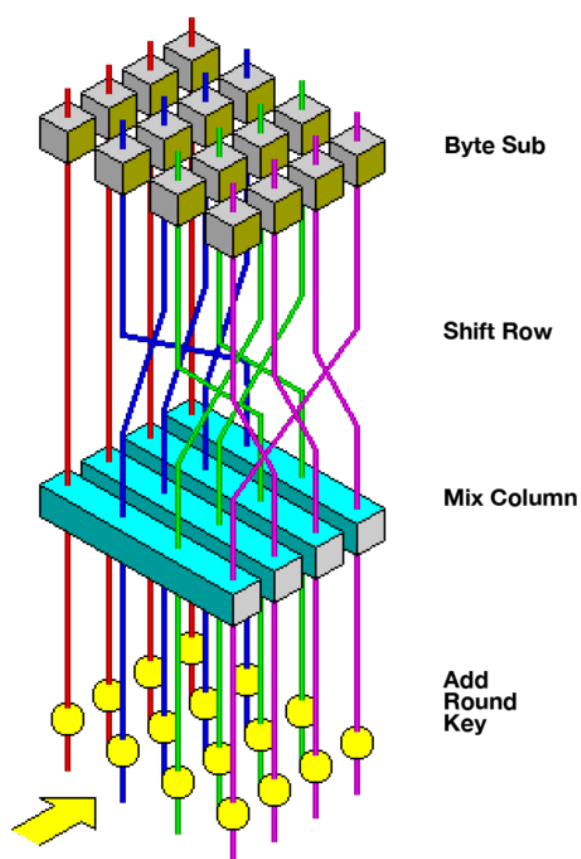


# Advanced Encryption Standard

## AES

Luca Rengo

v1.0.0  
MAGGIO | DICEMBRE  
2022





# Indice

<b>1</b>	<b>Storia di AES</b>	<b>1</b>
1.1	Introduzione . . . . .	1
1.2	Breve storia di AES . . . . .	1
1.3	AES vs Rijndael . . . . .	3
1.4	Cifratura simmetrica vs asimmetrica . . . . .	3
<b>2</b>	<b>L'Algoritmo</b>	<b>7</b>
2.1	Introduzione . . . . .	7
2.2	I tre concetti dietro la Crittografia . . . . .	7
2.3	Una panoramica sull'Algoritmo . . . . .	8
	Perché lo XOR è usato in crittografia? . . . . .	9
	Le modalità di AES . . . . .	12
<b>3</b>	<b>La Matematica dietro AES</b>	<b>13</b>
3.1	Introduzione . . . . .	13
3.2	Gruppi, Anelli e Campi . . . . .	14
	Gruppo . . . . .	14
	Monoide . . . . .	14
	Gruppo abelliano . . . . .	14
	Anello . . . . .	14
	Anello commutativo . . . . .	14
	Anello finito . . . . .	14
	Campo . . . . .	14
	Campi vs Anelli . . . . .	14
3.3	Il campo di Galois . . . . .	14
	Le operazioni del campo finito . . . . .	14
	Addizione e sottrazione . . . . .	15
	Moltiplicazione . . . . .	15

Esponenziazione . . . . .	15
Logaritmi . . . . .	15
Divisione . . . . .	15
Inverso . . . . .	15
3.4 Il teorema fondamentale della teoria di Galois . . . . .	15
<b>4 L'implementazione</b>	<b>17</b>
4.1 Introduzione . . . . .	17
<b>5 Le modalità di AES</b>	<b>19</b>
5.1 Introduzione . . . . .	19
5.2 A cosa servono le modalità? . . . . .	19
5.3 IV, Nonce, Salt e Pepper . . . . .	19
Nonce   Number Used Once . . . . .	20
Nonce sequenziali . . . . .	20
IV   Initialization Vector . . . . .	20
IV vs Nonce . . . . .	20
Salt . . . . .	20
Pepper . . . . .	20
5.4 Le modalità . . . . .	21
Modalità di cifratura senza integrità del messaggio . . . . .	21
ECB   Electronic Code Book . . . . .	22
CBC   Cipher Block Chain . . . . .	22
CFB   Cipher Feedback . . . . .	23
OFB   Output Feedback . . . . .	23
CTR   Counter Mode . . . . .	24
XTS   AES-XTS (XEX) Tweakable Block Cipher . . . . .	24
MACS   Message Authentication Codes . . . . .	25
ALG1-6 . . . . .	25
CMAC   Cipher-based Message Authentication Code . . . . .	25
HMAC   Keyed-hash Message Authentication Code . . . . .	25
GMAC   Galois Message Authentication Code . . . . .	25
CBC-MAC . . . . .	25
AEAD   Authenticated Encryption with Associated Data . . . . .	26
OCB   Offset Codebook . . . . .	26
CCM   Cipher Block Chaining . . . . .	26
GCM   Galois Counter Mode . . . . .	26

# Storia di AES

---

## *Introduzione*

---

AES (*Advanced Encryption Standard*) è un cifrario a blocchi simmetrico, inventato da due matematici belgi, Vincent Rijmen e Joan Daemen, da cui viene il nome *Rijndael*, nel 1998 per sostituire il precedente standard: DES (*Data Encryption Standard*).

---

## *Breve storia di AES*

---

DES era divenuto lo standard dopo un bando dell'NBS (*National Bureau of Standards*), oggi NIST (*National Institute for Security and Technology*) per trovare un buon e sicuro algoritmo per proteggere le comunicazioni private dei cittadini americani.

Venne così proposto un algoritmo chiamato *Lucifer*, sviluppato dall'IBM che dopo esser stato modificato dall'NSA (*National Security Agency*), riducendone la grandezza della chiave da 128 a 56 bits e rettificandone le funzioni contenute nell'S-box, venne designato come *Data Encryption Standard* (DES).

DES regnò per 20 anni, venne studiato in lungo e in largo dagli accademici e crittoanalisti di tutto il mondo, grazie a ciò, ci fu finalmente per la prima volta un cifrario certificato che tutti potevano studiare: nacque così il moderno campo della crittografia.

Negli anni, molti sfidarono DES e dopo diverse battaglie fu finalmente sconfitto.

L'unico modo per ovviare a questi attacchi era quello di combinare des tre volte, formando il 3DES (*Triplo DES*). Il problema di questo però era la sua lentezza.

Per questo, nel 1997, il NIST indisse un nuovo bando per cercare un nuovo algoritmo di cifratura, forte come il triplo-DES, ma veloce e flessibile.

## AES vs DES

	DES	AES
Date	1976	1999
Block size	64	128
Key length	56	128, 192, 256
Number of rounds	16	9,11,13
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but accept open public comment
Source	IBM, enhanced by NSA	Independent cryptographers

Figura 1.1: AES vs DES

Vari algoritmi competerono: Serpent, Twofish, MARS, RC6, ma alla fine spuntò Rijndael per la sua semplicità e velocità.

---

## *AES vs Rijndael*

---

AES è un'implementazione di Rijndael, divenuto l'algoritmo di cifratura standard del governo degli Stati Uniti d'America. Una differenza tra i due è che AES utilizza blocchi di dati da 128 bits, mentre Rijndael permette oltre a blocchi da 128, anche blocchi da 192 e 256 bits.

Sia AES che Rijndael permettono una grandezza della chiave di 128, 192 o 256 bits, da cui ne ricaviamo il numero di rounds: 10, 12 o 14 rispettivamente.

---

## *Cifratura simmetrica vs asimmetrica*

---

Nella cifratura simmetrica viene usata una chiave sia per la cifratura che per la decifratura di un messaggio.

La cifratura asimmetrica è basata sul concetto di chiave pubblica e chiave privata. Vengono, quindi usate due chiavi sia per la cifratura che per la decifratura. Usiamo la chiave pubblica per cifrare il messaggio e la chiave privata per decifrarlo.

Ulteriori differenze:

<b>Simmetrico</b>	<b>Asimmetrico</b>
Richiede una sola chiave sia per la cifratura che la decifratura.	Richiede due chiavi, una pubblica e una privata, una per cifrare e una per decifrare.
Lo spazio del testo cifrato è lo stesso o più piccolo del messaggio originale.	Lo spazio del testo cifrato è lo stesso o più grande del messaggio originale.
Il processo di cifratura è molto veloce.	Il processo di cifratura è molto lento.
È usato quando un grosso ammontare di dati deve essere trasferito.	È usato per trasferire piccole quantità di dati.
Fornisce solamente la confidenzialità.	Fornisce confidenzialità, autenticità e non ripudio.
La chiave usata è di solito di lunghezza 128 o 256 bits.	La lunghezza della chiave è di 2048 o più bits.
L'utilizzo delle risorse è basso.	L'utilizzo di risorse è alto.
Esempi: DES, 3DES, AES, RC4	Esempi: DSA, RSA, Diffie-Hellman, ECC, El Gamal



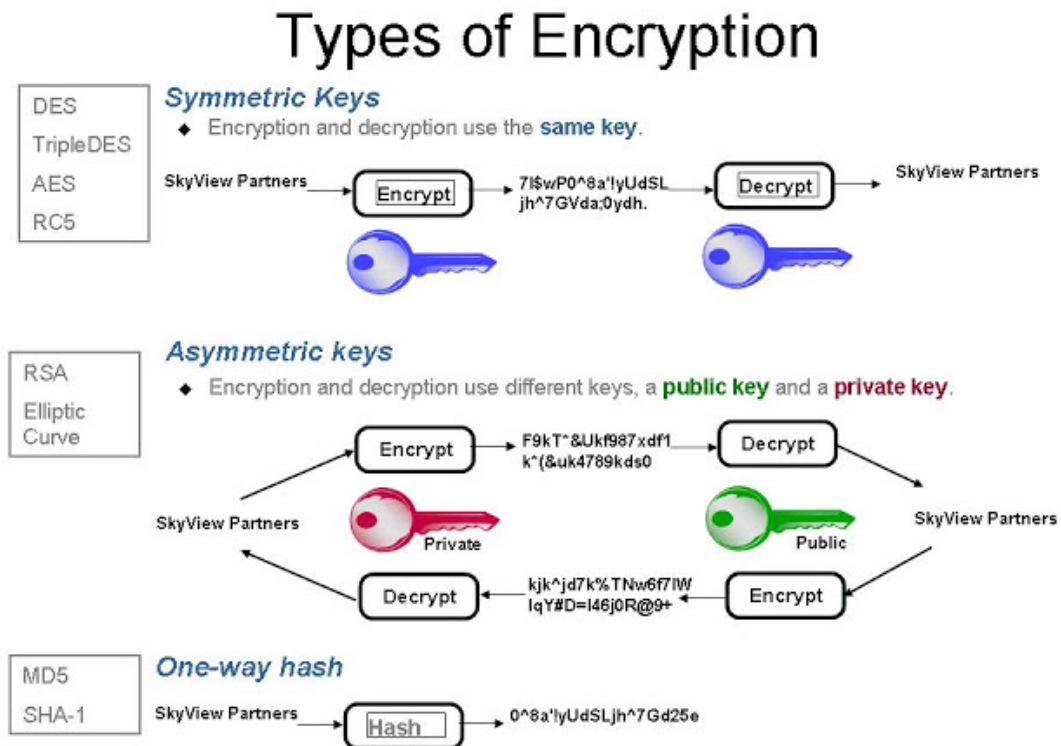


Figura 1.2: Tipi di cifratura

AES è di tipo simmetrico, quindi useremo la stessa chiave sia per criptare il nostro messaggio sia per decriptarlo.

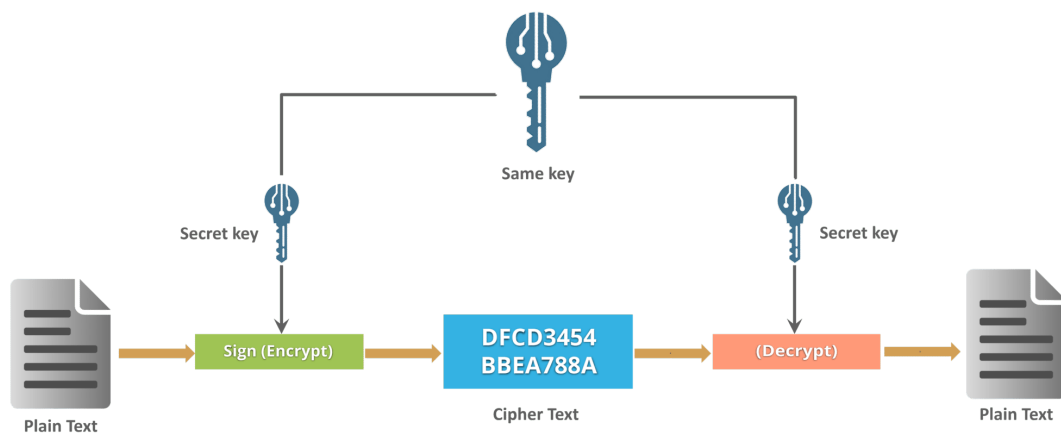


Figura 1.3: Cifratura a chiave simmetrica



# Capitolo 2

## L'Algoritmo

---

### *Introduzione*

---

In questo capitolo, tratteremo il funzionamento dell'algoritmo di AES con una panoramica dall'alto, per poi affrontare nel prossimo capitolo, più in dettaglio, la sua matematica.

---

### *I tre concetti dietro la Crittografia*

---

Alla base della crittografia, ci sono due importanti proprietà dei cifrari a chiave simmetrica, elaborati dal padre della teoria dell'informazione, Claude Elwood Shannon, ovvero: *diffusione* e *confusione*.

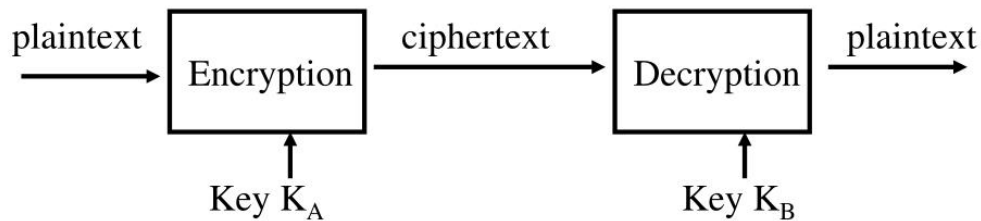
Il principio della *confusione* vela la connessione tra il messaggio originale e il testo cifrato.

La proprietà di *diffusione*, invece, riguarda lo scombussolamento della posizione dei caratteri del messaggio.

Un altro importante concetto è quello della *segretezza della chiave*, ovvero che l'algoritmo alla base del cifrario è conosciuto, è pubblico, ma la sola conoscenza di

questo non è sufficiente per poter conseguire l'accesso alle informazioni, perché per poter attingerle sarà necessario conoscere la chiave segreta.

## Confusion and Diffusion



- Terms courtesy of Claude Shannon, father of Information Theory
- "Confusion" = Substitution
  - $a \rightarrow b$
  - Caesar cipher
- "Diffusion" = Transposition or Permutation
  - $abcd \rightarrow dacb$
  - DES

Figura 2.1: Confusione e Diffusione

---

### *Una panoramica sull'Algoritmo*

---

I dati di input vengono caricati in una matrice 4x4, anche chiamata *state matrix* (matrice di stato), dove ogni cella rappresenta 1 byte di informazione e su queste compiamo diverse operazioni: *sub-bytes* (sostituzione dei bytes), *shift rows* (spostamento delle righe), *mix columns* (mescolamento delle colonne), *add round key* (aggiunta della chiave del round) per un numero di volte, di rounds pari alla grandezza della chiave.

	Key Length ( <i>Nk words</i> )	Block Size ( <i>Nb words</i> )	Number of Rounds ( <i>Nr</i> )
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Key-Block-Round Combinations

Figura 2.2: Key Size e Numero di Rounds

Nel primo round svolgiamo uno XOR tra il messaggio d'input e la chiave segreta.

Lo XOR (*EX*clusive-*OR*) bit-a-bit è un'operazione di maccheratura dei bit, dove se i due bit di input sono diversi, allora produrrà un 1 in uscita, altrimenti se sono uguali, uno zero.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Figura 2.3: Tabella della verità dello XOR

### *Perché lo XOR è usato in crittografia?*

- Lo XOR non *leaka* informazioni sull'input originale.
- Lo XOR è una *involutory function* (funzione involutoria) tale che se la applichi due volte riottiieni il testo originale.
- L'output dello XOR dipende da entrambi gli input. Non è così per le altre operazioni (AND, OR, NOT, ecc.).



Per poter elaborare i rounds, l'algoritmo ha bisogno di molte chiavi, una per round, queste vengono tutte derivate dalla chiave iniziale.

Il procedimento per ricavarle è questo:

1. Sposta la prima cella dell'ultima colonna della precedente chiave in fondo alla colonna.
2. Ogni byte viene posto in una substitution box che lo mapperà in qualcos'altro.
3. Viene effettuato uno XOR tra la colonna e una *round constant* (costante di round) che è diversa per ogni round.
4. Infine viene realizzato uno XOR con la prima colonna della precedente chiave.

Per le altre colonne, vengono semplicemente eseguiti degli XOR con la stessa colonna della precedente chiave (eccetto per le chiavi a 256 bit che hanno un procedimento un po' più complicato).

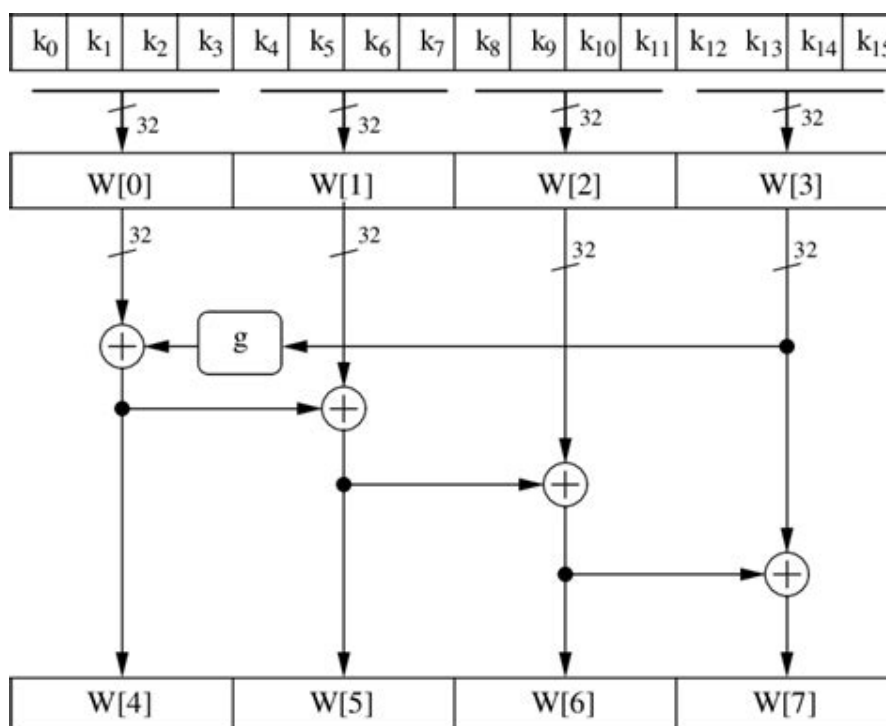


Figura 2.4: Key Expansion

Dopo aver ottenuto le chiavi, vengono compiuti i vari rounds.

Per ogni round, eseguiamo questi passaggi, tranne per l'ultimo dove non effettuiamo il passaggio delle *Mix Columns*, perché non aumenterebbe la sicurezza e semplicemente rallenterebbe:

Applichiamo il principio di *confusione* attraverso il passaggio *Sub-bytes*.

- Sub-bytes: Ogni byte viene mappato in un diverso byte attraverso una s-box. Questo step applica la proprietà di *confusione* di Shannon, perché oscura la relazione tra ogni byte.

Applichiamo la proprietà di *diffusione*:

- Shift Rows: La seconda riga della matrice viene spostata di 1 verso sinistra. La terza riga di 2 posizioni e la quarta di 3 (sempre verso sinistra).
- Mix Columns: Ogni bit delle colonne della matrice (di stato) vengono mischiate.

Applichiamo la proprietà di *segretezza della chiave*:

- Add Round Key: Viene applicata la chiave del prossimo round attraverso uno XOR.

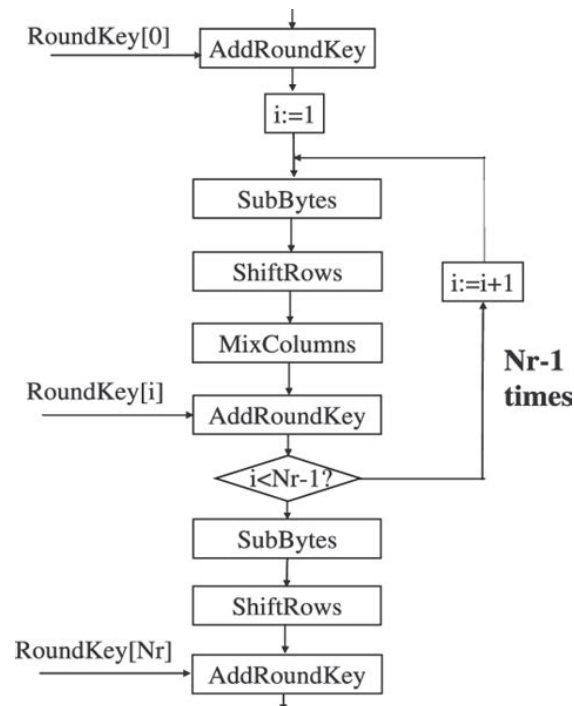


Figura 2.5: AES Rounds Flowchart

Più rounds aggiungiamo, più sicurezza, ma questo porterebbe ad un rallentamento dell'algoritmo e quindi delle performance. Per questo serve un compromesso tra sicurezza e prestazioni.

Quando AES era in sviluppo venne trovata una scorciatoia attraverso 6 rounds, per evitare ciò, sono stati aggiunti 4 rounds extra, come *margin*e di sicurezza.

## *Le modalità di AES*

AES non può essere utilizzato così com'è, ma necessita di essere adoperato in combinazione a una modalità. Una modalità è un processo/sistema/procedimento per aumentare/incrementare/trasformare/avanzare l'efficacia di un algoritmo crittografico.

Di seguito, alcune delle modalità di AES:

- ECB (*Electronic Code Book*)
- CBC (*Cipher Block Chaining*)
- CFB (*Cipher FeedBack*)
- OFB (*Output FeedBack*)
- CTR (*Counter mode*)



# Capitolo 3

## La Matematica dietro AES

---

### *Introduzione*

---

---

## *Gruppi, Anelli e Campi*

---

*Gruppo*

*Monoide*

*Gruppo abelliano*

*Anello*

*Anello commutativo*

*Anello finito*

*Campo*

*Campi vs Anelli*

---

## *Il campo di Galois*

---

*Le operazioni del campo finito*

*Addizione e sottrazione*

*Moltiplicazione*

*Esponenziazione*

L'esponenziazione è semplicemente la ripetizione della moltiplicazione.

*Logaritmi*

*Divisione*

*Inverso*

---

*Il teorema fondamentale della teoria di Galois*

---



# Capitolo 4

## L'implementazione

---

### *Introduzione*

---



# Capitolo 5

## Le modalità di AES

---

### *Introduzione*

---

---

### *A cosa servono le modalità?*

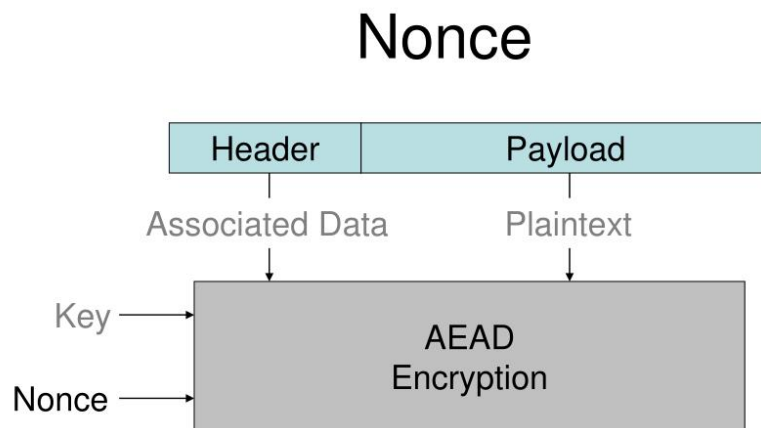
---

---

### *IV, Nonce, Salt e Pepper*

---

## *Nonce | Number Used Once*



Each encryption operation **MUST** have a distinct nonce

Figura 5.1: Nonce

### *Nonce sequenziali*

## *IV | Initialization Vector*

### *IV vs Nonce*

### *Salt*

### *Pepper*



---

## *Le modalità*

---

### *Modalità di cifratura senza integrità del messaggio*

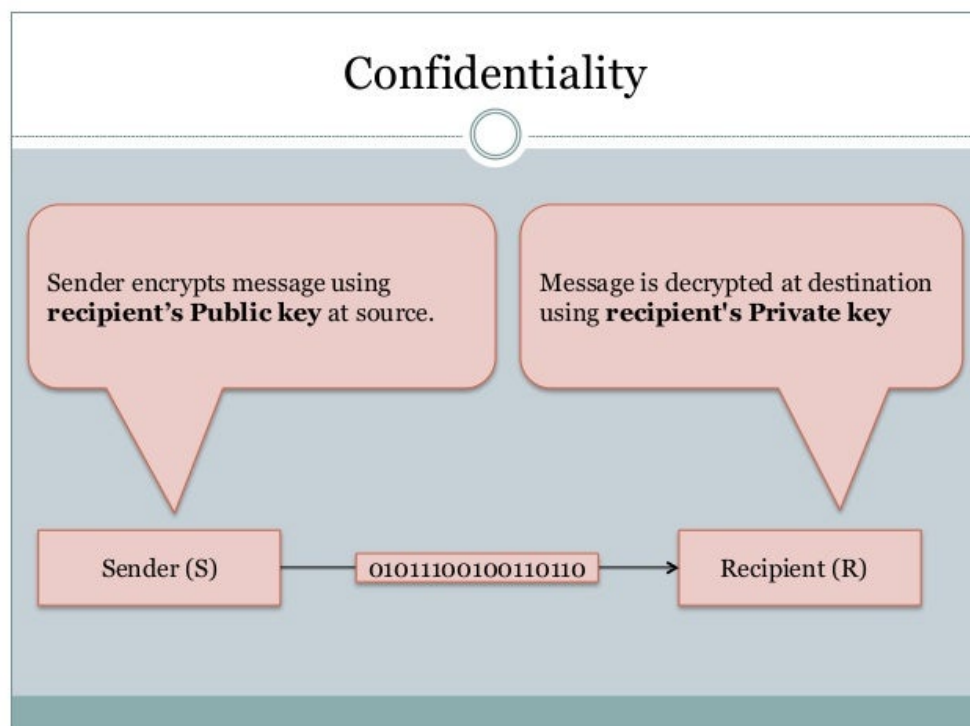


Figura 5.2: Confidenzialità

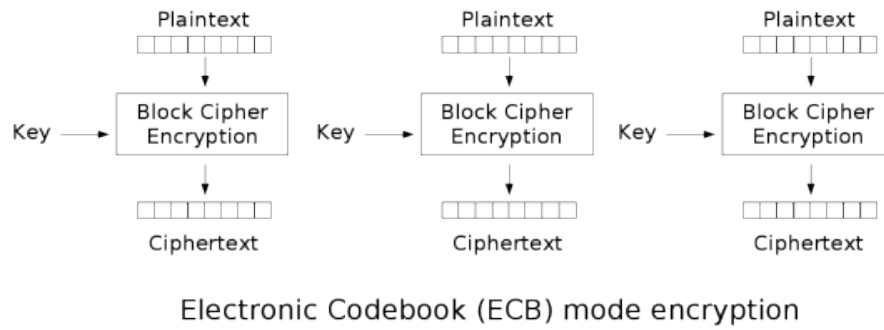
**ECB | Electronic Code Book**

Figura 5.3: ECB

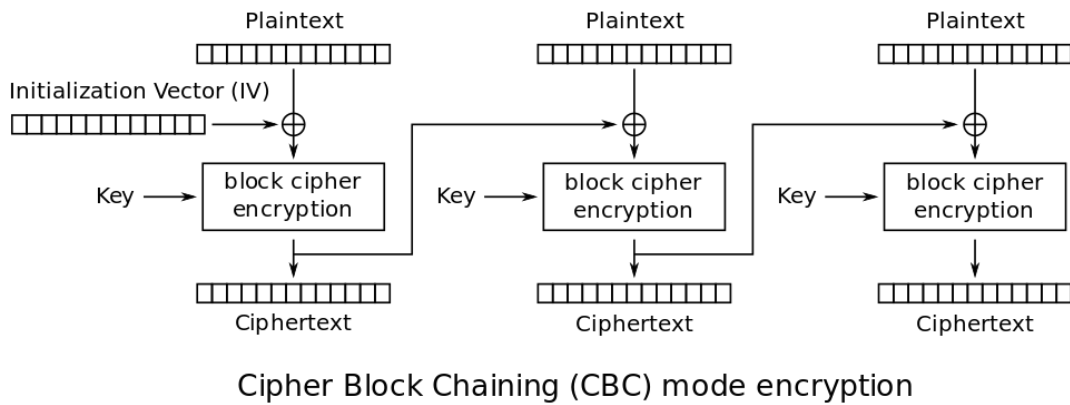
**CBC | Cipher Block Chain**

Figura 5.4: CBC

### CFB | Cipher Feedback

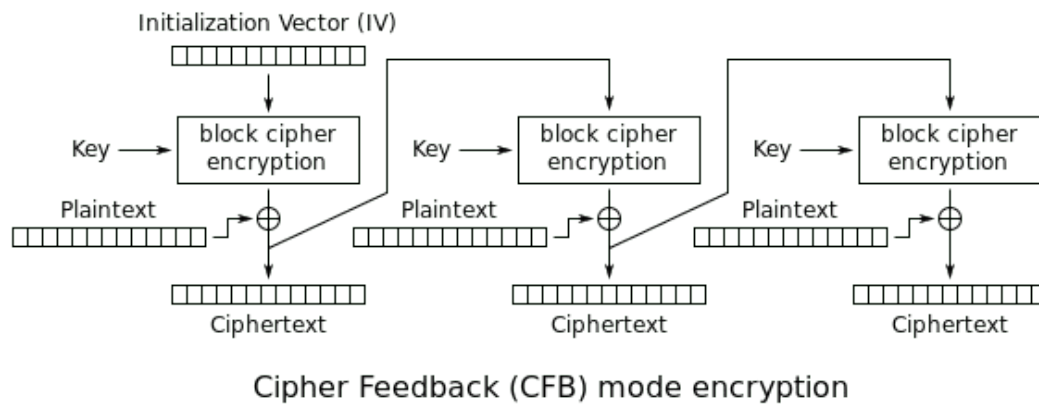


Figura 5.5: CFB

### OFB | Output Feedback

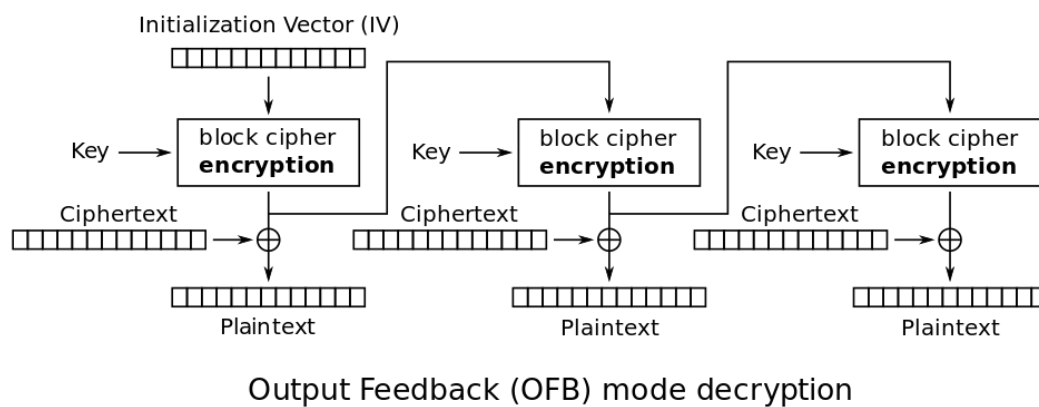
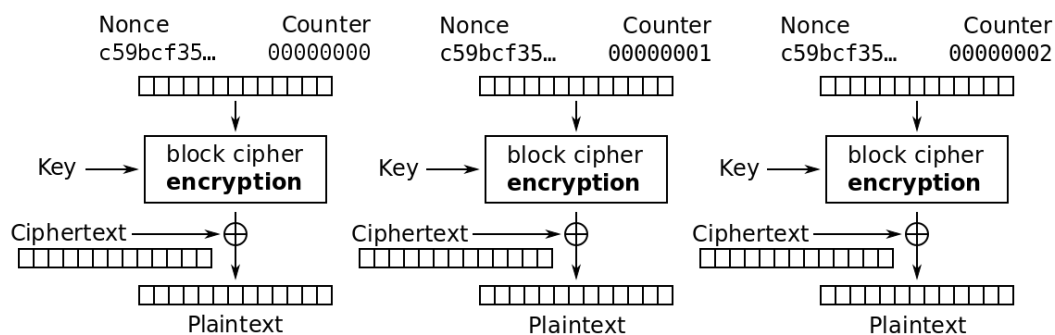


Figura 5.6: OFB

### CTR | Counter Mode



Counter (CTR) mode decryption

Figura 5.7: CTR

### XTS | AES-XTS (XEX) Tweakable Block Cipher

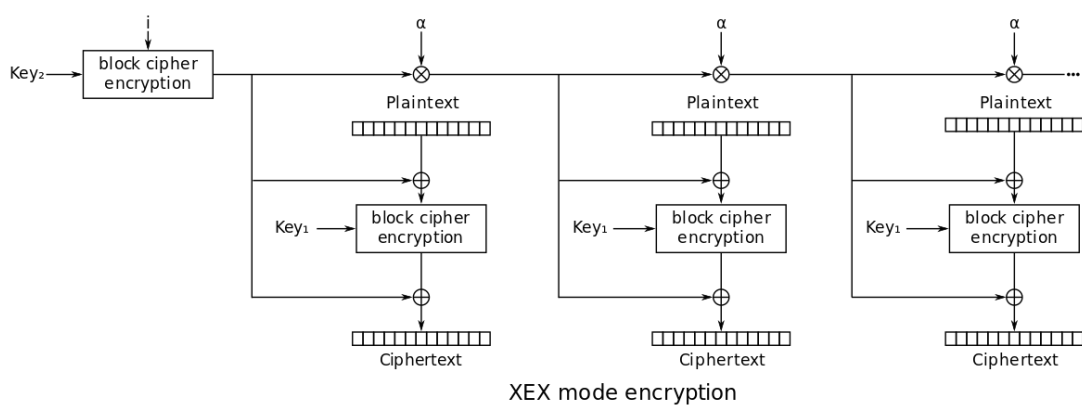


Figura 5.8: XEX

## MACS | *Message Authentication Codes*

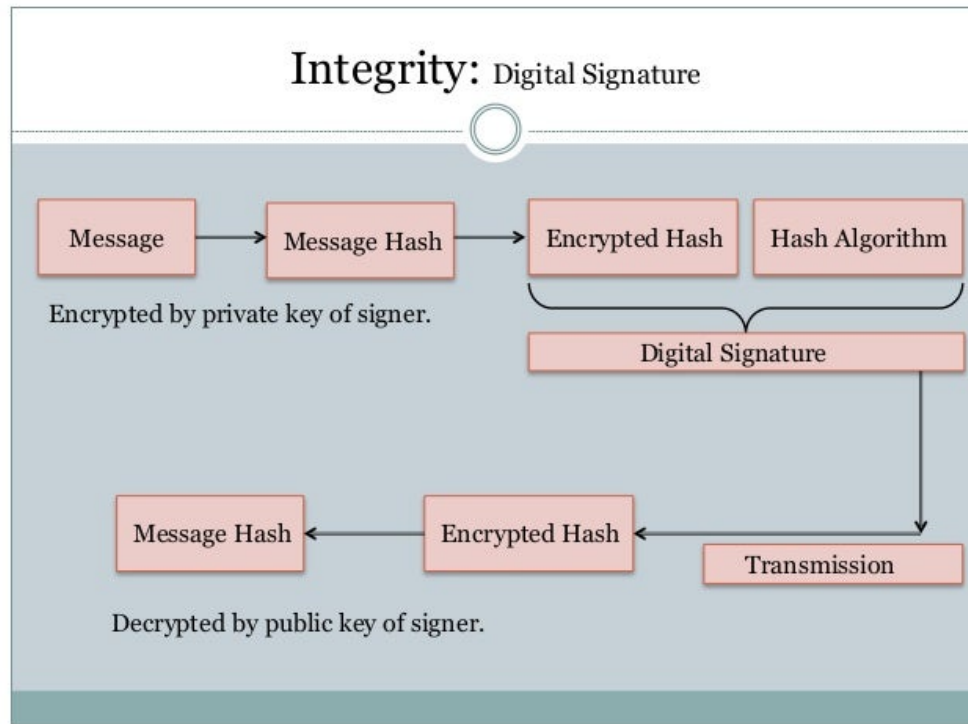


Figura 5.9: Integrità e non ripudio

### ALG1-6

**CMAC** | *Cipher-based Message Authentication Code*

**HMAC** | *Keyed-hash Message Authentication Code*

**GMAC** | *Galois Message Authentication Code*

**CBC-MAC**

***AEAD | Authenticated Encryption with Associated  
Data***

***OCB | Offset Codebook***

***CCM | Cipher Block Chaining***

***GCM | Galois Counter Mode***