TOPIC: 6.857 L04 (Part II)

DATE: 3/1/2021

FILE UNDER:

PAGE: L04-2

Admin:

- Put project idea(s) on Piazza
- next Monday: Pset #1 due, Pset #2 out

Today:

- Part I (Yael Kalai)

~~ Break ~~ (talk about projects)

- Part II (Ron Rivest)
  - intro to block ciphers
  - history of AES
  - structure of AES
  - using AES in CTR mode (counter mode)
  - using AES in CBC mode (cipher-block chaining) (and CBC-MACs)

Readings:

Katz & Lindell §6.2.5 (pp. 223-225)

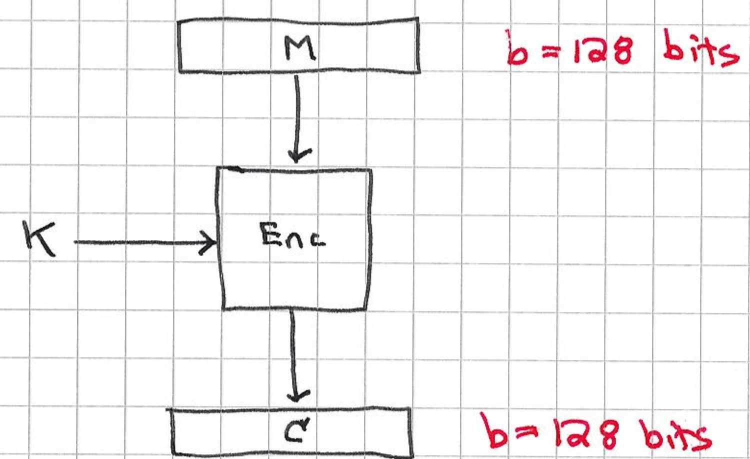
Wikipedia - AES

Block ciphers

fixed input size ("block size") (128 bits for AES)

fixed output size (128 bits for AES)

fixed key size (128, 192, or 256 bits for AES)



Issue: dealing with variable-length messages

"Ideal block cipher"

For each key K, Enc is a randomly chosen permutation ((1-1 map) of $\{0,1\}^b$ to itself)

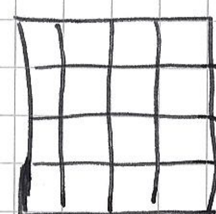
TOPIC:

DATE:

FILE UNDER:

Advanced Encryption Standard (AES) Contest

- Announced in 1997
- 15 algorithms submitted:
  RC6, Mars, Twofish, Rijndael, ...
- Winner (Rijndael) announced in 1999 ⇒ AES
  3 versions: 128, 192, or 256-bit key
  10, 12, or 14 "rounds"
- Byte-oriented design: Galois field $GF(2^8)$
- View input/state/output as 4 x 4 byte array:



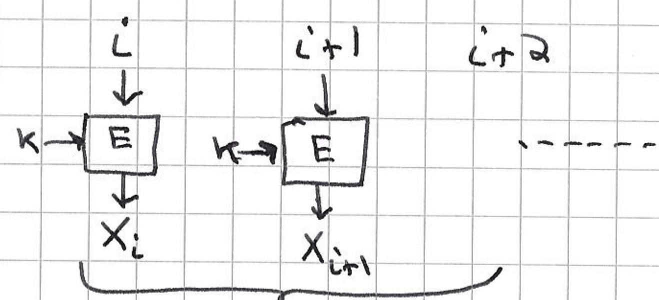
$4 \times 4 \times 8 = 128$ bits

- For 10-round version with 128-bit keys:
  - Derive 11 "round keys" each 128 bits
  - Initialize 4x4 array to message block
  - In each of 10 rounds:
    1. XOR round key into state
    2. Substitute bytes (lookup table)
    3. Rotate rows (by different amounts)
    4. Mix each column (by linear opn)

    (In last round, using another XOR key instead of mixing columns again)
  - Output final state
- Each of (1)-(4) is invertible, knowing key
- Some Intel CPUs have support for AES

Counter mode (CTR mode)

- Sender generates a random "starting value" i
- Sender encrypts $i, i+1, \ldots$ (counting up)



use as "pad" for OTP scheme
only use as many bits as needed...

$C_1 = M_1 \oplus X_i$
$C_2 = M_2 \oplus X_{i+1}$
$\ldots$ (last one may be partial)

- Sender xmits:

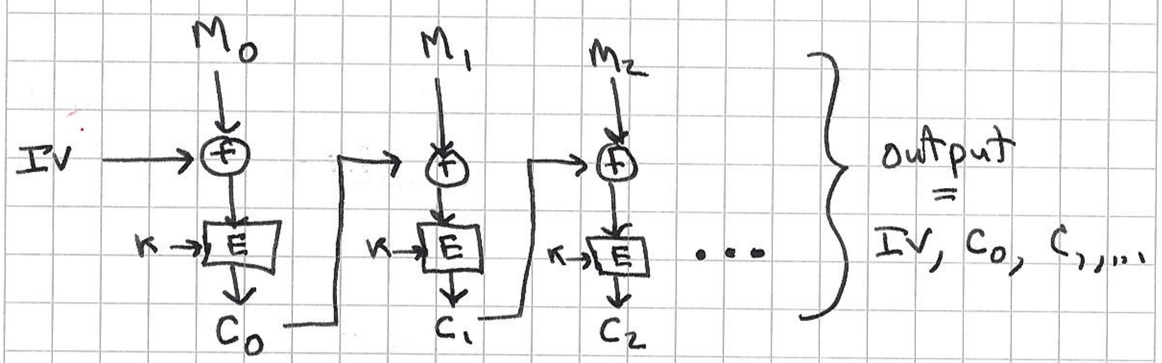
$\underbrace{i, C_1, C_2, \ldots}$

length $= |M| + |i|$, little "message expansion"

- Decryptor regenerates pad $X_i, \ldots$ using knowledge of i and K
- No need to run AES in "decryption" mode!
- Note: handles variable-length inputs!
- confidentiality only — no authentication

Cipher Block Chaining Mode (CBC)

IV = Initialization vector



If you need to handle messages that are not of length that is a multiple of block size, pad (append) a "1" (always) and as many zeros as you need to fill up a block.

CBC-MAC

Compute a tag for message authentication

$$tag = E(K_2, CBC(k, M))$$

last block of CBC encryption, using IV=0