Block Ciphers

CMSC 426 - Computer Security

Outline

- Elements of a Block Cipher
- Feistel Networks and DES
- Triple DES and AES
- Block Cipher Modes of Operation

Basic Elements

- *Block Size* - size in bits of a plaintext or ciphertext block.
- *Key Size* - size in bits of the secret key.
- *Round Function* - basic encryption function; iterated to form the encryption algorithm.
- *Number of Rounds* - the number of iterations of the round function.
- *Subkey Algorithm* (or *Key Schedule*) - algorithm that expands secret key into multiple round keys.

Additional Elements

- Implementation Efficiency

  Is the algorithm efficient in hardware? software?

- Ease of Analysis

  Is the algorithm easily analyzed for vulnerabilities?

Feistel Networks

Invented by Horst Feistel of IBM in 1973.

Input split into Left and Right

Non-linear function *F*

Subkeys K_1, ..., K_n

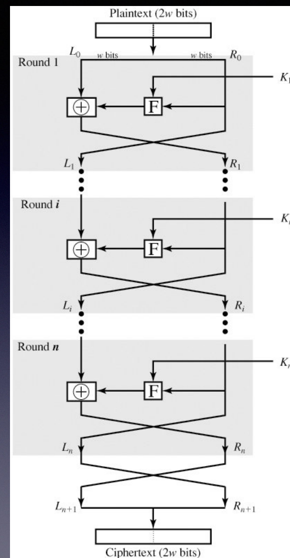


Image from Stallings & Brown, Computer Security: Principles and Practice, 2nd Ed

5

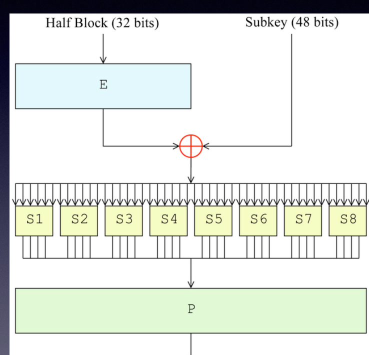
Data Encryption Standard (DES)

- DES published in 1977 (FIPS PUB 46)
- Modified Feistel Network (adds initial and final permutations)
    - 64 bit block size
    - 56 bit key size
    - 16 rounds, 16 round keys
- Decrypt using round keys in reverse order

6

DES *F*-function

- Half-block expanded to 48 bits by *E*
- Expanded half-block and Subkey are XORed
- Lookup-tables (S-boxes) S_1, ..., S_8
- Fixed permutation *P*



S-boxes were subject to intense scrutiny.

7

Problems with DES

- DES was great for its time, but the key is too small now.
- An attacker who can perform one hundred billion decryption attempts per second could break DES in about eight days.
- Cracked by EFF in 1998.

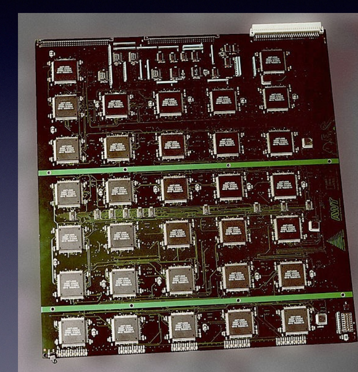


Image from Wikipedia, GNU Free Documentation License

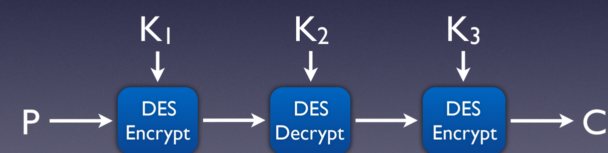
8

Triple DES and AES

9

Triple DES (3DES)

- 3DES published in ANSI X9.17 (1985); incorporated in FIPS PUB 46-3 (1999).
- Three keys; total key size 168 bits.
- Two keys ($K_1 = K_3$); total key size 112 bits.



Three-key Triple DES Encryption
(to decrypt, swap Encrypt and Decrypt and use keys in reverse order)

10

Attacking 3DES

At the rate of one *trillion* trial decryptions per second, it would take more than 10^{14} years to try 2^{112} 3DES keys.

However, **three-key 3DES is preferred** (FIPS SP 800-131A Rev. 1) due to the existence of a known-plaintext attack against two-key 3DES .

11

Advanced Encryption Standard (AES)

- Based on Rijndael block cipher; published in FIPS PUB 197 (2001).
  - Block size of 128 bits
  - Key sizes of 128, 192, and 256 bits; number of rounds is 10, 12, and 14, respectively.
  - Iterated round function, but not a Feistel Network.

12

AES Round Function

- Write input in a 4-by-4 array of bytes
- Round key w_i is 128 bits
- Round function F_i consists of the following invertible steps:
  - Substitute Bytes
  - Shift Rows
  - Mix Columns
  - Add Round Key w_i

Substitute Bytes

- Apply function *S* to each byte in array

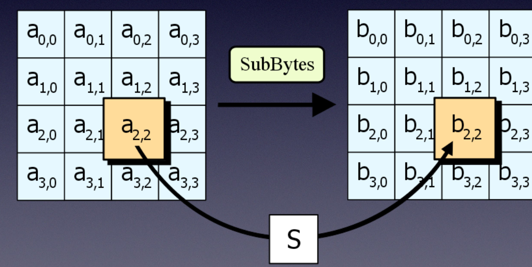


Image from Wikipedia, Public Domain

Shift Rows

- Circular shift rows by 0, 1, 2, or 3 bytes

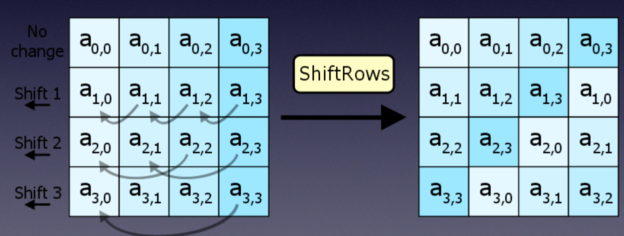


Image from Wikipedia, Public Domain

Mix Columns

- Invertible linear map applied to columns

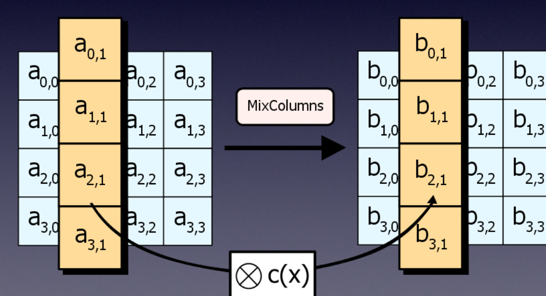
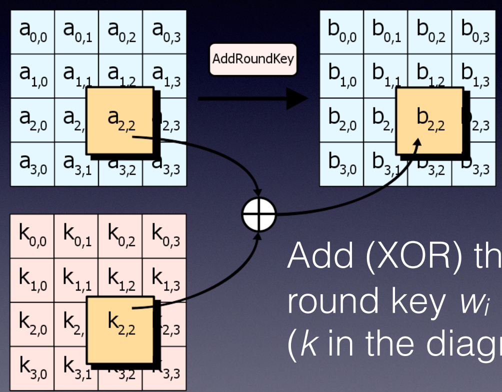


Image from Wikipedia, Public Domain

Add Round Key



Add (XOR) the round key w_i (*k* in the diagram)

Image from Wikipedia, Public Domain

AES Algorithm

- AES with 128 bit key (AES-128) encryption consists of:
  - Initial Add Round Key with w_0
  - Nine rounds F_1, ..., F_9 with $w_1,...,w_9$
  - One special round F' (no Mix Columns) with w_{10}
- Decryption similar, with keys used in the reverse order.
- AES-192 and AES-256 have 12 and 14 rounds, respectively.

Key Expansion

- Should not overlook the key expansion function!
- Takes the secret key and expands it to as many words of Subkey as are needed (11 for AES-128).
- First published “attack” on AES took advantage of the relatively simple key expansion algorithm.

Bottom Line

- For new system development, AES is the preferred block cipher for data encryption. AES-128 is appropriate in most instances; AES-192 or AES-256 for classified data.
- For legacy systems, 3DES with three keys is acceptable.
- DES is now unacceptable.

Modes of Operation

21

Outline

- Notation
- Electronic Codebook Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Counter Mode (CTR)

22

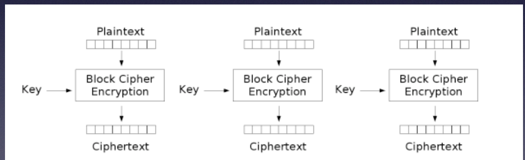
Notation

- $E_K(M)$ - encryption of message M with key K using an arbitrary block cipher.
- $D_K(C)$ - decryption of cipher C with key K using an arbitrary block cipher.
- *Arbitrary block cipher* - think DES, 3DES, or AES.

23

Electronic Codebook Mode (ECB)

- Simplest mode of operation
  - Encryption: $C_i = E_K(M_i)$, i = 1, 2, ...
  - Decryption: $M_i = D_K(C_i)$, i = 1, 2, ...



Electronic Codebook (ECB) mode encryption

Image from Wikipedia; Public Domain

24

Problem with ECB

- If $M_i = M_j$ then $E_K(M_i) = E_K(M_j)$
- Suppose a message has long constant blocks (e.g .doc files, bitmap images), then, cipher blocks will repeat in these areas.

Tux, Tux encrypted in ECB mode, Tux encrypted in a different mode
Images from Wikipedia; attributed to Larry Ewing, 1996

25

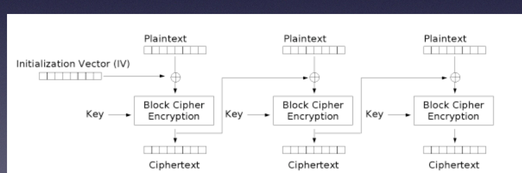
Efficiency of ECB

- Encryption and decryption can be performed in parallel. That is, multiple blocks can encrypted or decrypted simultaneously.
- Requires padding of plaintext.

26

Cipher Block Chaining Mode (CBC)

- Requires an *Initialization Vector* C_0, which is a block filled with pseudo-random values.
  - Encryption: $C_i = E_K(M_i \oplus C_{i-1})$, i = 1, 2, ...
  - Decryption: $M_i = D_K(C_i) \oplus C_{i-1}$, i = 1, 2, ...



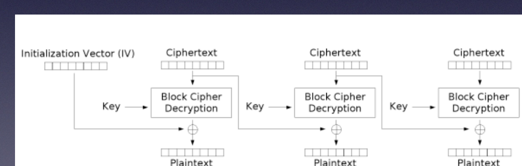
Cipher Block Chaining (CBC) mode encryption

Image from Wikipedia; Public Domain

27

CBC and ECB

- Addresses the problem we saw with ECB.
- Current cipher block depends on key *and* previous cipher block.



Cipher Block Chaining (CBC) mode decryption

Image from Wikipedia; Public Domain

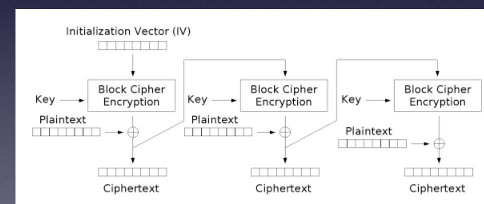
28

Efficiency of CBC

- Encryption can not be parallelized. Since encryption of a block depends on the previous block, blocks must be encrypted serially.
- Decryption can be parallelized.
- Requires padding of plaintext.

Cipher Feedback Mode (CFB)

- Requires an *Initialization Vector* C_0, which is a block filled with pseudo-random values.
  - Encryption: $C_i = E_K(C_{i-1}) \oplus M_i,\ i = 1, 2, \ldots$
  - Decryption: $M_i = E_K(C_{i-1}) \oplus C_i,\ i = 1, 2, \ldots$



Cipher Feedback (CFB) mode encryption

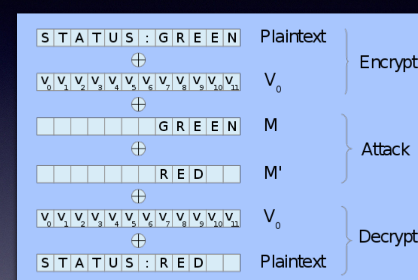
Image from Wikipedia; Public Domain

CFB is really different

- CFB operates as a *stream cipher*. It uses the block cipher as a *key generator*, producing random blocks which are xor-ed with plaintext blocks.
- Encryption and decryption are identical.

Attack on CFB Mode

- CFB mode (really any stream cipher) is susceptible to a type of attack.
- Depends on having structured plaintext.
- Requires "active" access to the communications media.



The message "STATUS: GREEN" is changed to "STATUS: RED".

Efficiency of CFB

- Encryption can not be parallelized. Since encryption of a block depends on the previous block, blocks must be encrypted serially.
- Decryption can be parallelized.
- Plaintext does **not** need to be padded. Can discard “left over” additive key.

33

Counter Mode (CTR)

- An efficient stream cipher mode
- Requires a pseudo-random seed or nonce S
  - Encryption: $C_i = E_K(S + i - 1) \oplus M_i$, i = 1, 2, ...
  - Decryption: $M_i = E_K(S + i - 1) \oplus C_i$, i = 1, 2, ...

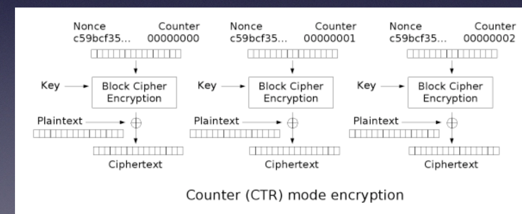


Image from Wikipedia; Public Domain

34

Efficiency of CTR

- Encryption can be parallelized. Only need the seed S and block number.
- Decryption can be parallelized.
- Plaintext does **not** need to be padded. Can discard “left over” additive key.

35

Summary

	Parallel Encrypt	Parallel Decrypt	Padding Required	Stream Cipher	Repeats in Cipher[1]
ECB	✔	✔	✔		✔
CBC		✔	✔		
CFB		✔		✔	
CTR	✔	✔		✔	

[1]Encrypting structured or repeating plaintext results in repeating cipher blocks.

36

Next time: Stream Ciphers, Hashing, MACs

37