6.857 Rivest
2/18/09 L5.1

Administrivia: next week: Eran Tromer

Outline:

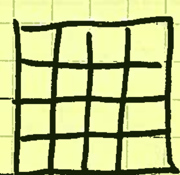
- ☐ AES
- ☐ Modes of operation
  - ☐ ECB
  - ☐ CTR
  - ☐ CBC
  - ☐ cipher feedback
  - ☐ IND-CCA model
  - ☐ UFE
- ☐ MAC's

AES (Advanced Encryption Standard)

- replaced DES (Data encryption std, adopted 1976, 56-bit key)
- "contest" 1997 – 1999, 15 entries: RC6, Mars, twofish, Rijndael, ...
- Winner = Rijndael (by Joan Daemen & Vincent Rijmen; Belgians)
- Specs: 128-bit input/output blocks

  128, 192, or 256-bit key

  10, 12, or 14 rounds internally
- How it works (128-bit key, 10 rounds)
  - byte-oriented spec
  - does some math in $GF(2^8)$
  - view input as 4x4 array of bytes



  - derive 11 'round keys' each 128-bits
  - In each round:

    (1) Add Round Key: bytewise XOR round key into array

    (2) SubBytes: invert (over $GF(2^8)$) each elt of array
    ($0 \Rightarrow 0$)
    apply affine xfrm to bits of each elt
    $ax+b$
    a is 8×8 matrix
    b is vector of size 8 } over $GF(2)$

    (3) ShiftRows:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \Rightarrow \begin{bmatrix} a & b & c & d \\ f & g & h & e \\ k & l & i & j \\ p & m & n & o \end{bmatrix} \begin{pmatrix} \text{shift} \\ 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}$$

(4) <u>Mix Columns</u>

For each column $\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$

treat it as polynomial $a + bx + cx^2 + dx^3$ over $GF(2^8)$

multiply by $3x^3 + x^2 + x + 2$

reduce modulo $x^4 + 1$

(since $\gcd(x^4+1, 3x^3+x^2+x+2) = 1$, this is invertible).

[In last round: no <u>Mix Columns</u>; instead use another <u>Add Round Key</u>]

- Decryption: run backwards, invert each step
- $\exists$ very fast implementations: table lookups & XOR's
  - 16 lookups in four 256 x 32-bit tables
  - 1~~6~~ XOR's (32-bit)

  $\}$ per round

  gigabit/sec in hardware

- <u>Security</u>: # rounds could be larger (?)

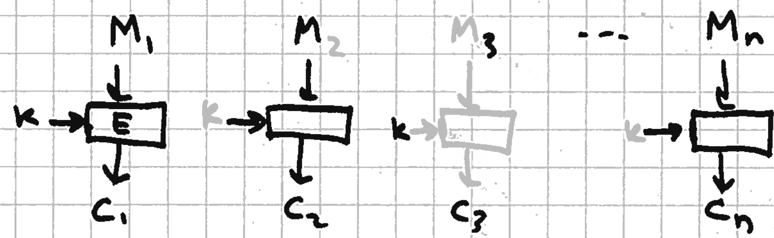
$E_k: \{0,1\}^b \to \{0,1\}^b$ given

Modes of Operation (there are *many*! we'll see a few...)

- How to use given block cipher to encrypt data of arbitrary length?

ECB "Electronic Code Book"

- Divide M into b-bit blocks, encrypt each separately



Patterns in data $\Rightarrow$ patterns in ciphertext (e.g. all-zero blocks...)

- Only really good for encrypting random data (e.g. keys)

To handle data not multiple of b bits:

can "pad" by appending 1 & just enough 0's to give length a multiple of b.

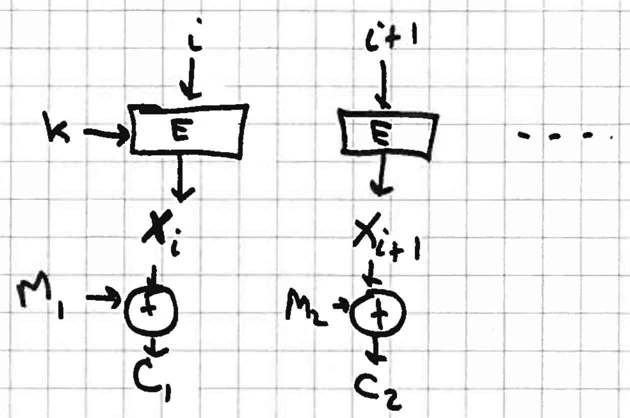
$\underbrace{01101}_{msg}\underbrace{10000\ldots0}_{pad}$ $\qquad \{0,1\}^* \longleftrightarrow \left(\{0,1\}^b\right)^*$ (except 0^b)

padding is invertible.

Pad before encrypting; unpad upon decryption... (treat 0^b as ε ?)

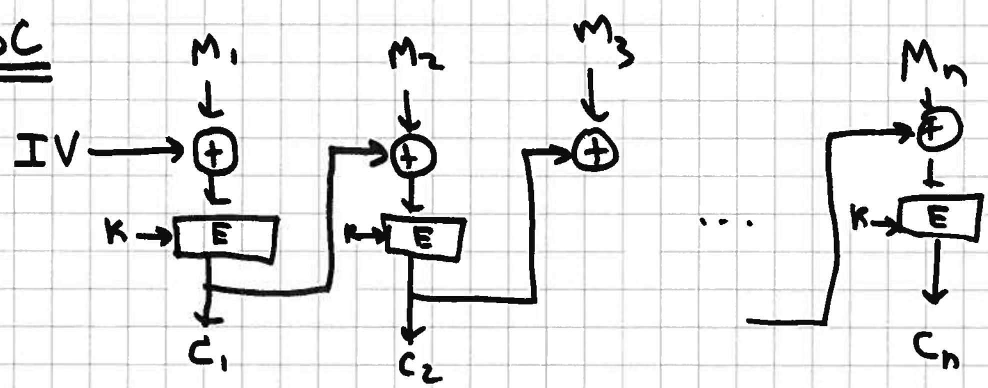
Counter mode

- generate a PR sequence by encrypting $i, i+1, \ldots$



- need way to share i between sender & receiver
  could send $(i, C_1, C_2, \ldots)$

CBC



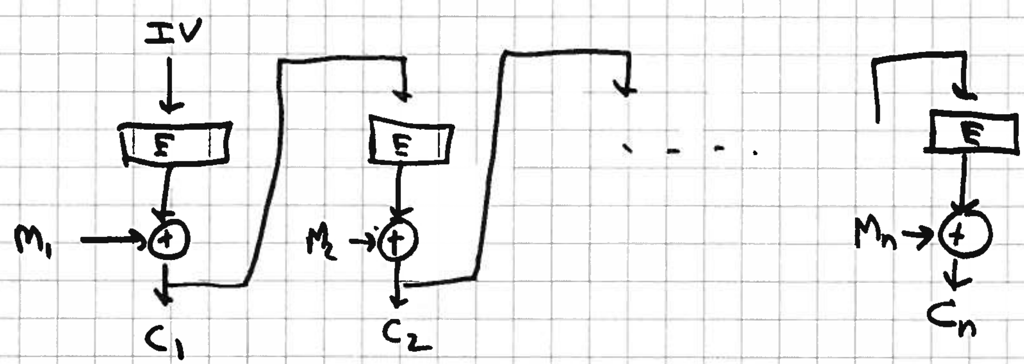
- ~~need to~~ should generate IV randomly, xmit with ciphertext

  $IV, C_1, C_2, \ldots, C_n$

- Decryption easy (& parallelizes! ⇒ little error propagation)
- Last block C_n is "CBC-MAC" for message $M_1, \ldots, M_n$ (with fixed IV)

Cipher feedback

•



Do these modes give us what we want?

What *do* we want?

mode give us $E'_k : \{0,1\}^* \longrightarrow \{0,1\}^*$ (invertible)

based on block cipher $E_k : \{0,1\}^b \rightarrow \{0,1\}^b$

"**IND-CCA**" (Indistinguishability based on chosen-ciphertext attack)

gives us standard to measure encryption modes.

Defined as game with adversary. Mode is secure (IND-CCA) if adversary can win with prob $\leq 1/2 + \varepsilon$ for small ε.

Phase I ("Find")

Adversary outputs two messages m_0, m_1 $(m_0 \neq m_1)(|m_0| = |m_1|)$

(+ state info s)

Adversary can use E_K, D_K freely during this phase

Phase II ("Guess")

$d \xleftarrow{R} \{0,1\}$ random bit chosen by examiner, unknown to adversary

$y \longleftarrow E_K(m_d)$ prepares challenge ciphertext

Adversary now given y, s, access to E_K, and access to D_K (except on y)

Adversary must produce guess $\hat{d}$ for d.

Adversary's advantage is $|(\text{Prob}[\hat{d}=d] - 1/2)|$

Scheme is secure against CCA attack if advantage is negligible.

Fact: to be IND-CCA secure, scheme must be randomized.

(since Adv can encrypt m_0, m_1 himself...)

Previous modes are <u>not</u> IND-CCA secure!
(Decrypting 1^{st} half of long ciphertext $\Rightarrow$
1^{st} half of message...)

Here is <u>sketch</u> of one IND-CCA secure method (UFE - Desai)

M = long input message of length n b-bit blocks $\quad K = (K_1, K_2$

$r \xleftarrow{R} \{0,1\}^b$ $\quad$ b-bit random value

$\text{pad} = P_1, P_2, \ldots, P_n$

where $P_i = E_{K_1}(r+i)$ $\quad$ (CTR mode PRG)

$C = C_1 \cdots C_n$

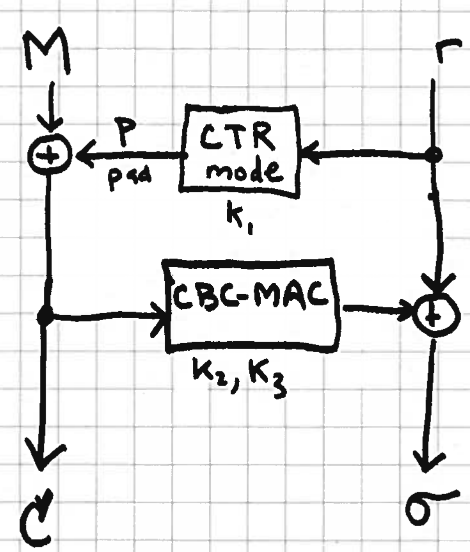
where $C_i = M_i \oplus P_i$

Let $X_0 = 0^b$

$X_i = E_{K_2}(X_{i-1} \oplus C_i)$ $\quad 1 \le i \le n-1$ $\quad$ (CBC mode)

$X_n = E_{K_3}(X_{n-1} \oplus C_n)$ $\quad \sigma = r \oplus X_n$

Output $(C_1, C_2, \ldots, C_n, \sigma)$



UFE
≡
"unbalanced
Feistel
encryption"

[only designed for confidentiality, as that is all IND-CCA cares about...]

MAC's (Message Authention Codes)

Alice — K — Bob — K

$$M \xrightarrow[M,\ MAC_K(M)]{} $$

- Bob recomputes $MAC_K(M)$ & rejects message as not from Alice, or damaged, if he gets different result.
- Integrity, not confidentiality
- can layer on top of encryption (e.g. M = ciphertext for some other msg)
- MAC can be short, e.g. $b = 64$ bits
- Infeasible for an adversary to create new valid pair

$$M,\ MAC_K(M)$$

(that Bob will accept), even after seeing many previous valid pairs (even for messages of his choice).

- CBC-MAC (use CBC mode, return last block)
- HMAC $\approx$ hash$(K_1 \,\|\, \text{hash}(K_2 \,\|\, M))$

Combined modes

For confidentiality and integrity, you can encrypt then MAC, or use one of the combined modes of operation (CCM, EAX, OCB modes...)