Admin:

Project!
Pset #2 out

Today:

- AES ✓
- Ideal cipher ✓
- Modes of operation (ECB ✓, CTR ✓, CBC to do)
- IND-CCA security (to do)
- MACs (to do)
- Combining Enc & MACs - authenticated encryption modes (to do)

Readings:

Katz - Chapter 6 & 4

Wikipedia - AES

AES

"Advanced Encryption Standard" (U.S. govt)

Replaces DES

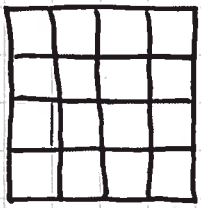
AES "contest" 1997-1999:
15 algorithms submitted: RC6, Mars, Twofish, Rijndael, ...
Winner = Rijndael (by Joan Daemen & Vincent Rijmen, (Belgians))

Specs: 128-bit plaintext/ciphertext blocks
128, 192, or 256-bit key
10, 12, or 14 rounds (dep. on key length)

Byte-oriented design (some math done in Galois field $GF(2^8)$)

View input as 4x4 byte array:

$4 \times 4 \times 8 = 128$



For version with 128-bit keys, 10 rounds:

- Derive 11 "round keys", each 128 bits (4x4x byte)
- In each round:
  1. XOR round key
  2. Substitute bytes (lookup table)
  3. Rotate rows (by different amts)
  4. Mix each column (by linear opn)

(last round has another round key xored in instead of mix-column)

- Output final state

See readings for details.
There are very fast implementations. Also Intel has put supporting hardware into its CPU's.

Security: Good; perhaps # rounds should be a bit larger...

Ideal cipher model

For practical purposes, can treat AES as ideal block cipher:

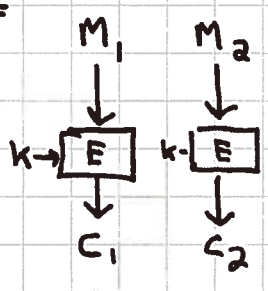
> For each key, mapping Enc(K, •) is a random independent permutation of $\{0,1\}^{128}$ to itself.

Modes of Operation:

How to encrypt variable-length messages? (using AES)

"ECB" = "Electronic code book"
"CTR" = "Counter mode"
"CBC" = "Cipher-block chaining" (& CBC-MAC)
"CFB" = "Cipher feedback"
... (others...)

ECB:



← divide data into b-bit blocks, where b = input block size

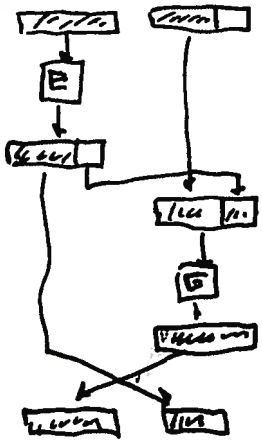
To handle data that is not a multiple of b bits in length:
- Append a "1" bit (always)
- Append enough "0" bits to make length a multiple of b bits.

This gives invertible (1-1) "padding" operation.
Pad before encryption; unpad after decryption, always!

ECB preserves many patterns: repeated message blocks ⇒ repeated ciphertext blocks

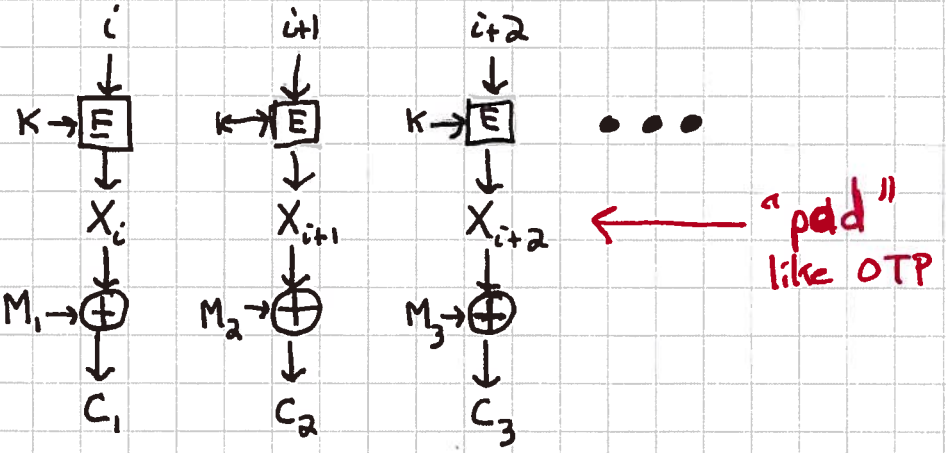
ECB really only good for encrypting random data (e.g. keys)

Ciphertext stealing



CTR (Counter mode):

Generate a PR (pse dorandom) sequence by encrypting $i, i+1, \ldots$
XOR with message to obtain ciphertext.



Initial counter value can be transmitted first:

$i, C_1, C_2, \ldots$

Of course, no counter value should be re-used!

Message does not need to be padded to be a multiple of block length

$|C| = |M|$ (not counting initial counter value)