Data Encryption Standard (DES)
Final remarks

How did they design DES?

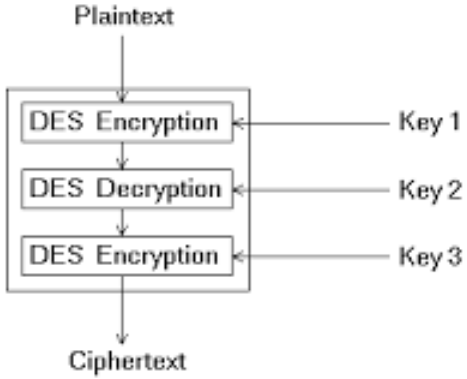
- It seems just a sequence of shuffles of bits (that anyone can design), is it enough to design such an algorithm?
- Unfortunately, the design process of DES is a secret.
- Probably, some details have been chosen at random, some other details have been chosen to prevent specific attacks, finally some other details have been chosen for fitting the hardware and software means available when DES has been developed (about 1970).
- For instance at the beginning IBM implemented 128 bits key. As a result of discussions involving external consultants including the NSA, the key size was reduced from 128 bits to 56 bits to fit on a single chip (or perhaps for other reasons! See next slide.)
- There are theoretical and experimental studies on the performance of DES. However they are really specific and therefore we do not see them in this course.

Data Encryption Standard (DES)
Final remarks

Use of a 56-bit key is one of the most controversial aspects of DES.

- Even before DES was adopted, people outside of the NSA complained that 56 bits provided inadequate security.
- The disadvantage of using 8 bits of the key for parity checking is that it makes DES considerably less secure.
- People have suggested that US government consciously decided to weaken the security of DES just enough so that NSA would be able to break it.
- In January, 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes! (given a <plaintext, ciphertext> pair, it calculates the used key).

3DES (overview)



- Currently DES is considered insecure.
- 56 bits were sufficient when that algorithm was designed, nowadays brute-force attacks are feasible.
- Triple DES or 3DES applies DES three times to each data block.
- 3DES uses three DES keys, K_1, K_2 and K_3, each of 56 bits (excluding parity bits).
- Each triple encryption encrypts one block of 64 bits of data.
- 3DES currently is considered secure (it has been not break yet) but it is slow.

3DES

- The standards define three keying options:
  - Keying option 1: All three keys are independent.
    - Strongest: 3 × 56 = 168 independent key bits.
  - Keying option 2: K_1 and K_2 are independent, and $K_3 = K_1$.
    - 2 × 56 = 112 independent key bits.
  - Keying option 3: All three keys are identical, i.e. $K_1 = K_2 = K_3$.
    - It offers backward compatibility with DES.

Advanced Encryption Standard (AES) (very short overview)

- AES is based on the Rijndael cipher developed by Vincent Rijmen and Joan Daemen
  - Rijndael is a family of ciphers with different key and block sizes (in principle any multiple of 32); block and key sizes can be different.
  - AES has a fixed block size of 128 bits, and a key size of 128 (AES-128), 192 (AES-192), or 256 (AES-256) bits.

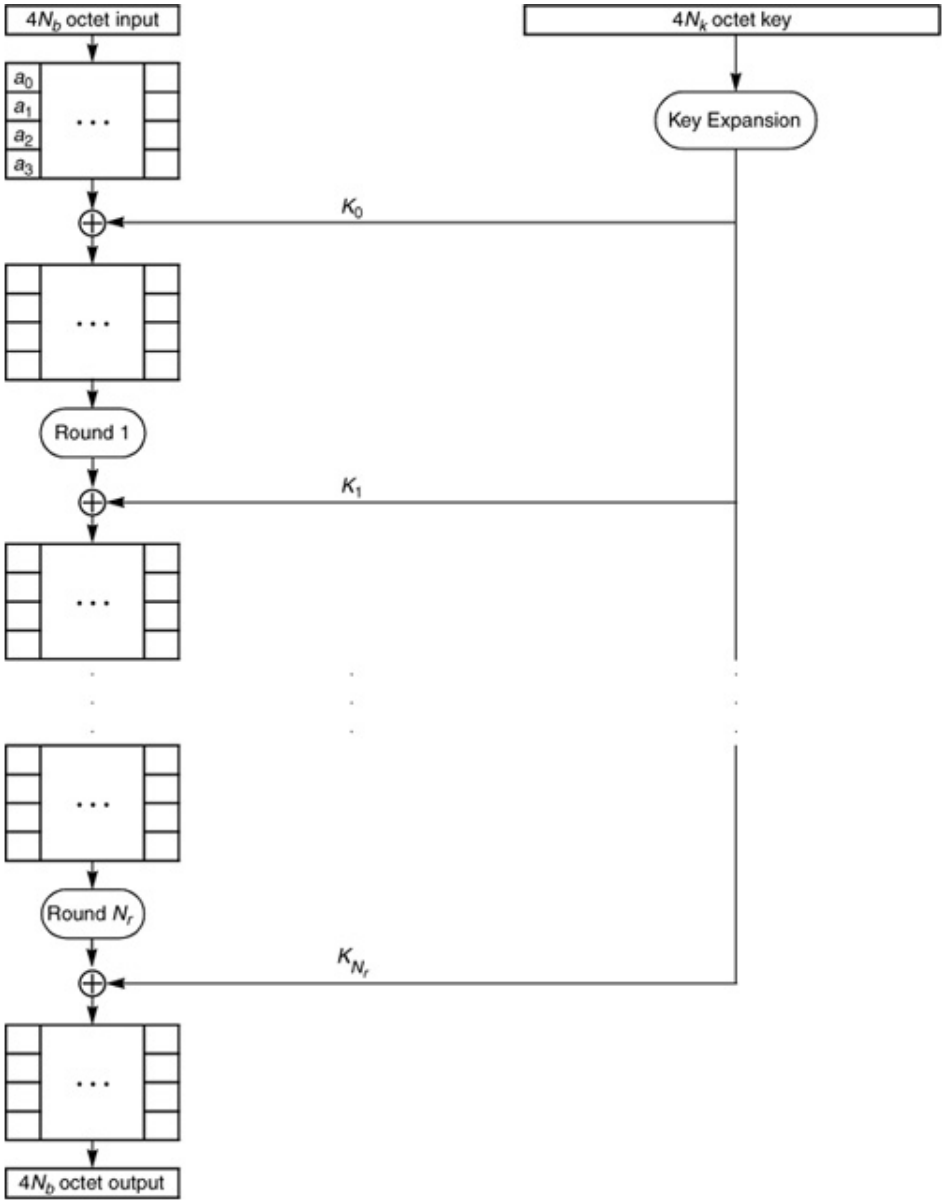
Advanced Encryption Standard (AES) (very short overview)

- Rijndael uses three parameters, the third is derived from the first two
  - The block size N_b: the number of 32-bits words in a block.
    - In AES N_b=4
  - The key size N_k: the number of 32-bits words in the key.
    - In AES-128 N_k=4; In AES-192 N_k=6; In AES-256 N_k=8;
  - The number of rounds N_r: It is larger for longer blocks/keys: $N_r = 6 + \max(N_b , N_k)$.
    - In AES-128 N_r=10; In AES-192 N_r=12; In AES-256 N_r=14;

Advanced Encryption Standard (AES) (very short overview)

- State: an array of $4N_b$ octets bits.
- Initially, the state is the input block.
- After N_r rounds the state is the output block.
- Before round 1, between rounds and after round N_r, the state is XORed with the (expanded) key.
- The final state is the output block.

Advanced Encryption Standard (AES) Overview



Advanced Encryption Standard (AES) Primitive operations

Rijndael uses four primitive operations:

- XOR.
- An octet-by-octet S-box (given by a specific table).
- A rearrangement of columns comprising rotating a row or column by some number of cells.
- An operation called MixColumn which replaces a 4-octet columns with another one.
  - It exploits a given table.
  - It exploits a network with some XOR operations on the columns/octets.

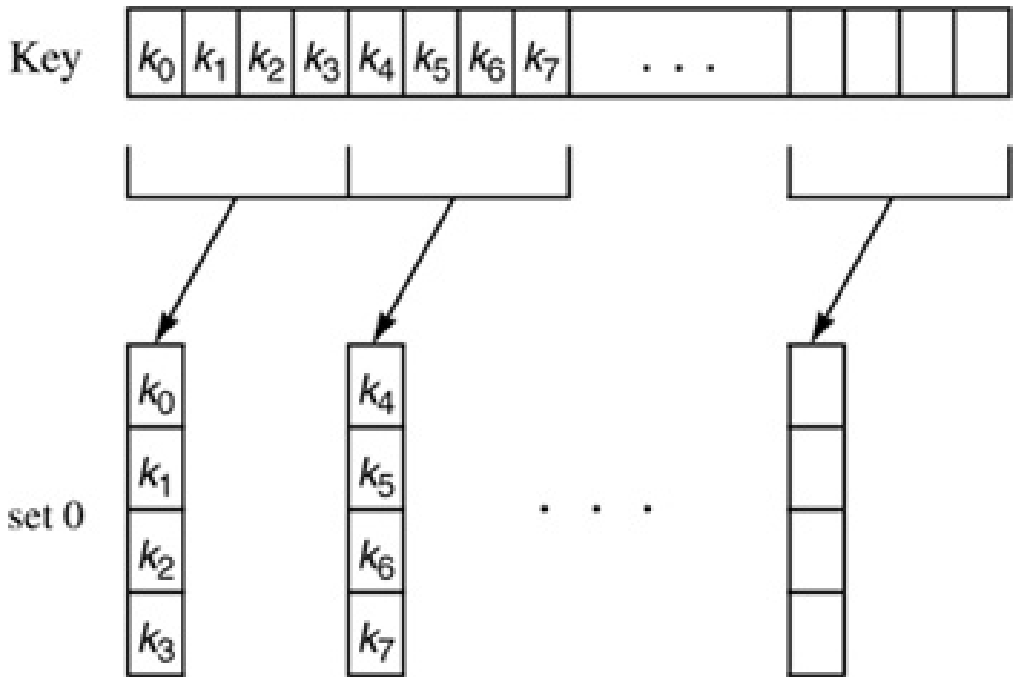
Inverse primitive operations:

- XOR is its own inverse.
- The S-box exploits a different table.
- The inverse of a rotating operation is just rotating in the opposite direction by the same amount.
- The inverse of MixColumn exploits the same network but with a different table.

Advanced Encryption Standard (AES)
Key expansion

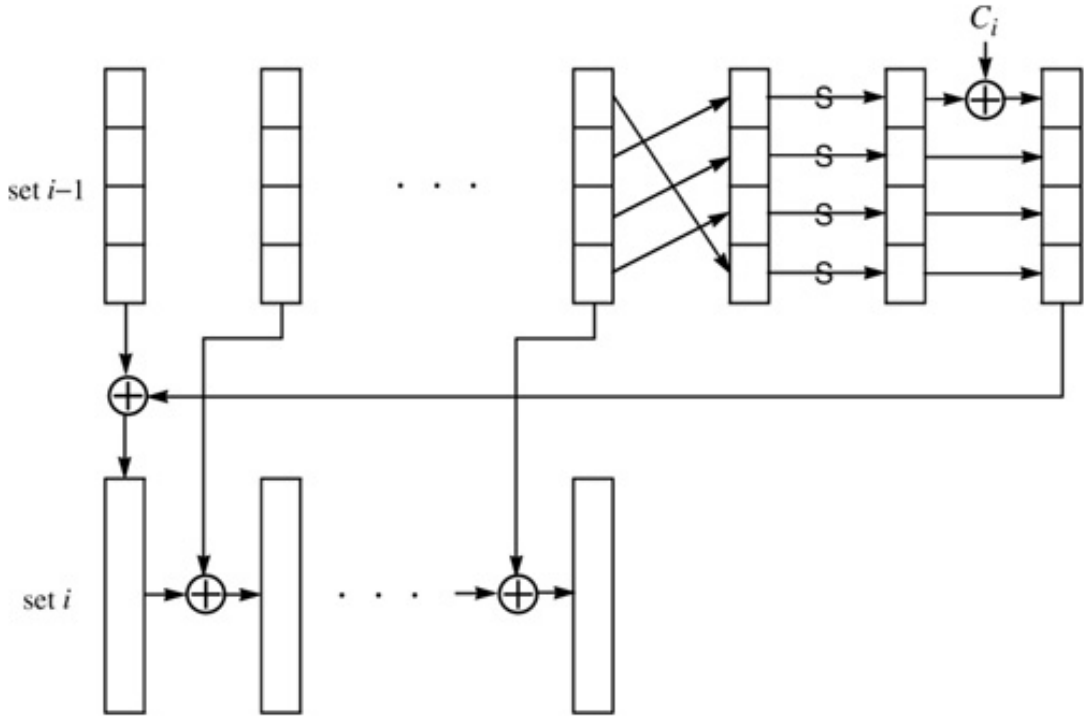
- Key expansion starts with the key arranged N_k columns of 4 octets each, and iteratively generates the next N_k columns of the expanded key.
- In the figure, each $k_0, k_1, \ldots, k_{4N_k - 1}$ is an octet (8 bits)

- Creation of set 0:



Advanced Encryption Standard (AES)
Key expansion

- Generation of set i > 0
- Column 0 of set i is obtained by
  - Rotating the last column of the (i-1)th set upward one cell,
  - Applying the S-box to each octet,
  - XORing with a constant C_i, into octet 0.
  - XOR with column 0 of set i-1
- The rest of the columns
  - XOR the previous column with the corresponding column from set i-1.



Advanced Encryption Standard (AES) Key expansion

- Key expansion terminates as soon as $(N_r+1)N_b$ columns of expanded key have been generated.
  - this may happen in the middle of a set.

Advanced Encryption Standard (AES) Rounds

Each round is an identical sequence of the following three operations:

1. Each octet of the state has the S-Box applied to it.
2. Row 1 of the state is rotated left 1 column.

   Row 2 of the state is rotated left 2 columns

   Row 3 of the state is rotated left 3 columns .

   Each column of the state has MixColumn applied to it. Round N_r omits this operation.

Advanced Encryption Standard (AES) Inverse rounds

- Since each operation is invertible, decryption can be done by performing the inverse of each operation in the opposite order from that for encryption, and using the round keys in the reverse order.
- We can also make decryption having the same structure as encryption.
  - We use the round keys in the opposite order,
  - We apply InvMixColumn to each column of all but the initial and final round keys.
  - Then, each inverse round is an identical sequence of three operations:
    1. Each octet of the state has the inverse S-Box applied to it.
    2. Row 1 of the state is rotated right 1 column.
       Row 2 of the state is rotated right 2 columns.
       Row 3 of the state is rotated right 3 columns.
    3. Each column of the state has InvMixColumn applied to it. Round N_r omits this operation.