - La chiave è lunga quanto il messaggio
 - Cifra con il cifrario di cesare ogni lettera con la corrispondente della chiave.
- Meccanismo di cifratura perfetto
 - E' assolutamente impossibile da rompere, poichè non esiste alcuna associazione tra ciphertext e plaintext
- Problema: Come distribuire le chiavi?

Crittologia e Sicurezza

- In sicurezza la crittologia è uno **STRUMENTO**
- Necessario spesso Conoscere **COSA FA** non tutti i dettagli del **COME FA**
 - Lo usiamo nei protocolli e nei sw
- Utilizzo di **protocolli** e **software** che ne sfruttino adeguatamente le **proprietà**
- L'implementazione può nascondere altrettante vulnerabilità degli algoritmi.
 - Un algoritmo sicuro **NON** garantisce un software sicuro.

Crittologia e Sicurezza

- La Crittologia è un campo della matematica estremamente complesso
 - Valutare e usare strumenti basati sulla crittologia NON sviluppare nuovi algoritmi
 - Cosa è necessario conoscere:
 - I Diversi tipi di Algoritmi Crittografici:
 - Crittografia Simmetrica Con una chiave possono criptare e decrittare
 - Crittografia Asimmetrica Esistono 2 o più chiavi. Una per criptare, una per decrittare
 - Funzioni HASH*
 - Più garanzie su identità ecc.
 - Loro proprietà e caratteristiche
- * A 1 via sola: produce stringhe delle quali non sono mai in grado di ricostruire plain-text. Senza più integrità e modifiché dati

BREVE PANORAMA DEGLI ALGORITMI CRITTOGRAFICI

Crittografia a chiave Simmetrica

- Crittografia a blocchi:
 - L'algoritmo opera su blocchi di dimensione fissa del plaintext,
 - Il Plaintext va suddiviso in blocchi di dimensione fissata
 - Se l'ultimo blocco non è della dimensione corretta, va utilizzato del Padding
- Crittografia a stream:
 - il plaintext è un flusso continuo di dati;
 - la chiave viene utilizzata per generare un flusso continuo di bit (KeyStream) generazione di flusso di bit casuale che divide la chiave
 - Il cyphertext è ottenuto come plaintext XOR keystream

↑
Se faccio XOR con 2 stringhe e riapplico, ottengo la prima.

Algoritmi Simmetrici

- Crittografia Simmetrica a blocchi
 - Data Encryption Standard (DES):
 - è il primo algoritmo di cifratura scelto come standard dal Federal Information Processing Standard (FIPS) per il governo degli Stati Uniti d'America (1976).
 - Chiave a 56 bit, Blocchi a 64 bit. \Rightarrow 2 caratteristiche importanti. Dunq. chiave è importante per la funzionalità.
 - Rotto pubblicamente per la prima volta nel 1999.
 - Triple DES: l'uso di algoritmi diversi
 - Esecuzione del DES tre volte innestata (più schemi possibili) \Rightarrow Risultato dato con altra chiave non generabile.
 - Chiave compresa tra 112 e 168 bit a seconda dello schema. Blocchi a 64 bit incremento di sicurezza. Su sollecitazione che se ci si può con una chiave e' necessario un nuovo passo da complessità 2^{56} a 2^{57} .
 - Advanced Encryption Standard (AES):
 - AES è stato adottato dalla National Institute of Standards and Technology (NIST) è dalla US FIPS PUB nel novembre del 2001 dopo 5 anni di studi, standardizzazioni e selezione finale tra i vari algoritmi proposti.
 - Chiave a 128, 192 o 256 bit, Blocchi di 128 bit
- Crittografia Simmetrica a Stream:
 - RC4:
 - sviluppato da Ron Rivest della RSA Security nel 1987. Alla base di molti protocolli per la sicurezza in reti wireless (es. WEP e WPA1)
 - Chiave 40 a 256 bit
 - Generalmente considerato molto debole



- * Prima critt con una chiave, poi decritt con una chiave diversa, poi critt con nuova chiave.
Complessità che ve ritrovano ad 2^{100} .
- Δ Problema del DES era crescita di potenza di calcolo. AES progettato per funzionare con 3 chiavi di dimensione diversa.
Ma quant'è meno sicura quella da 128 bit? La vera domanda è quanto più siamo con lo crescere.

RICORDA: "nessuno dei questi algoritmi è imbocchabile" - HYPE

Con $t \rightarrow \infty$, $O(n) \rightarrow \infty$ si rompono tutti i framme del CEO degli algoritmi.

Crittografia Simmetrica a blocchi: Modes of operation

Dati più grandi del blocco, che faccio?

- **Problema:** Dimensione dei Dati Maggiore del singolo blocco
 - NIST SP 800-38A definisce 5 “Modes of Operation”
 1. Electronic Codebook Book (ECB)
 2. Cipher Block Chaining (CBC)
 3. Cipher FeedBack (CFB)
 4. Output FeedBack (OFB)
 5. Counter (CTR)
-
- The diagram consists of two curly braces. The left brace groups the four modes from 1 to 4 (ECB, CBC, CFB, OFB) and is labeled "BLOCCHI". A handwritten note above it says "Divide i dati in blocchi e critta interando". The right brace groups mode 5 (CTR) and is labeled "STREAM".

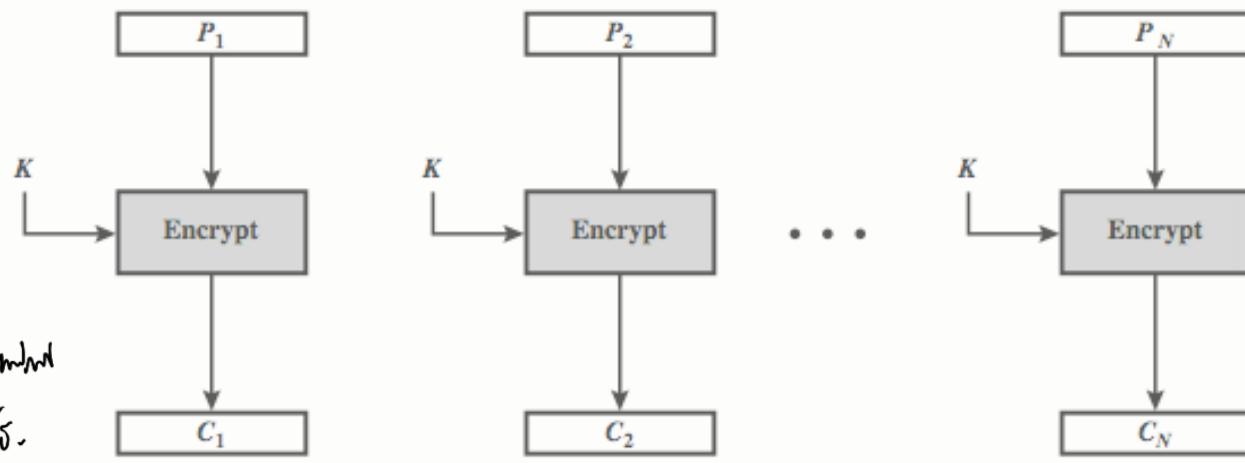
Electronic Codebook Book (ECB)

Non garantisce intesa
di confidenzialità e integrità.

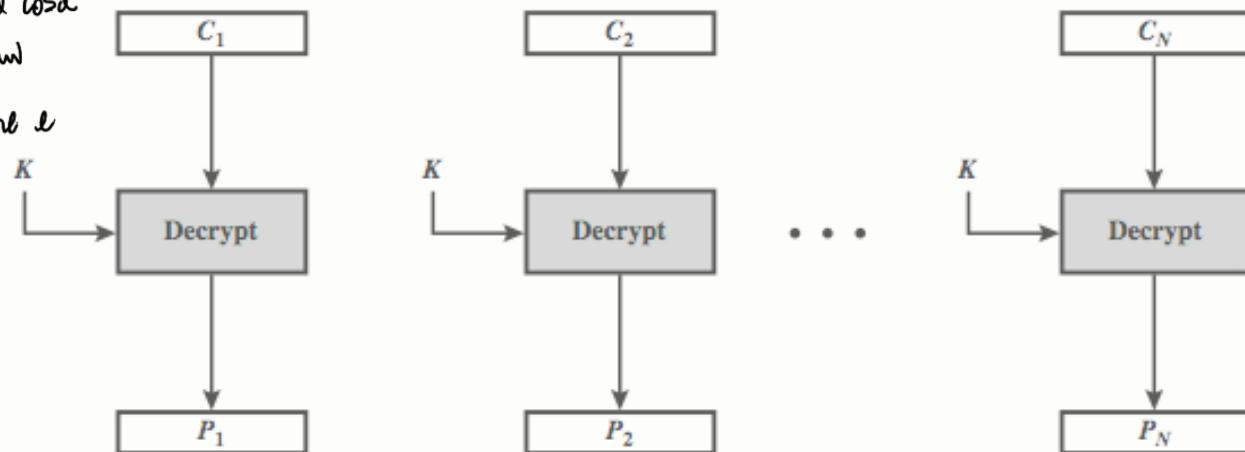
Se le dimensioni sono troppo grandi:

- in chiavi Sono un po' deboli visto gli algoritmi

↳ dal ciphertext, se vittima qualche cosa
il plaintext diventa illegibile. Quindi
ci sono alterazioni che possono fare e
far diventare tutto illegibile?



(a) Encryption



(b) Decryption

- Immagine P₁. Info transazione bancaria. P₁ contiene 0010, 10 euro. Prima ho gli ultimi 4 caratteri del codice bancario 1027.
Se prendo P₁ e lo metto su P₂, ho pagato 1027 euro senza errori e correttivi. Questo plaintext è stato trasdotto con la stessa chiave. È come se fosse monocod. con lettere ohe alfabeto di 64 bit.
- Considerando: immagine un'immagine raw RGB, dove per le R, G, B uso 8 bit. 24 bit per pixel RGB. Immagine un cerchio. Ballo è tutto 0. Nero è tutto 1. Dopo aver invertito cambia tutto ma la verità vale e questo non pratica. Tutto nel caso monocodifico.

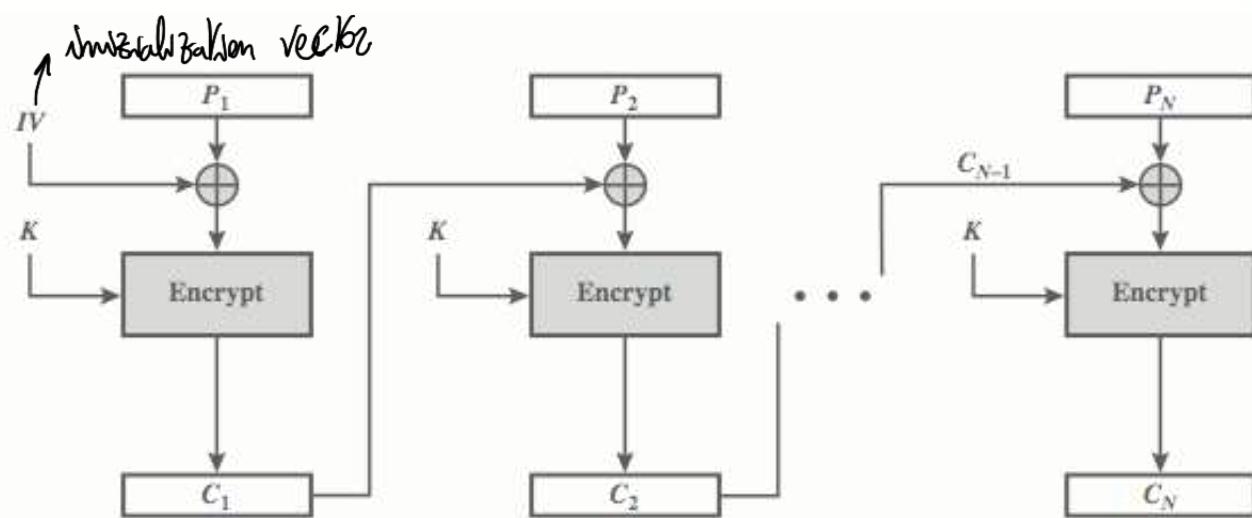
Cipher Block Chaining (CBC)

$$C_i = E_K(P_i \text{ XOR } C_{i-1})$$

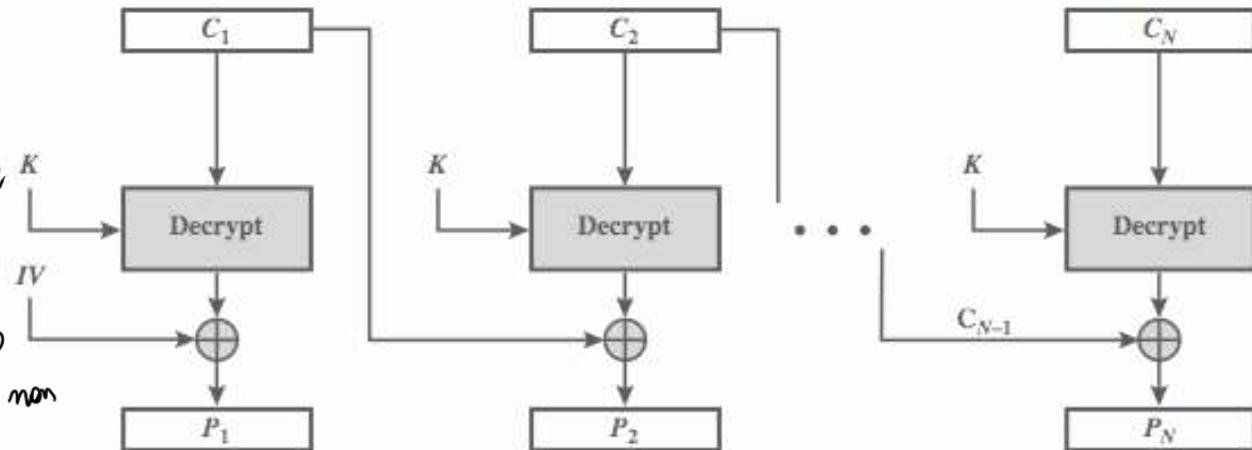
$$C_{-1} = IV$$

Protegge molto la confidenzialità
ma ha qualche problema
di integrità.

NOTA: Se genero messaggi di lunghezza
imposta chi decodifica il messaggio. Entrare non
può essere specificissimo.



fo uso per mettendo un XOR
Quindi, se do 2 volte lo
stesso messaggio e mando un IV diverso ho messaggio finale diverso.



s-bit Cipher FeedBack (CFB-s)

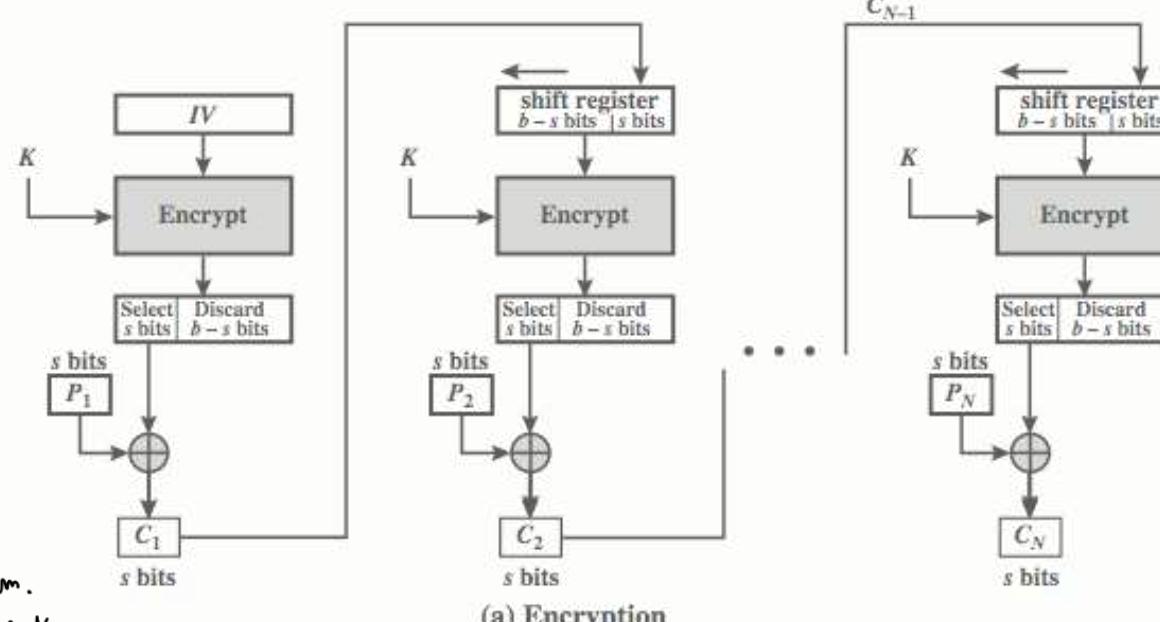
$$C_i = P_i \text{ XOR } E_K(C_{i-1})$$

$$C_{-1} = IV$$

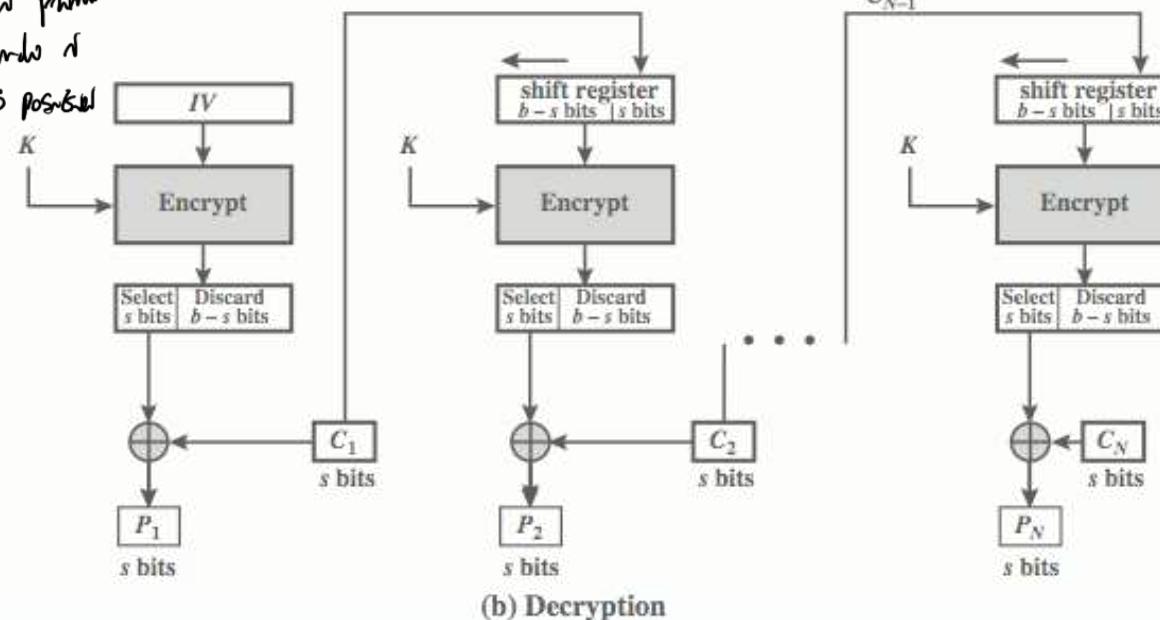
Funziona allo stesso modo di un algoritmo a stream.

Ho sempre IV. Chiavi sempre uguali. IV scambiato all'inizio tra le parti. Cifro IV e prendo i primi s bit per lo stream e metto un XOR. Perdono i bit del blocco C_1 e shiftlo a sinistra di s posizioni.

Procedo così all'infinito.



(a) Encryption

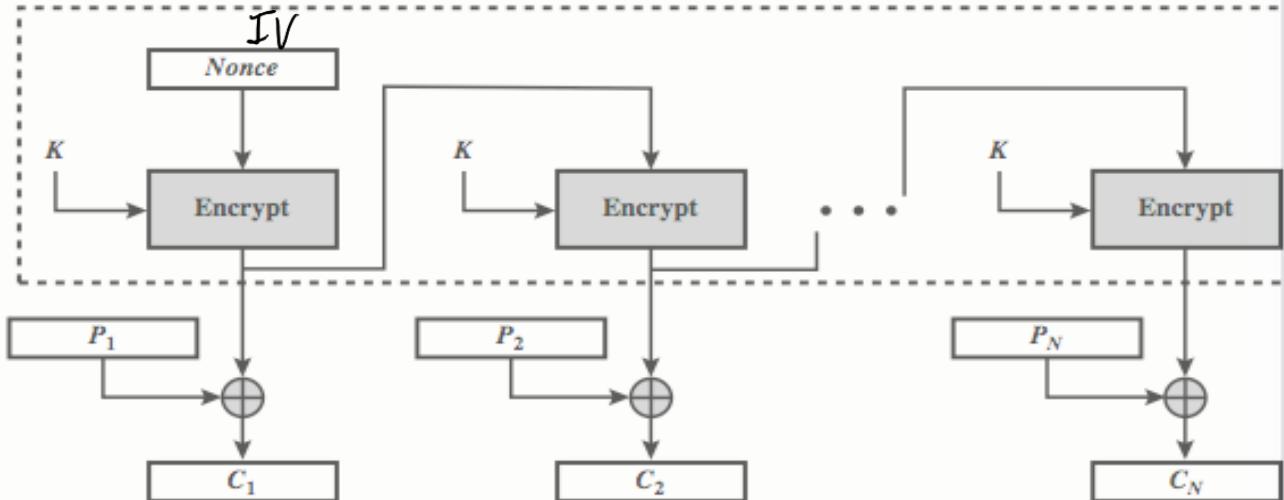


(b) Decryption

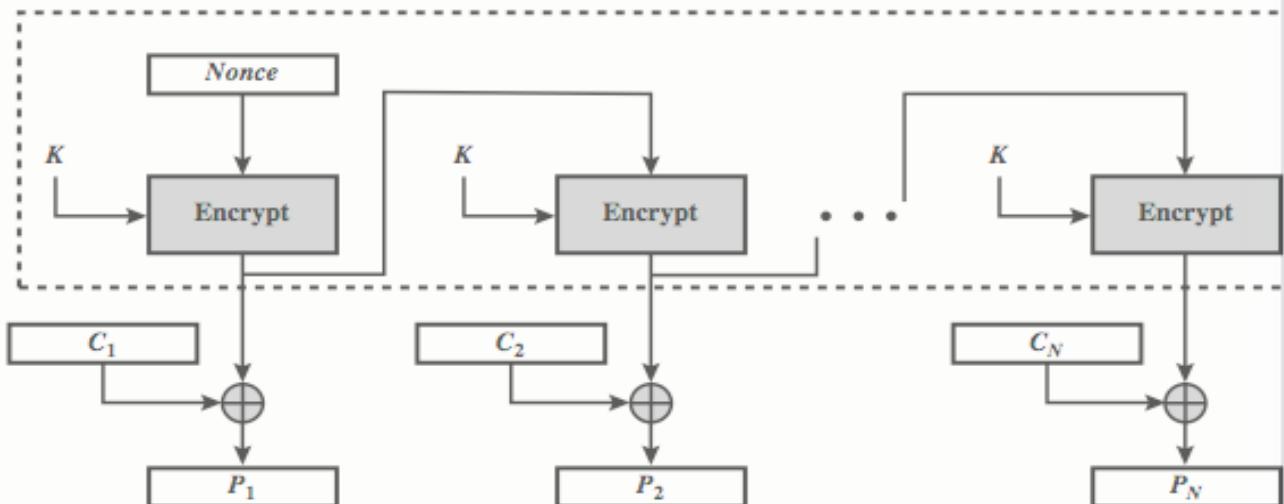
Output FeedBack (OFB)

$$\begin{aligned}O_i &= E_K(O_{i-1}) \\C_i &= P_i \text{ XOR } O_i \\O_1 &= IV\end{aligned}$$

Dó làm input output dell'encrytion. IV



(a) Encryption

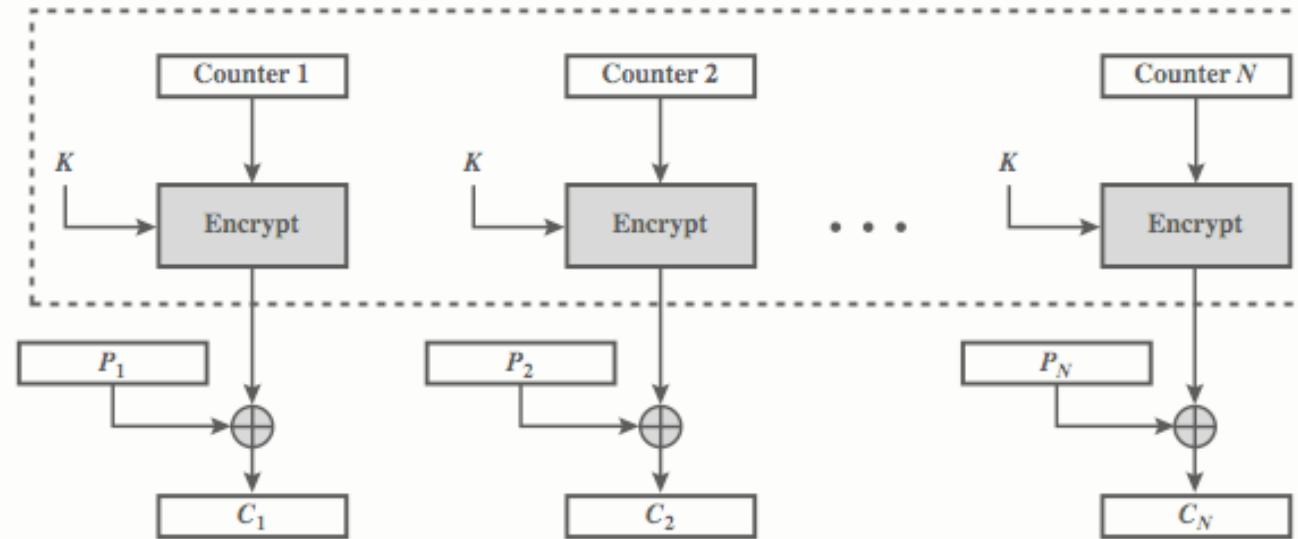


(b) Decryption

Counter (CTR)

$$O_i = E_K(i)$$

$$C_i = P_i \text{ XOR } O_i$$

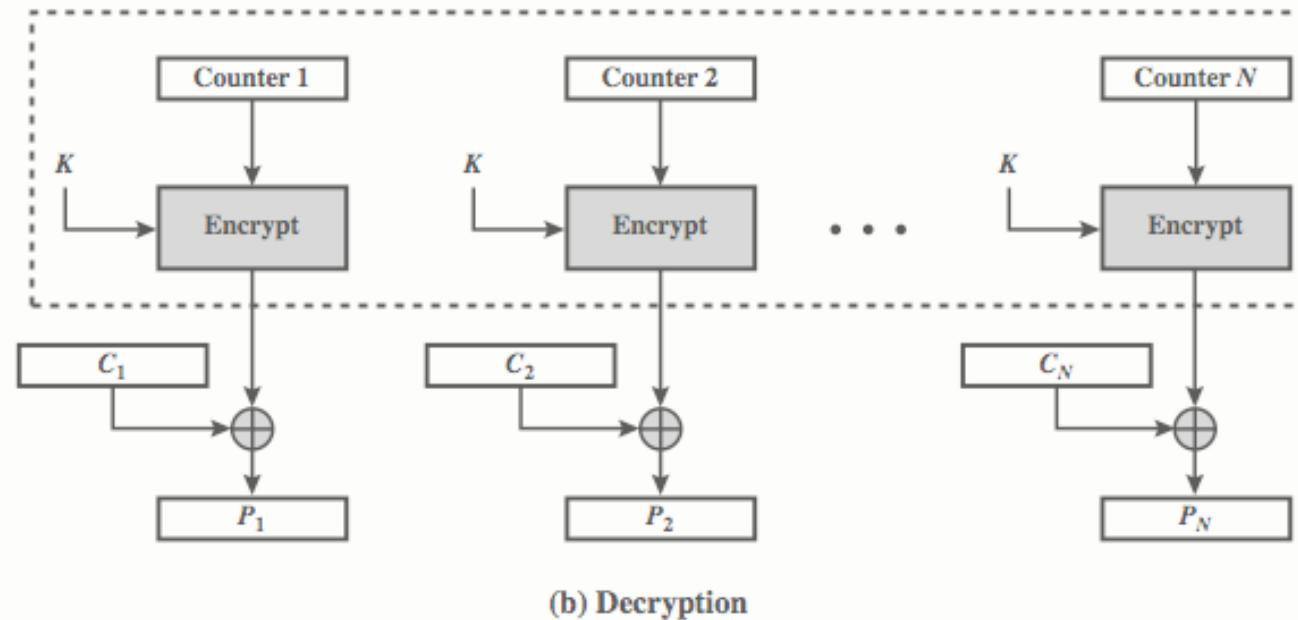


Usa direttamente un contatore per input a un blocco. Encryption è efficiente. (a) Encryption

Basta non sapere da dove partono.

Quanti KeyStream posso avere?

2^N (ottenzione del flusso)



(b) Decryption

Crittografia a chiave Asimmetrica

- **Tecnica a chiave Asimmetrica:**
 - Due chiavi differenti per codificare e decodificare
 - $\text{ciphertext} = E(K_1, \text{plaintext})$, $\text{plaintext} = D(K_2, \text{ciphertext})$
 - $\text{ciphertext} = D(K_2, \text{plaintext})$, $\text{plaintext} = E(K_1, \text{ciphertext})$

Utilizzo della chiave Asimmetrica

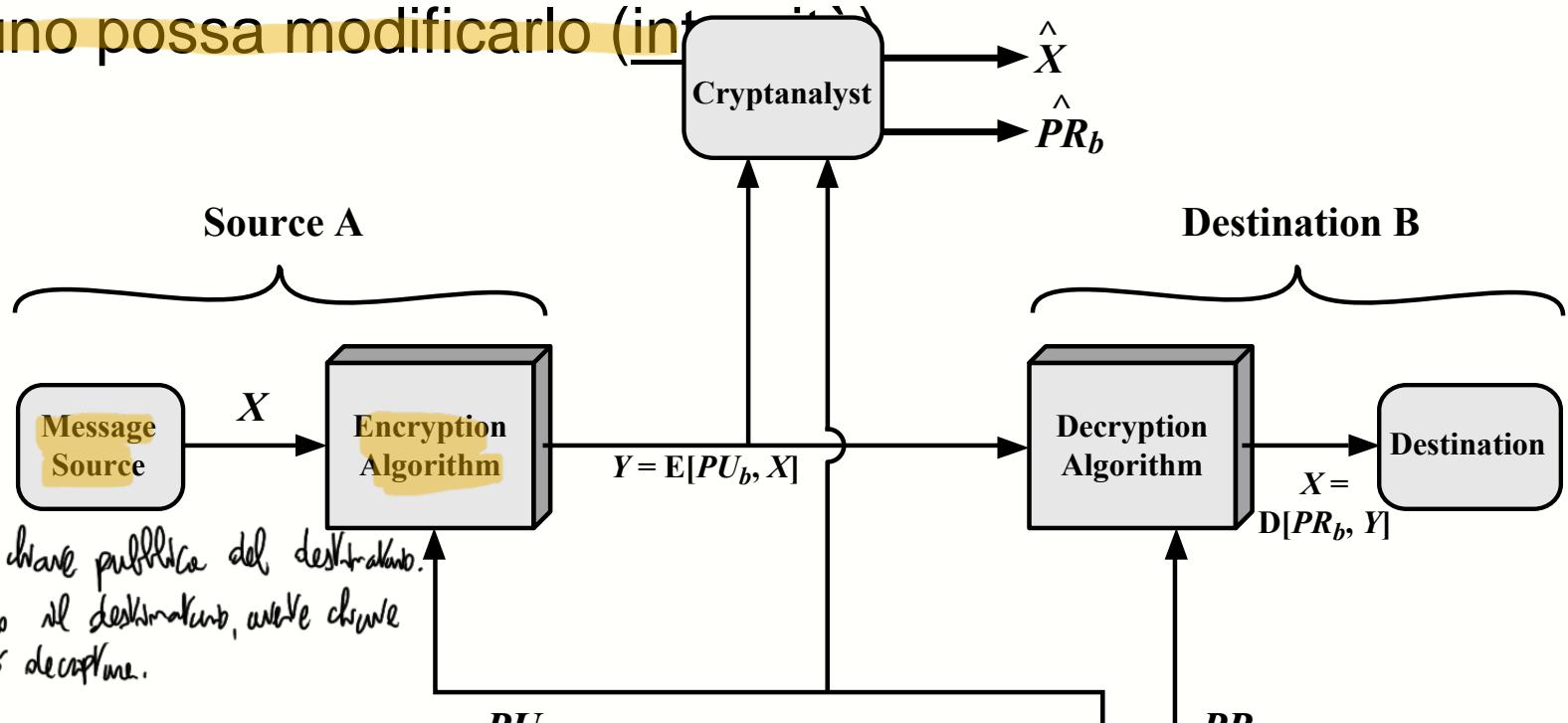
Risolve problema di distribuzione.

Qua ogni utente ha 2 chiavi, Chiave Pubblica e Chiave Privata.

- Il Principio di utilizzo è abbastanza semplice:
- Ogni utente (Alice e Bob) acquisisce una coppia di chiavi
 - Una la mantiene per sè (chiave privata),
 - l'altra la mette a disposizione (chiave pubblica)
- Chiunque può prendere la chiave pubblica di un utente ed utilizzarla sia per crittare che per decrittare, ma se critta non può decrittare e viceversa
- Per questo motivo la tecnica è detta anche “ a chiave pubblica”

Spedizione di Dati Confidenziale

Bob vuole mandare un messaggio (m) ad Alice e vuole essere sicuro che solo lei possa leggerlo (confidenzialità) e nessuno possa modificarlo (integrità)



Grazie con chiave pubblica del destinatario.
Quindi solo il destinatario, avendo chiave privata, può decifrare.

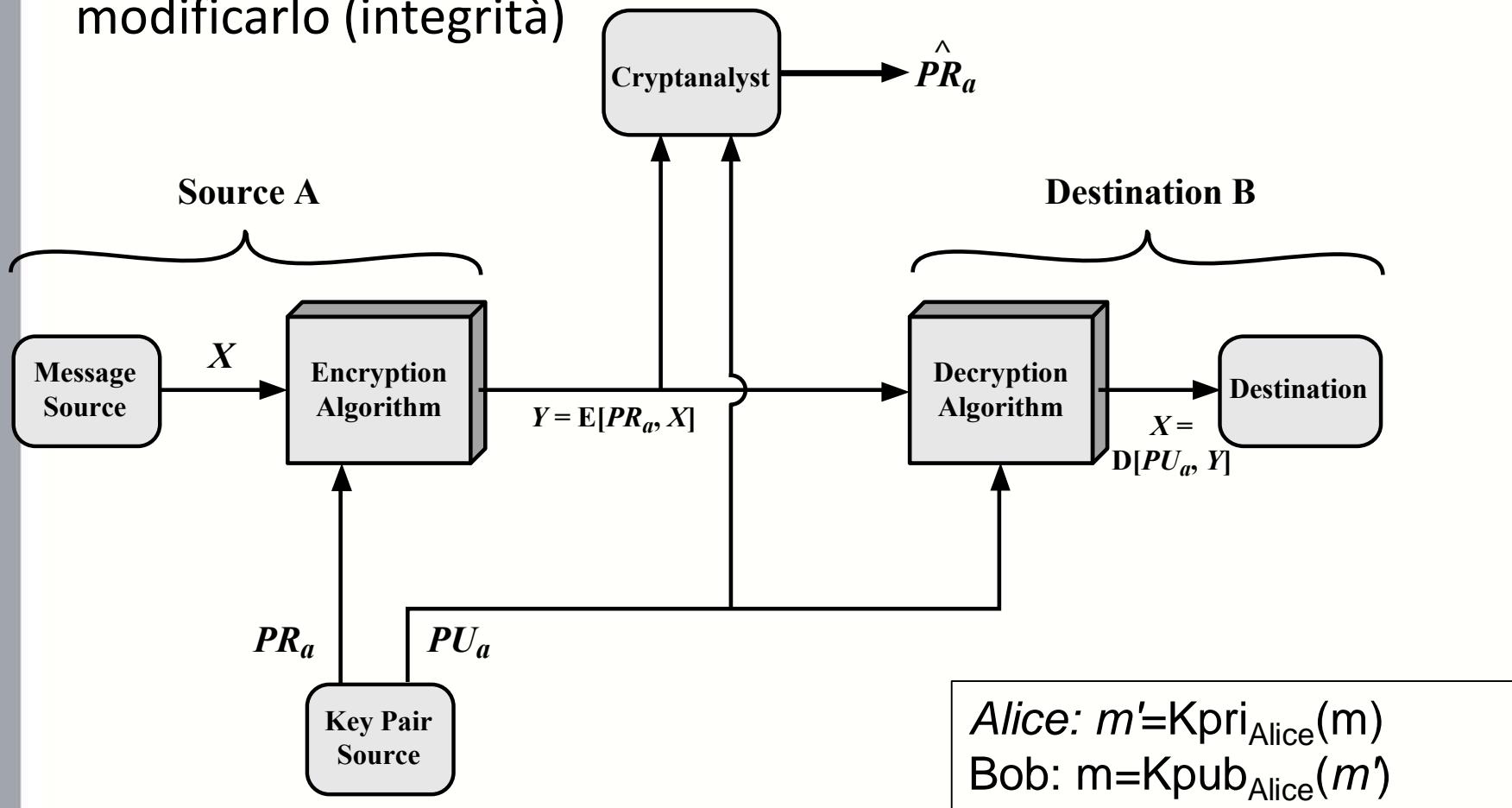
Bob: $m' = K_{pub, Alice}(m)$
Alice: $m = K_{pri, Alice}(m')$

Malfattori: se altero messaggio lo hanno tutti alterato. Ricorda che anche lui lavorando a blocchi è soggetto agli attacchi di padding.

- Se i segni generativi che messaggia è mandato dal verso, gli faranno avere le chiavi giuste per capire.
Tutti vedono messaggio, ma solo che vede da lontano.

Ricezione di Dati Sicura

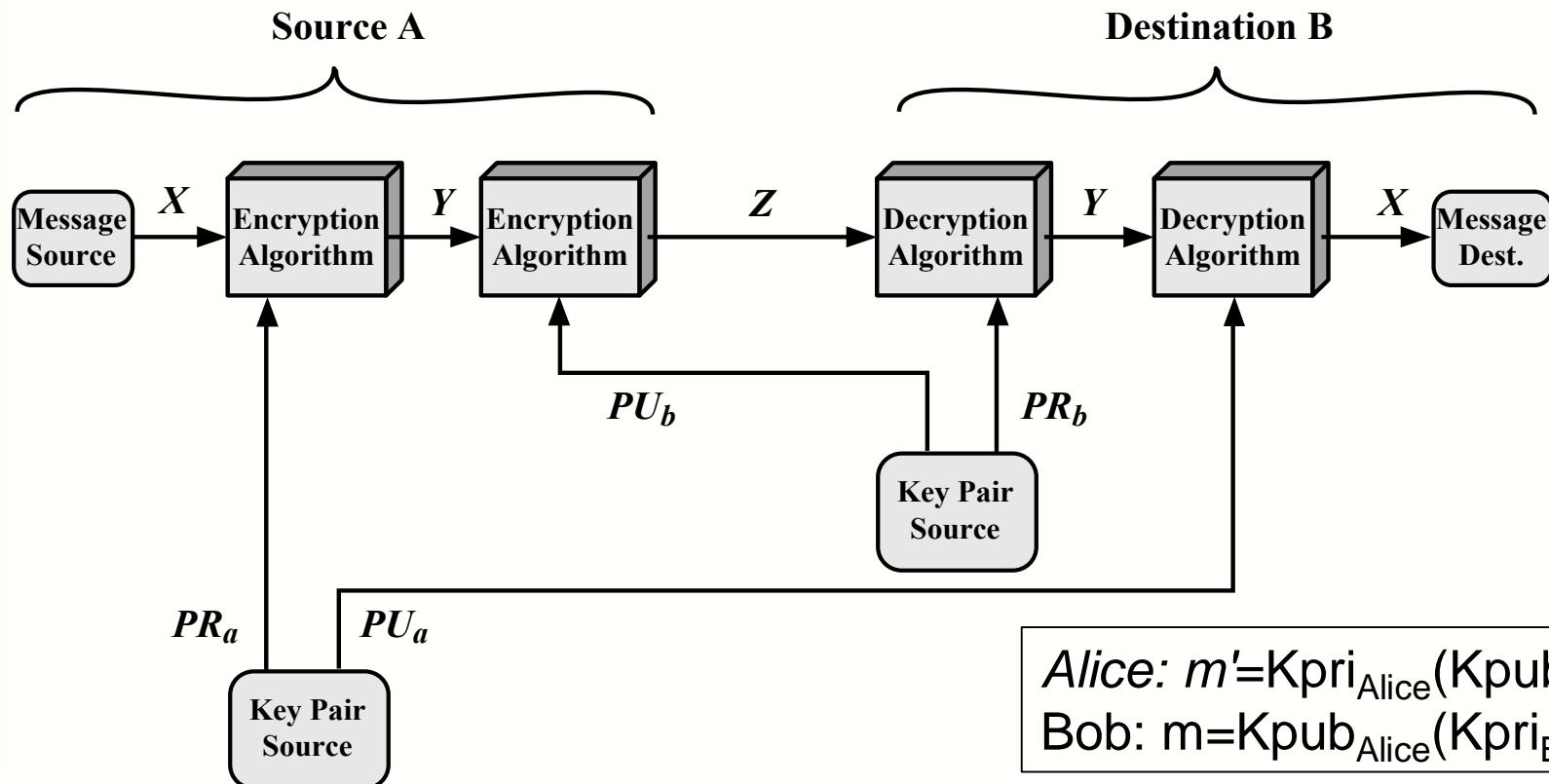
Alice vuole mandare un messaggio (m) a Bob e vuole che lui sia sicura che è stata lei a spedirlo (autenticazione) e nessuno possa modificarlo (integrità)



Approccio 2: prendo messaggio, critto con chiave privata del mittente, critto con chiave pubblica del destinatario. Per decrittare, uso prima la chiave privata del destinatario e poi quella pubblica del mittente. È un cerchio, lo uso per passare key di algoritmo simmetrico. Già fatto entrambe le cose.

Scambio sicuro

Alice vuole mandare un messaggio (m) a Bob e vuole essere sicura che solo lui possa leggerlo (confidenzialità), nessuno possa modificarlo (integrità) e che Bob sia sicuro che lo ha spedito lei (non ripudiabilità)



Crittografia Asimmetrica: Algoritmi

- **Diffie-Hellmann:**
 - è il primo algoritmo che introduce il principio della chiave pubblica e privata
 - NON permette la crittazione di qualsiasi dato, ma solo la generazione di una chiave condivisa tra due party e nota solo a loro usata per criptare
 - Algoritmo di criptazione simmetrica, ma con chiavi private differenti
 - Soggetto a man-in-the-middle ~~while~~ solo per fare scambio di chiavi,
- **RSA**
 - Basato sulla fattorizzazione in numeri primi, viene generata la coppia di chiavi, ogni messaggio può essere crittato
 - Chiavi minimo da 1024 bit, oggi consigliato almeno 2048
- **ECC (Crittografia ellittica)**
 - Utilizza il concetto di funzione ellittica invece della fattorizzazione per garantire la non invertibilità
 - Ha le stesse proprietà e modalità di utilizzo di RSA, ma permette 'adozione di chiavi più piccole'
 - Chiave a 224 bit è ad oggi considerata sicura

Funzioni Hash Sicure

- **Tecnica di Hash:**
 - Dato un plaintext di lunghezza qualsiasi, produce un ciphertext di lunghezza fissata
 - $\text{hash} = H(\text{plaintext})$
 - La funzione di Hash è pubblica
 - Computazionalmente semplice
 - deve essere computazionalmente impraticabile:
 - Dato hash sia possibile ricavare un plaintext che lo produca (one-way property)
 - Due plaintext che producano lo stesso hash (collision free)

Funzione che applicabile a dovrà qualsiasi e lunghezza da uscita uguale.
Voglio sia impossibile che sia n. 1. Ma non si può!

Requisiti di una funzione hash

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h , it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$. <small>funzione generica su x</small>
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$. <small>↪ deve non esistere nessun altro paio (x, y)</small>
Pseudorandomness	Output of H meets standard tests for pseudorandomness

La funzione CRC funziona come hash ma non può essere manipolata.

} similitudine
che non nasce
permette di avere y ,
estendo memoria
caso per "valori"

Utilizzo di Hash

- Integrità dei messaggi
 - Dato un messaggio, generare $\text{hash} = H(\text{mess})$
 - Spendendo mess e hash (separatamente o assieme) è possibile verificare che il messaggio non sia stato alterato
- Considerazioni
 - Paradosso del compleanno
 - Lunghezza minima di 256 bit / 512 oggi

Se prendo f hash con l'immagine, poiché la complessità è $2^{\frac{m}{n}}$, se hash non dà retta sufficientemente grande, la probabilità di trovare collisioni è molto alta.

↳ L'hash va applicato sempre sulle coppie (x, y) e non singolo valore.

Steganografia

- Alternativa alla crittazione
- Nasconde non il contenuto del messaggio ma la sua stessa esistenza
- Una parte del messaggio viene utilizzato per contenerne un'altro leggibile utilizzando solo parte dei caratteri.
- Esempi
 - Inchiostro invisibile
 - Messaggio nelle prime lettere di ogni riga
 - Watermark nelle immagini
- Contro: Grande overhead per piccoli messaggi

Crittografia: strumento per la confidenzialità

Hash: strumento per l'integrità.