Cifrari Simmetrici

Riccardo Longo

Simulazione di OTP

- One Time Pad ha sicurezza perfetta
- Non è pratico da usare:
  - Scambio delle chiavi
  - Gestione delle chiavi
- Approssimazione di OTP:
  - *Keystream* pseudorandom
  - Chiave più piccola e gestibile

Stream Cipher

- Il cifrario fa da **generatore pseudocasuale** di bit (**flusso**)
- La chiave è usata per **inizializzare** il generatore (*seed*)
- Cifratura vera e propria `XOR` **del flusso con il plaintext**
- Estremamente **veloce** e pratico in hardware
- Ottimo quando la lunghezza del plaintext non è nota e/o variabile
- **Delicato**: facile da usare in maniera errata

Sincronizzazione

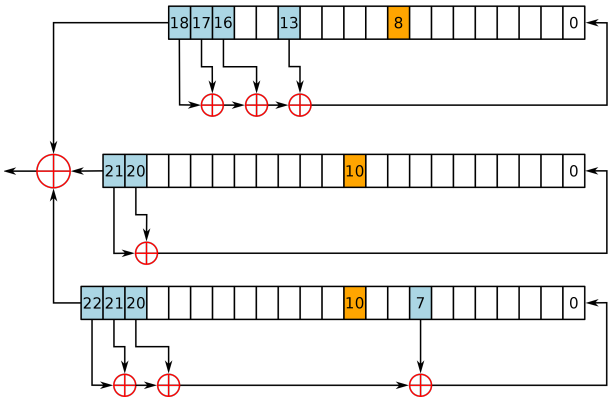
- Per decifrare i flussi di chiave devono essere **sincronizzati**
- **Stream Cipher Sicrono**
  - Si perde il sync se si **perdono o aggiungono bit** in trasmissione
  - Si sincronizza con **marker** nel CT o provando diversi **offset**
  - La corruzione di un bit in trasmissione rovina **solo un bit** nella decifratura
  - **Suscettibile ad attacchi attivi**

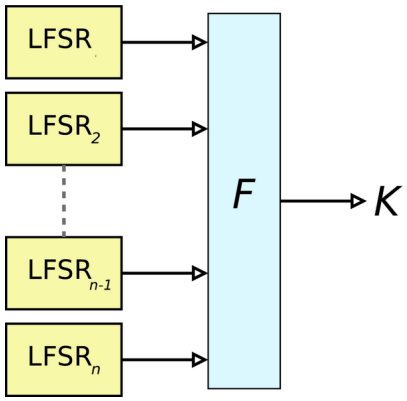
Self-Synchronising Stream Cipher

- Il flusso è **derivato dagli ultimi N bit del ciphertext**
- Ci si **risincronizza automaticamente** se si hanno N bit di ciphertext validi
- Gli **errori si propagano** ma limitatamente

LFSR

- La generazione di base è affidata a **Linear Feedback Shift Registers**
  - Facilmente implementabili in **hardware**
  - Estremamente **efficienti**
  - Proprietà controllabili **matematicamente**
- Per eliminare la linearità si usano più registri **combinati con funzioni non lineari** e/o di *clock*





Sicurezza

- Il **periodo del flusso** dev'essere lungo
- Il flusso dev'essere **indistinguibile da rumore random**
- Le **chiavi non devono mai essere riusate**
- non fornisce autenticazione ma **solo privacy**

Block Cipher

- Plaintext diviso in **blocchi di lunghezza fissa**
- Viene processato blocco per blocco
- Se la **lunghezza non è un multiplo** del blocco:
  - **Padding**
  - **Ciphertext Stealing**
  - **Residual Block Termination**
  - **Stream-like Operation Mode**

Primitiva crittografica

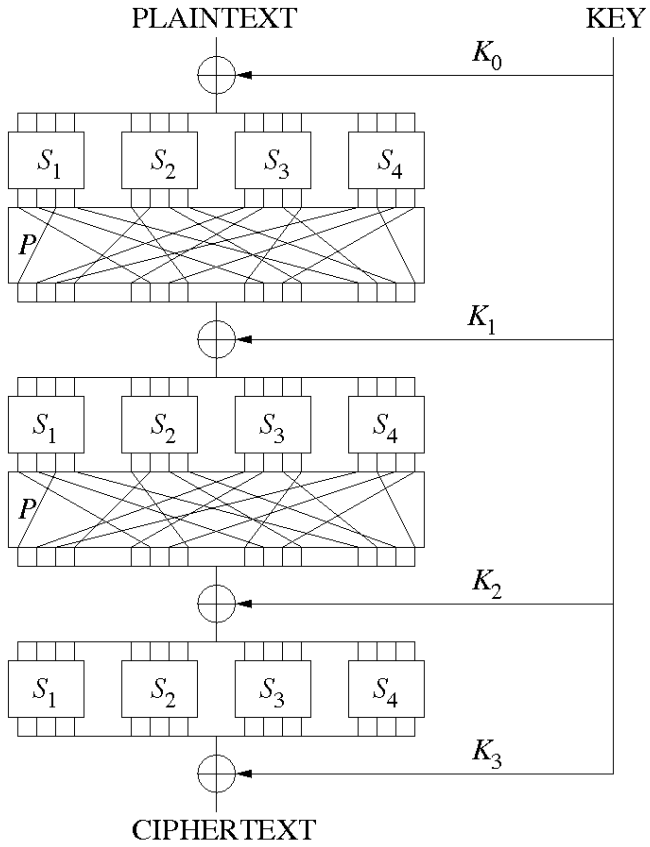
- Block cipher molto utili per **costruire altre primitive**:
  - Stream Cipher
  - Hash crittografiche
  - Generatori Pseudorandom
  - Message Authentication Codes

Round

- La maggior parte dei design si basa sull'**iterazione**
- Al blocco viene applicata una **trasformazione invertibile**
- La cifratura avviene ripetendo la trasformazione un certo numero di volte (**round**)
- Ad ogni ciclo viene usata una diversa **chiave di round** derivata dalla chiave tramite una funzione di **key-schedule**

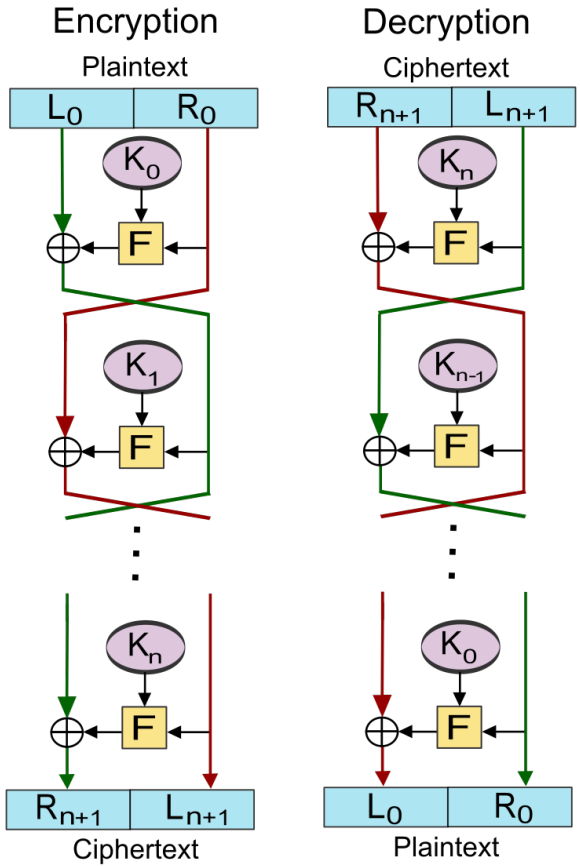
Sostituzione-Permutazione

- Struttura di **AES** (Rijndael)
- Blocco suddiviso in sottoblocchi
- Ad ognuno viene applicata una sostituzione (**s-box**)
  - mappa **biettiva**
  - molto **non-lineare**, cambia molti bit (**confusione**)
- I sotto blocchi vengono poi riuniti e vi si applica una permutazione (**mixing layer**)
  - **invertibile**
  - diffonde bene i bit di un sottoblocco in molti sottoblocchi (**diffusione**)
- `XOR` con la chiave di round



Feistel

- Blocco **diviso in due**
- Ad una metà è applicata la **funzione di round** (usando la chiave di round)
  - La funzione **non è necessariamente invertibile**
- Il risultato è `XOR`**ato con l'altra metà**
- I due pezzi sono **invertiti**



Sicurezza

- Dimensione del **blocco**
- Numero di **round**
- Lunghezza della **chiave**
- **Modo d'uso**

Operation Modes

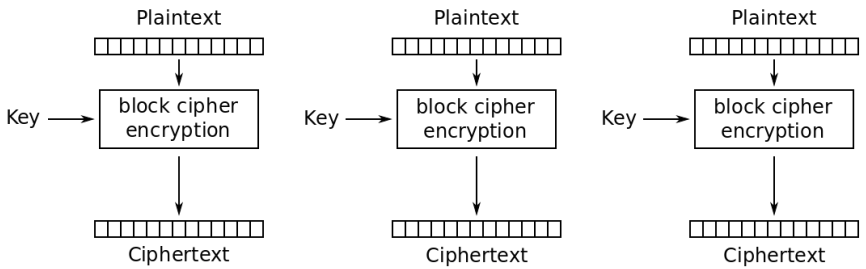
- Un cifrario a blocchi di per sé è sicuro solo per la cifratura di un **blocco singolo**
- Gli **Operation Mode** ne estendono le funzionalità:
  - Cifratura sicura di **plaintext lunghi**
  - Evitare il **padding**
  - Integrare l'**autenticazione**

Initialisation Vector

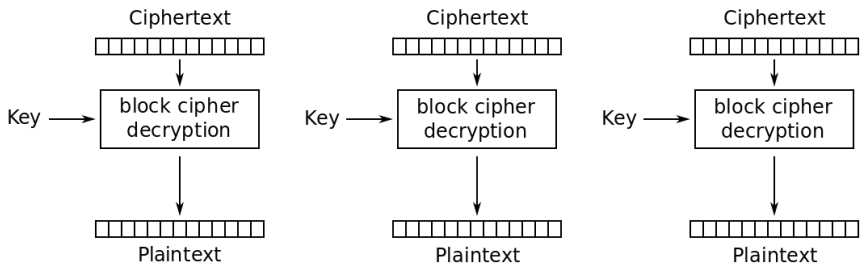
- Molti **OM** richiedono input addizionale (**IV** o **SV** o **nonce**)
- Serve per **randomizzare** la cifratura e poter cifrare blocchi uguali con la stessa chiave
- Dev'essere **unico** (mai riusato)
- In molti casi **non prevedibile** (random)
- Generalmente **non deve essere segreto**

Electronic Codebook (ECB)

- Il più semplice
- Il plaintext è diviso in blocchi (con **padding**)
- Ogni blocco è processato **indipendentemente**
- **Parallelizzabile** sia in cifratura che in decifratura, permette accesso random
- Quasi mai abbastanza sicuro: **poca confidenzialità e suscettibile ad attacchi**



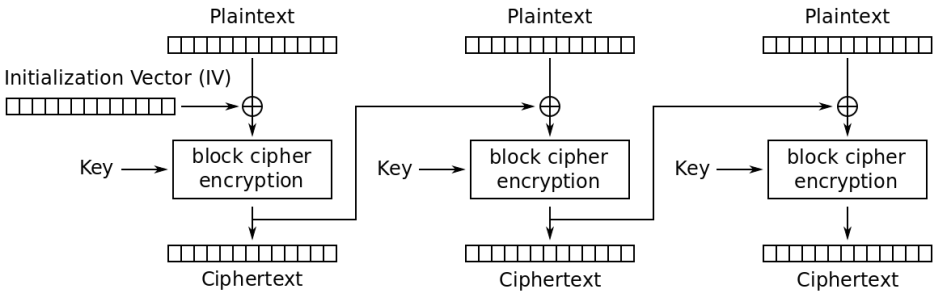
Electronic Codebook (ECB) mode encryption



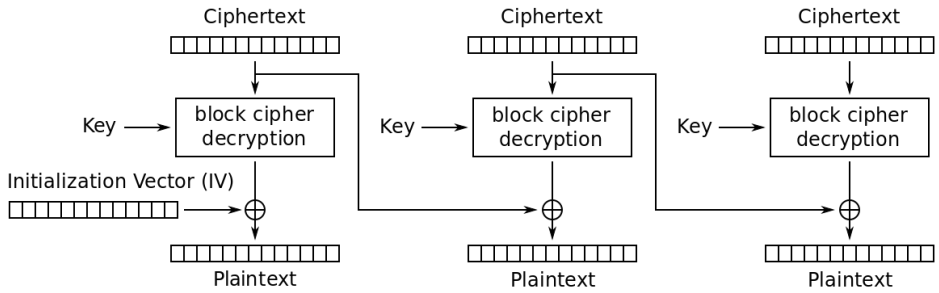
Electronic Codebook (ECB) mode decryption

Cipher Block Chaining (CBC)

- Il cifrato di un blocco è xorato con il blocco successivo prima della cifratura
- Richiede **padding** ed **IV**
- Parallelizzabile in decifratura ma non in cifratura, permette accesso random
- **Errori si propagano** al blocco successivo
- Vulnerabile ad attacchi **padding oracle**



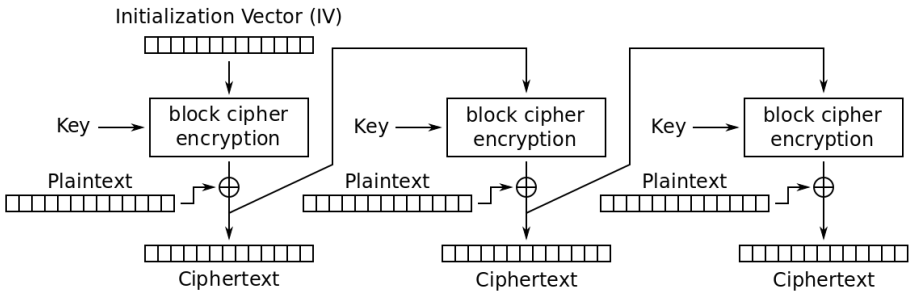
Cipher Block Chaining (CBC) mode encryption



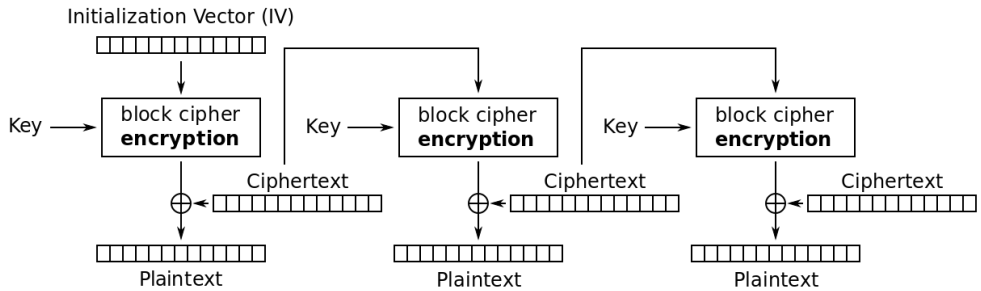
Cipher Block Chaining (CBC) mode decryption

Cipher Feedback (CFB)

- Trasforma il cifrario a blocchi in uno **stream cipher** autosincronizzante
- Il flusso è ottenuto cifrando l'**IV** e poi cifrando ancora il cifrato (`XOR` di plaintext e flusso)
- La **decifratura è parallelizzabile** ma non la cifratura, permette accesso random
- **Non richiede padding**
- Simile ad **Output Feedback** (**OFB**), dove viene cifrato un blocco del flusso per produrre il successivo (stream **sincrono**, **non parallelizzabile**)



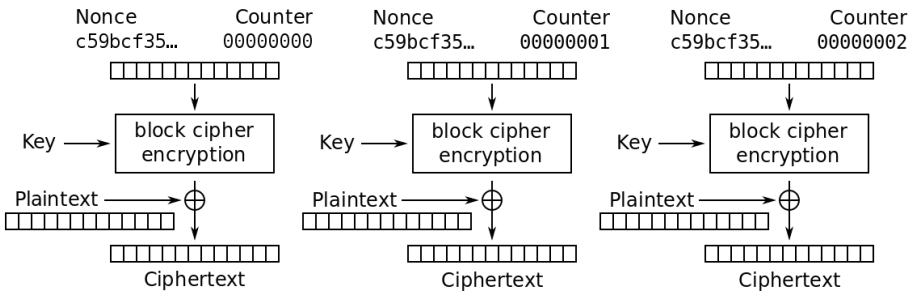
Cipher Feedback (CFB) mode encryption



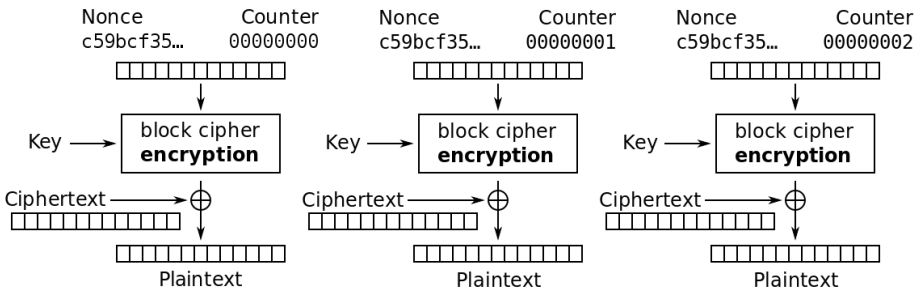
Cipher Feedback (CFB) mode decryption

Counter (CTR)

- Produce un flusso come uno **stream cipher**
- Completamente **parallelizzabile**, con accesso random
- Un **nonce** viene combinato con un **contatore** per produrre dei blocchi **unici**
- Il flusso è generato cifrando questi blocchi



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Authenticated Encryption

- Alcuni OM integrano un processo di **autenticazione** del dato
- Oltre al cifrato viene prodotto un **tag** che permette di controllare l'integrità
- **Galois Counter Mode** (**GCM**):
  - Standard, single-pass, efficiente
- **Encrypt-then-Authenticate-then-Translate**(**EAX**), **Counter with CBC-MAC** (**CCM**): anche basati su CTR, double-pass
- **Offset CodeBook**(**OCB**), efficiente, standard, selezionato dalla competizione CAESAR
- **Synthetic Initialization Vector** (**SIV**): resiste all'uso improprio del nonce