Lecture 3

Note Title 1/11/2006

Bellare & Rogaway notes
Ch. 3-5
John Benaloh 2002 Course Sides
PMP Current Courses

Symmetric Encryption

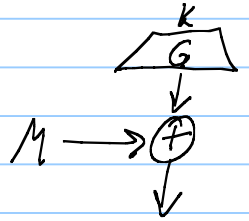
Stream Cipher – Suppose we had pseudorandom number generator

$$G: \{0,1\}^k \to \{0,1\}^m \qquad m >> k$$

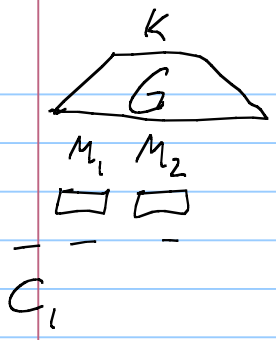
Key Generation $k \xleftarrow{R} \{0,1\}^k$ ~~$M \to \{0,1\}^m$~~

$$C = E_k(M) \leftarrow G(k) \oplus M$$

$$M = D_k(C) \leftarrow C \oplus G(k)$$



one-time pad with expanded key



Block Ciphers

A n-bit block cipher is a function: n is fixed

$$E = \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$$

s.t. for each $k \in \mathcal{K}$

E_k is a permutation of $\{0,1\}^n$

$$E_k^{-1} : \{0,1\}^n \to \{0,1\}^n$$

ex. one-time pad $k = n$

Given $C_1 = E_k(M_1)$, $C_2 = E_k(M_2)$ get relations between M_1, M_2 or $M_1 \oplus M_2$

Ideally, given $M_1, M_2, \ldots M_q$

k chosen at random

Want no obvious relationships between

$$E_k(M_1), \ldots E_k(M_q)$$

$$C_1 \quad \text{---} \quad C_q$$

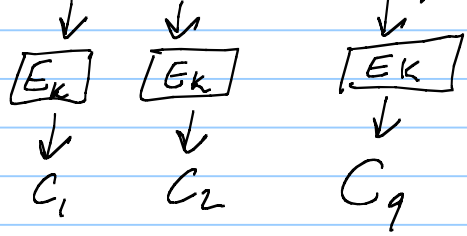
Ciphertext only attacks

We want to encrypt messages of length $\geq n$

Ways to do it: Break M into n-bit chunks

Generally Bad

$M = M_1, M_2, \ldots M_q$



$C_1 \quad C_2 \quad C_q$

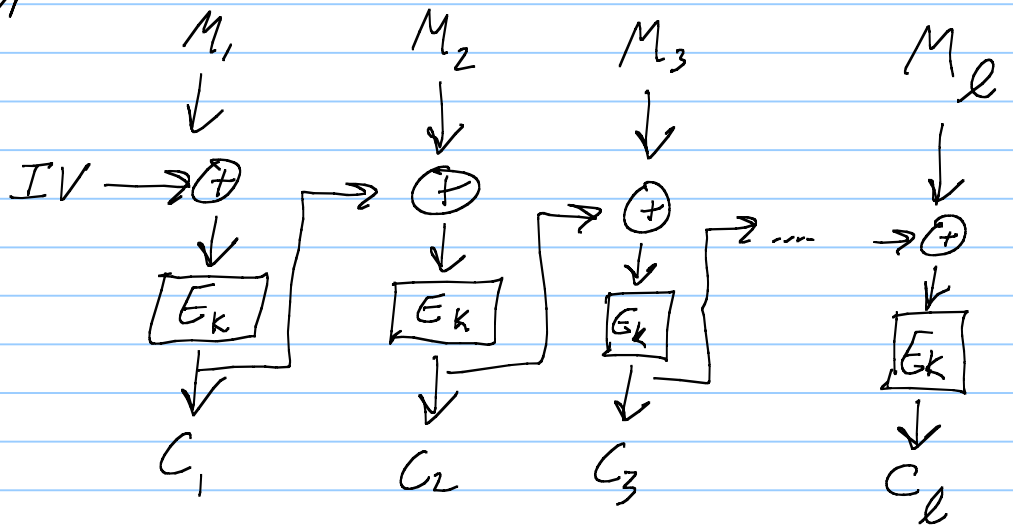
ECB (Electronic Code Book) Method

Cipher Block Chaining (CBC)

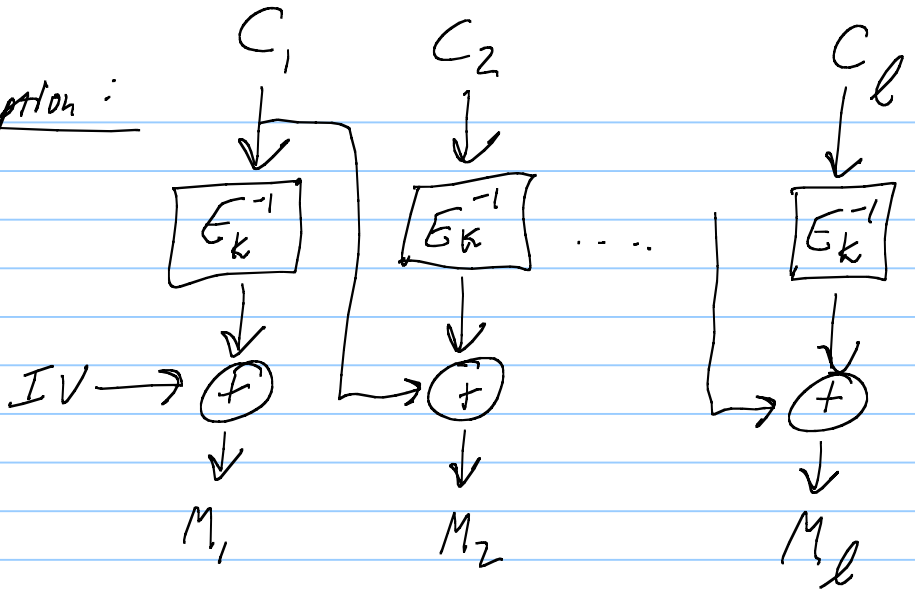
new: Initialization Vector (IV)

- public
- random
- must change each time we start a stream

Encryption



Decryption:



larger than n-bit messages

Suppose message length is not a multiple of n

Ex. $n = 64 \rightarrow 8$ bytes

last block has i bytes missing at end.

$i = 0, 1, 2, \ldots 7$

append i copies of the ASCII for i

04 04 04 04

For $i = 0$ pretend $i = 8$

08 08 08 08 08 08 08 08

Must have one byte available in last block

Desire data for E_k:

looks like a random function

{ but no collisions } { permutations }

Feistel Cipher:

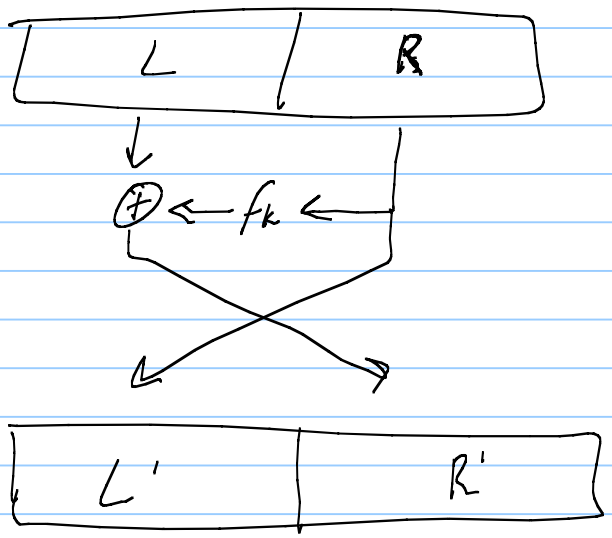
Suppose we have $\{f_k\}_{k \in K}$ which looks random

We want to build permutations to these

$$f_k = \{0,1\}^k \to \{0,1\}^n$$

E_k : $n = 2m$

Feistel Round $n = 2m$ bits



$L' \leftarrow R$

$R' \leftarrow L \oplus F_k(R)$

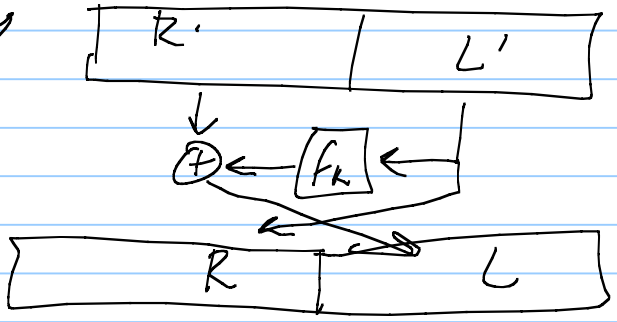
---

inverse

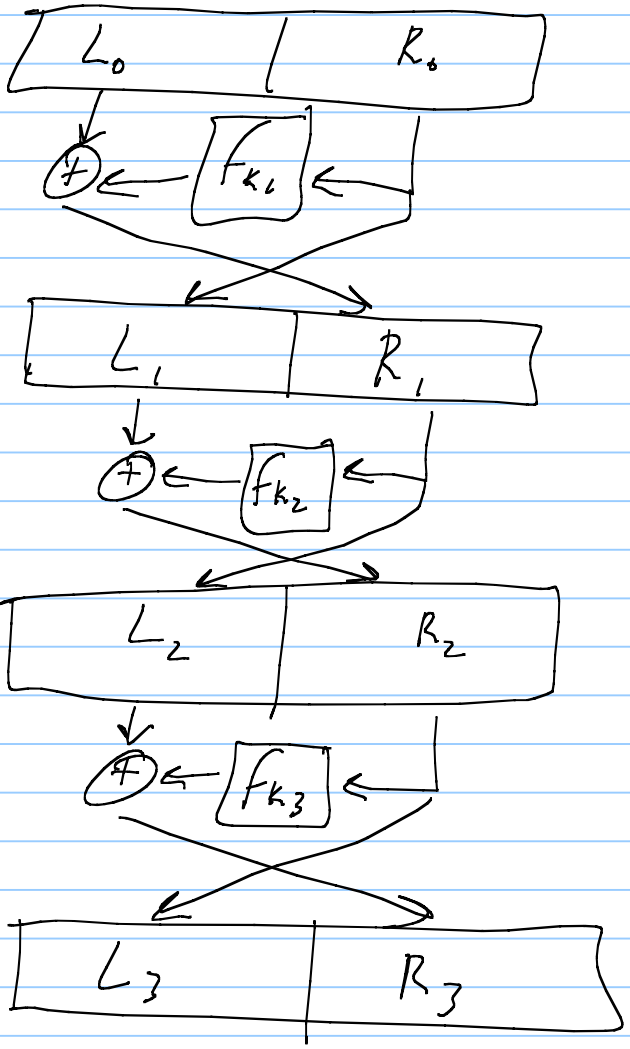
$R \leftarrow L'$

$L \leftarrow R' \oplus F_k(L')$

inverse



L_0 is not encrypted well, so repeat with different keys



if f_k is really good, then 3 times is enough!

Data Encryption Standard

DES — n = 64-bit block cipher

56 bit key K

expanded into 16 48 bit keys

$k_1, \ldots, k_{16}$

16 Feistel Rounds

$$f = \{0,1\}^{48} \times \{0,1\}^{32} \to \{0,1\}^{32}$$

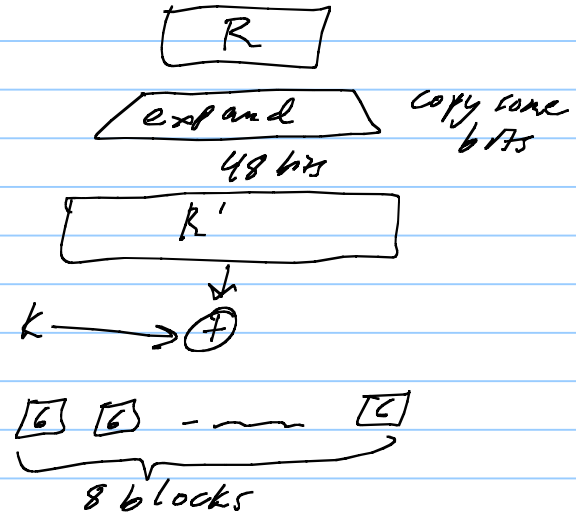
permutation of bits to start and ciphertext at end

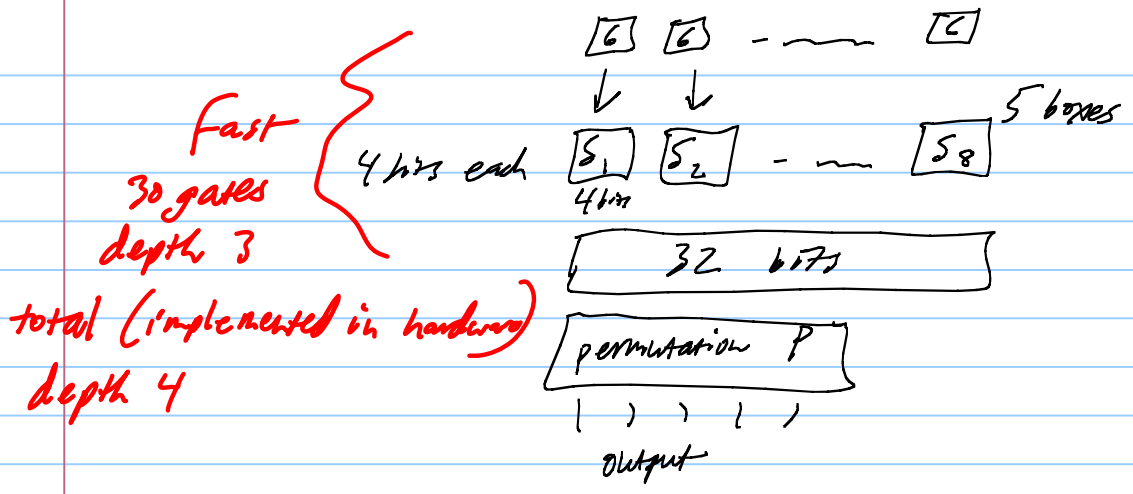
What does f_k do?

48 bits

32-bit R

$f(k, R) \to R'$




Fast
30 gates
depth 3
total (implemented in hardware)
depth 4
6
6
6
4 bits each
S1
S2
S8
4 bits
5 boxes
32 bits
permutation P
output