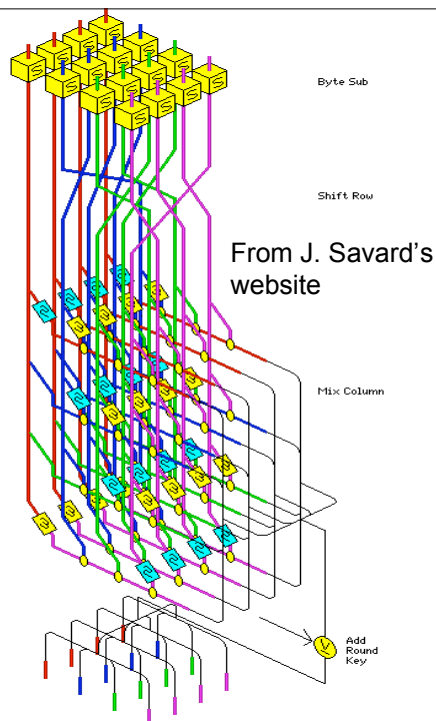


Block Ciphers

Modes of Operation for Encryption and Authentication

Definition

- A **block cipher** $E_{\pi}(\bullet)$ is a (parametrized) deterministic function mapping n -bit plaintext blocks to n -bit ciphertext blocks. The value n is called the **blocklength**.
 - It is essentially a simple substitution cipher with character set = $\{0, 1\}^n$.

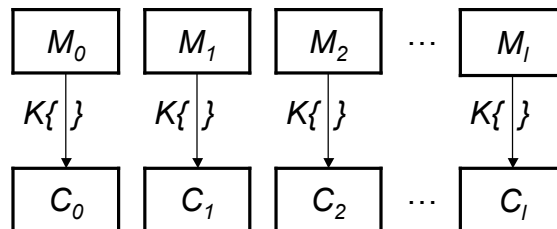


The Key to the Cipher

- The parameter **key** is a k -bit binary string.
 - It may be that the set of all keys, the **keyspace** K , is a proper subset of all k -bit binary strings. In that case, we say that the **effective key size**, or **security parameter**, provided by the cipher is $\log_2|K|$
- The keyed block cipher $E_k(\bullet)$ is a bijection, and has a unique inverse: the decryption function $D_k(\bullet)$.
 - Alternative notation: $K\{\bullet\}$ and $K^{-1}\{\bullet\}$

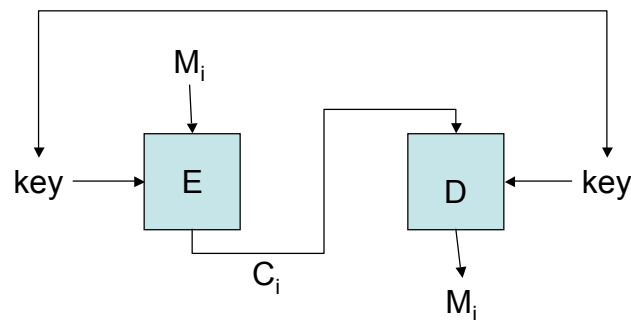
Modes of Operation

- Clearly, the block cipher can be used exactly as a substitution cipher, i.e., by encrypting each block of plaintext independently using the same key. This is called the **Electronic Codebook Mode**, or **ECB**:



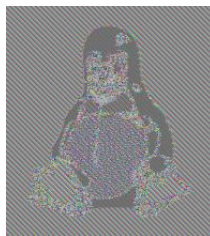
ECB (continued)

- Decryption also works block by block (inverse substitution):



ECB limitations

- ECB is the least secure mode
 - Does not not diffuse plaintext information over more than one block. Use is limited -- for instance, to transmit IVs.



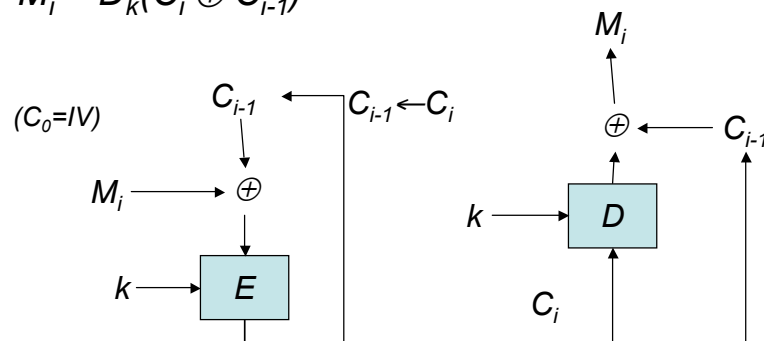
Pictures from
http://en.wikipedia.org/wiki/Cipher_Block_Chaining

Cipher Block Chaining (CBC)

- An initial vector (IV) is *xored* into the first block before encryption:
 - $C_0 = E_k(IV \oplus M_0)$
- Subsequent blocks are first *xored* with the previous cipherblock before encrypting:
 - $C_{i+1} = E_k(C_i \oplus M_{i+1})$
- The encrypted message is transmitted as
 - IV, C_0, \dots, C_l

CBC (continued)

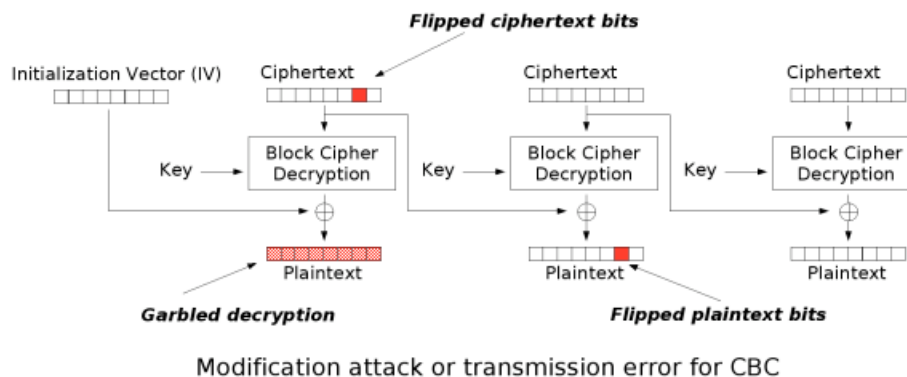
- Decryption of C_i uses knowledge of C_{i-1} (where $C_0 = IV$):
 - $M_i = D_k(C_i \oplus C_{i-1})$



CBC issues

- Not parallelizable
- A single-bit transmission error in ciphertext block C_i results in whole plaintext block P_i and the same bit in plaintext block P_{i+1} being corrupted.
- The IV should be integrity-protected
- The IV can be sent in the clear.

CBC error propagation



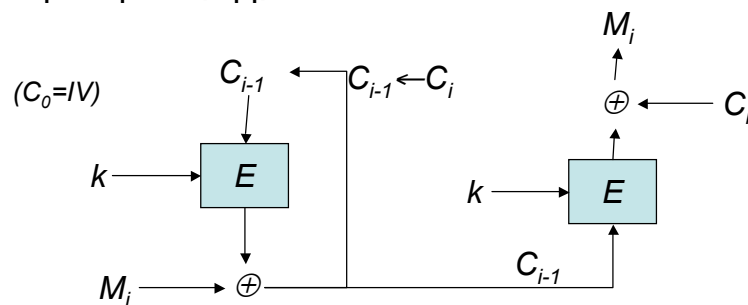
http://en.wikipedia.org/wiki/Cipher_Block_Chaining

Block Ciphers as Stream Ciphers

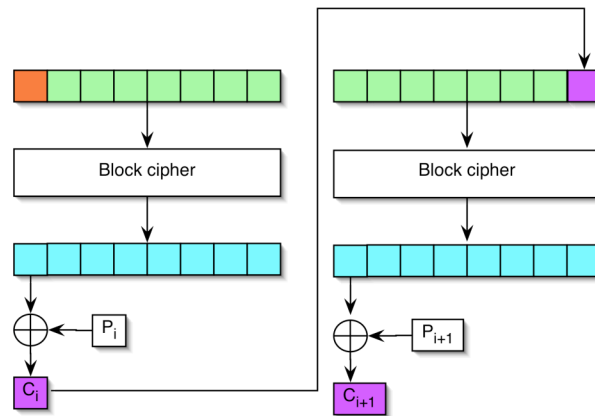
- Two modes of operation of a block cipher implement a stream cipher:
 - **Cipher Feedback Mode (CFB)**, a **Ciphertext-auto-key** stream cipher (**CTAK**)
 - **Output Feedback Mode (OFB)**, a **Key-auto-key** stream cipher (**KAK**)
 - In both cases encryption is obtained by xoring a keystream with the plaintext.
 - CFB: Keystream depends on previous ciphertext
 - OFB: Keystream depends on previous keystream

CFB

- The keystream (output of encryption) is xored into plaintext to obtain ciphertext. The ciphertext is the input for next chained encryption.
 - $C_i = M_i \oplus E(C_{i-1})$



k -bit CFB mode

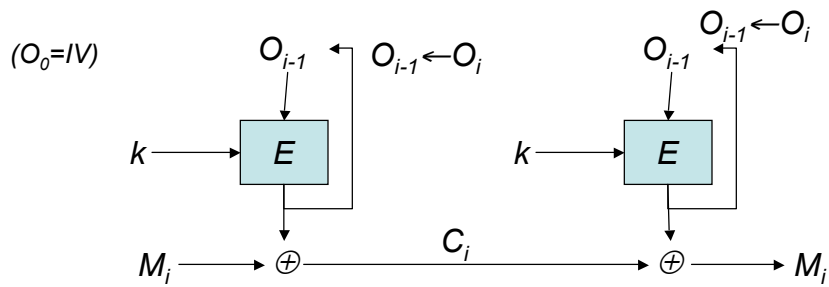


CFB issues

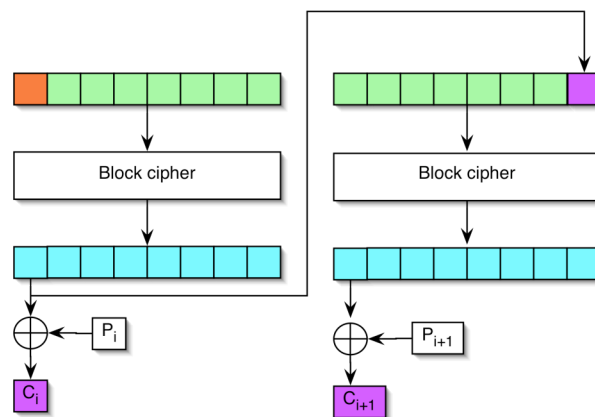
- The IV must be generated in a strongly pseudo-random fashion
- Not parallelizable (as CBC)
- Same amount of error propagation
- Self-synchronizing (as CBC)
 - If a ciphertext block is missing, only that block and the following will decrypt incorrectly.

OFB

- The keystream (output of encryption) is xored into plaintext to obtain ciphertext. The keystream is also the input for next chained encryption.
 - $C_i = M_i \oplus O_i; O_i = E(O_{i-1})$



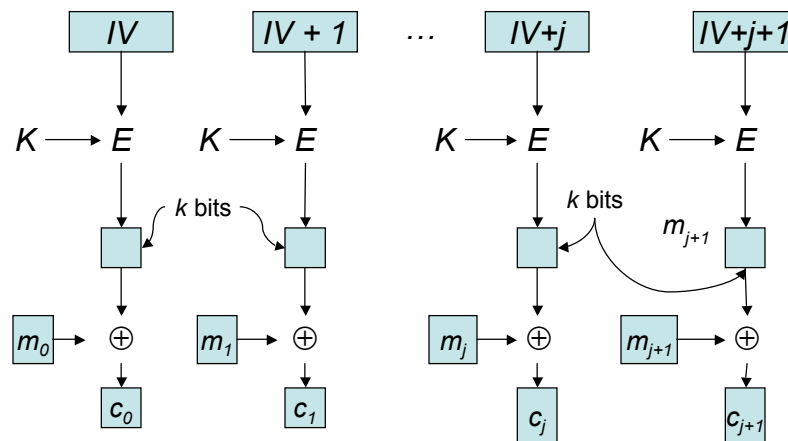
k -bit OFB mode



OFB issues

- IV repetition completely compromises security
- More parallelizable than CBC---the **key stream** is independent of the ciphertext, and can be pre-computed to enable random-access to plaintext.
- The operation of encryption and decryption must be synchronous---if a ciphertext block is missed, the two operations will not fall back in synch.

Counter Mode



Providing Integrity

- Message Authentication Code (MAC)
- CBC-residue MAC:

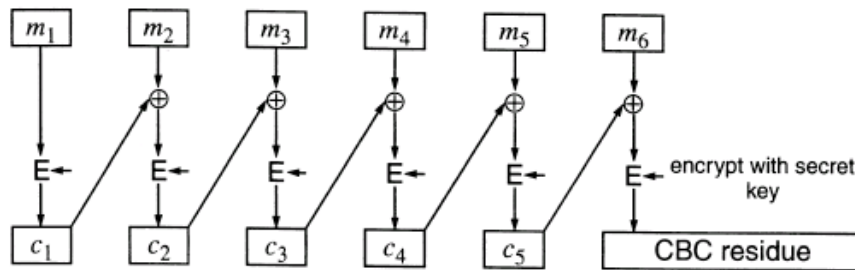


Figure 4-11. Cipher Block Chaining Residue

Encryption + Integrity

- False solution:
 - Use a weak (non-cryptographic) checksum inside CBC: May prove to be completely insecure!
- Possible solutions
 1. Use two different keys in CBC mode (expensive).
 2. Use a different authentication mechanism, such as HMAC, which still requires processing the data twice, but less computationally costly.
 3. Use another encryption mode that provides both encryption and authentication (the future?)

Some care must be taken when combining encryption with MACs, in general

Order of encryption/authentication

- Encrypt then authenticate:
 - $E_{k'}(m) \parallel \text{MAC}_{k''}(E_{k'}(m))$
- Generally secure, independent of the mode of encryption used
- Has the advantage to permit MAC verification before decryption (early compromise detection and avoidance of unnecessary cryptographic operations)

Authentication+Encryption

- Authenticate then encrypt:
 - $E_{k'}(m, \text{MAC}_{k''}(m))$
- Unsafe if a mode other than CBC is used.
- Provably secure with CBC.
- Does not permit verification before decryption.
- Authentication tag can be pre-computed, and remains associated with the original message after decryption.

Other strategies

- Encrypt and authenticate:
 - $E_k(m) \parallel \text{MAC}_k(m)$
- Not secure in general. Avoid.