Today in Cryptography (5830)

Modes of operation for block ciphers
Padding oracle attacks against CBC mode

Recap: Block ciphers, feistel & length preserving encryption

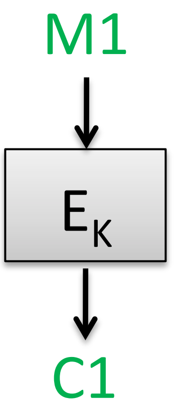
Block cipher is a map $E : \{0,1\}^k \times \{0,1\}^n \dashrightarrow \{0,1\}^n$

Each key K defines permutation $E_K : \{0,1\}^n \dashrightarrow \{0,1\}^n$

Permutation: 1-1, onto

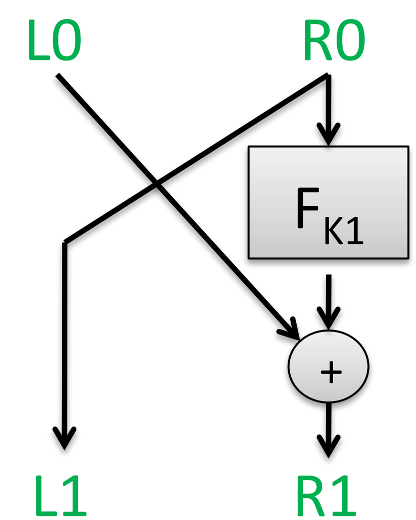
Block ciphers must be efficient

Should behave like random permutation

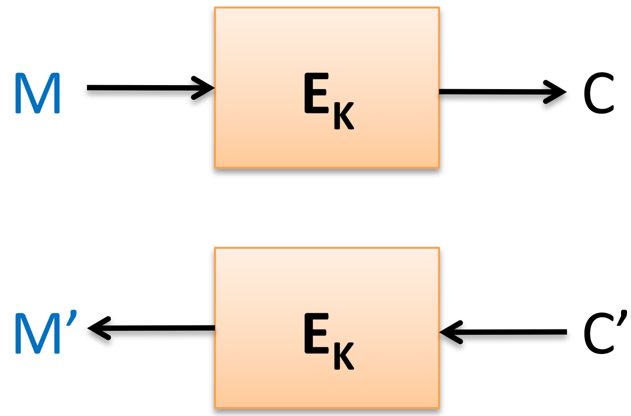


Feistel networks turn function into permutation.

- Used in DES
- Useful for building length-preserving encryption on arbitrary length messages



Security problems with length-preserving encryption?



Looks like a random string (subject to permutivity)

Looks like a random string (subject to permutivity)

Any C' ≠ C

But determinism has problems:

	Plaintext	Ciphertext
Jane Doe	1343-1321-1231-2310	1049-9310-3210-4732
Thomas Ristenpart	9541-3156-1320-2139	7180-4315-4839-0142
John Jones	2321-4232-1340-1410	5731-8943-1483-9015
Eve Judas	1343-1321-1231-2310	1049-9310-3210-4732

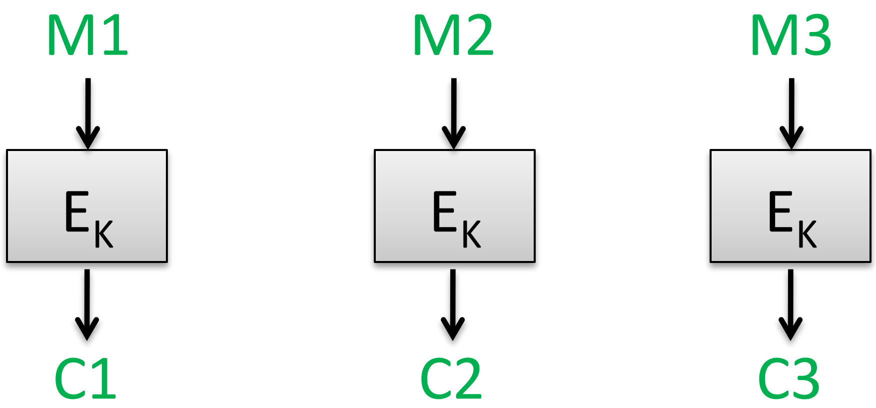
Length-extending encryption security

- Not a bit of information about plaintext leaked
  - Equality of plaintexts hidden
  - Even in case of active attacks
    - Padding oracles we will see later
- Eventually: authenticity of messages as well
  - Decryption should reject modified ciphertexts

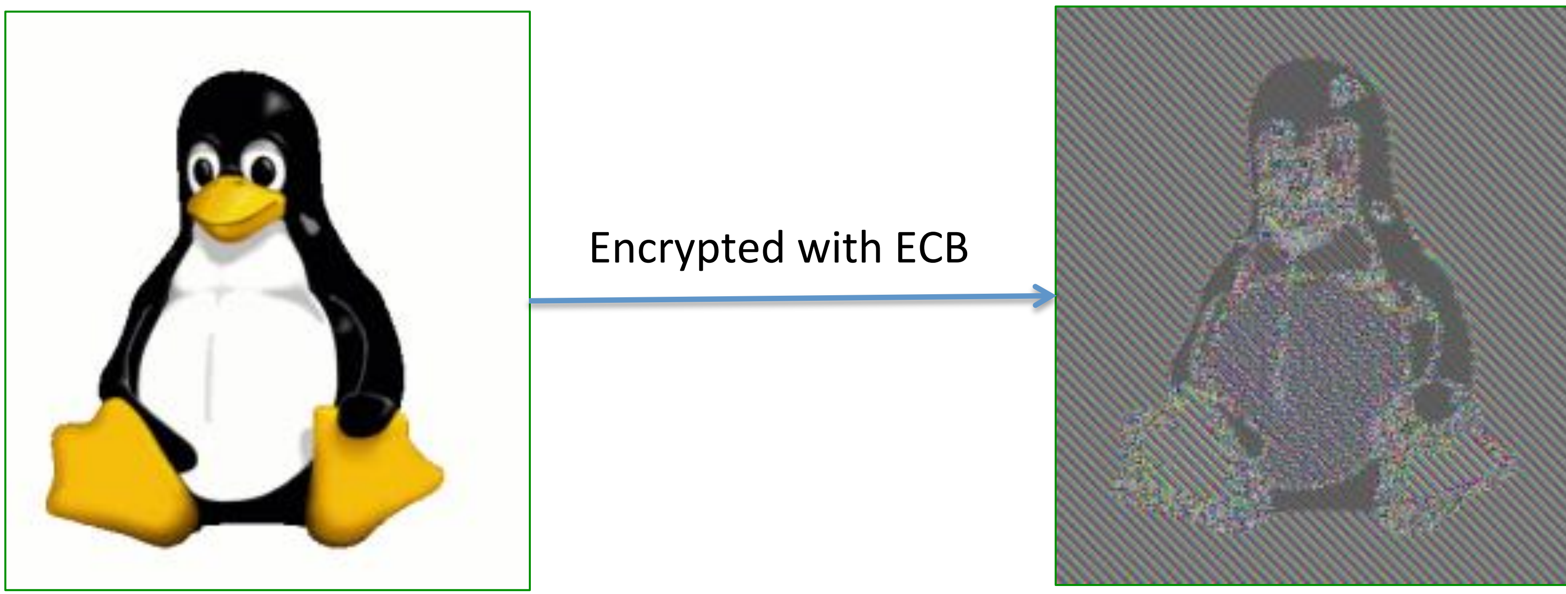
Block cipher modes of operation

How can we build an encryption scheme for arbitrary message spaces out of a block cipher?

Electronic codebook (ECB) mode
Pad message M to M1,M2,M3,... where each block Mi is n bits
Then:



ECB mode is a more complicated looking substitution cipher



Images courtesy of
http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

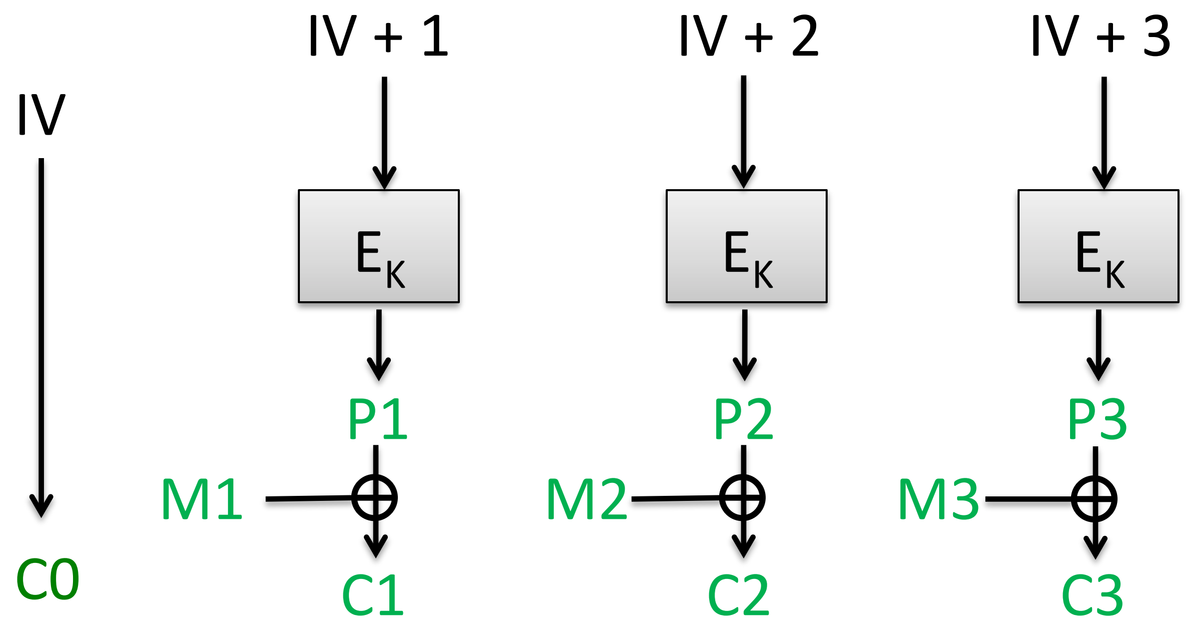
CTR mode encryption using block cipher

Counter mode (CTR)

Pad message M to M1,M2,M3,... where each is n bits except last

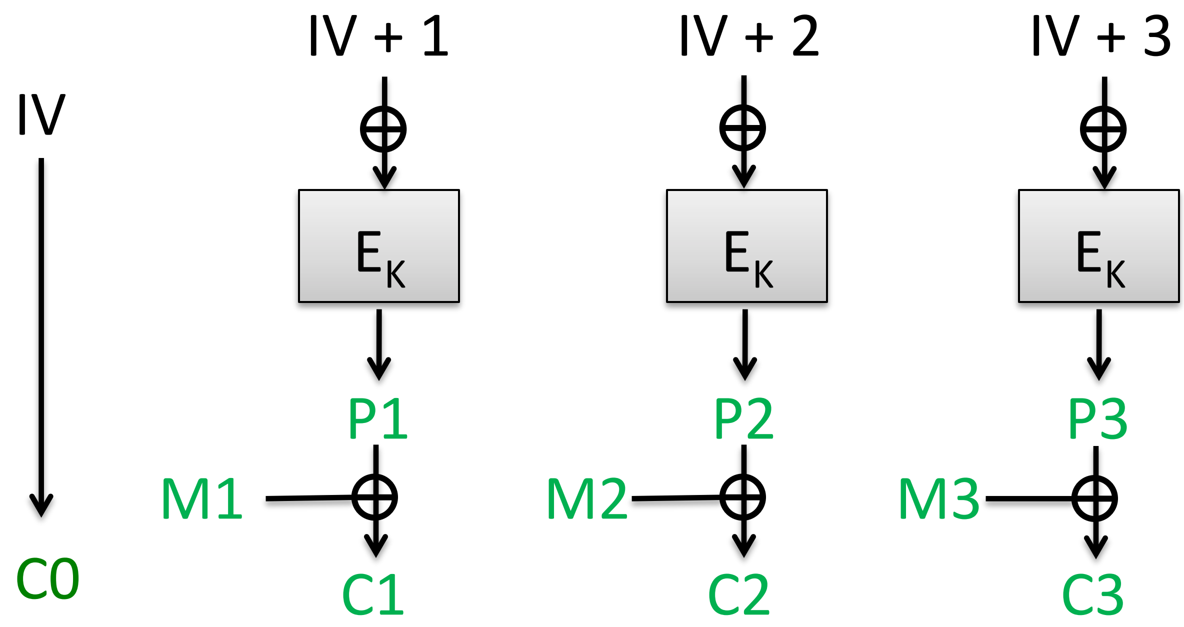
Choose random n-bit string IV

Then:



Maybe use less than full n bits of P3

How do we decrypt?



Can attacker learn K from just C0,C1,C2,C3?

Implies attacker can break E, i.e. recover block cipher key

Can attacker learn M = M1,M2,M3 from C0,C1,C2,C3?

Implies attacker can invert the block cipher without knowing K

Can attacker learn one bit of M from C0,C1,C2,C3?

Implies attacker can break PRF security of E

Passive adversaries cannot learn anything about messages

Malleability example: Encrypted cookies

GET /index.html HTTP/1.1

HTTP/1.1 200 OK
Content-type: text/html
Set-cookie: sessionToken=ab35h013490...

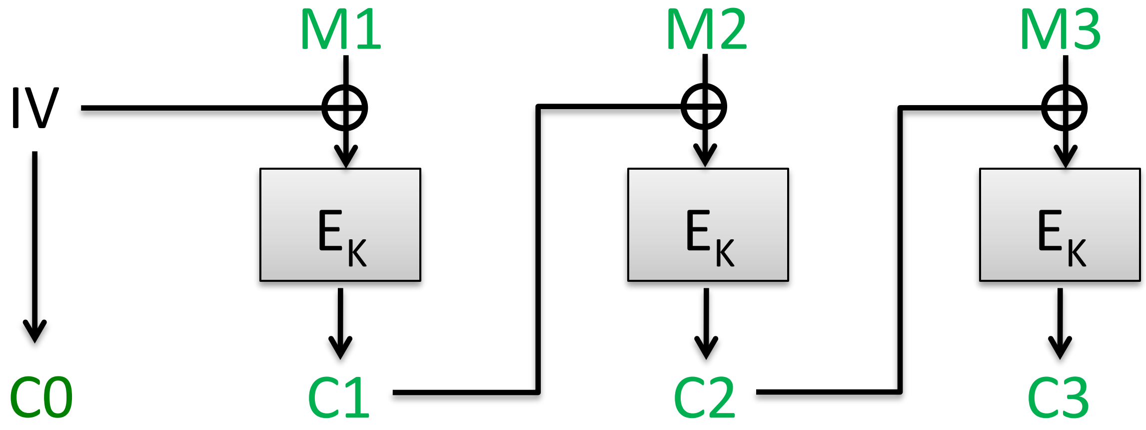
GET /info.html HTTP/1.1
Cookie: sessionToken=ab35h013490…

abc35h013490… = CTR-Mode(K, “admin=0”)

Malicious client can simply flip a few bits to change admin=1

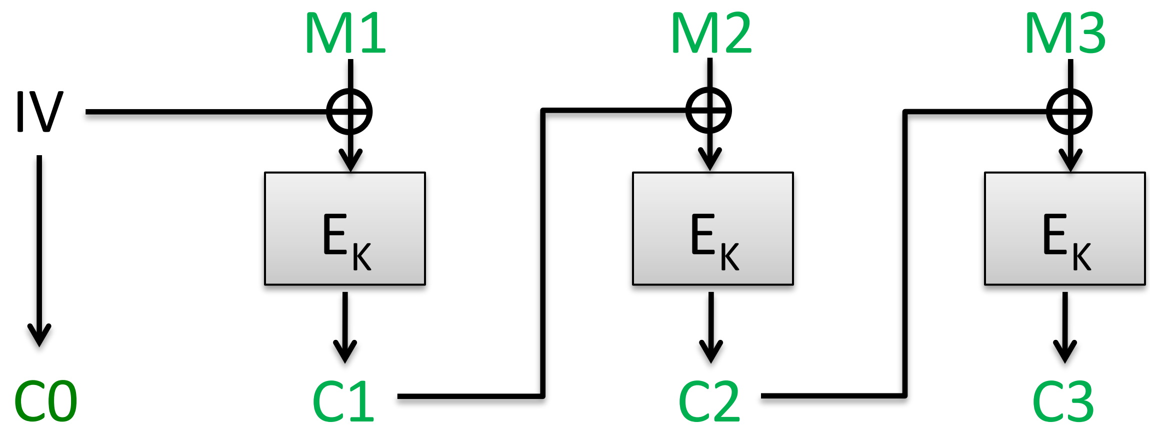
CBC mode

Ciphertext block chaining (CBC)
Pad message M to M1,M2,M3,... where each block Mi is n bits
Choose random n-bit string IV
Then:



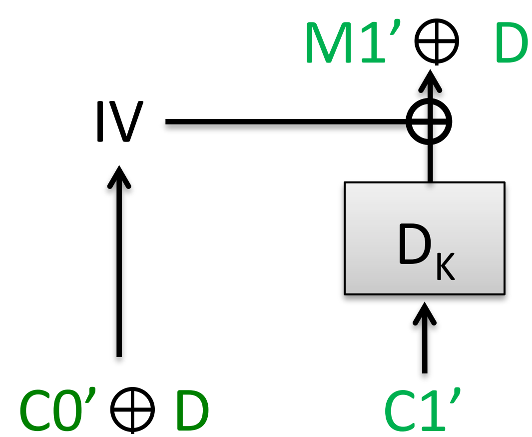
How do we decrypt?

CBC mode has similar "malleability" issues



How do we change bits of M1 received by server??

For any D:

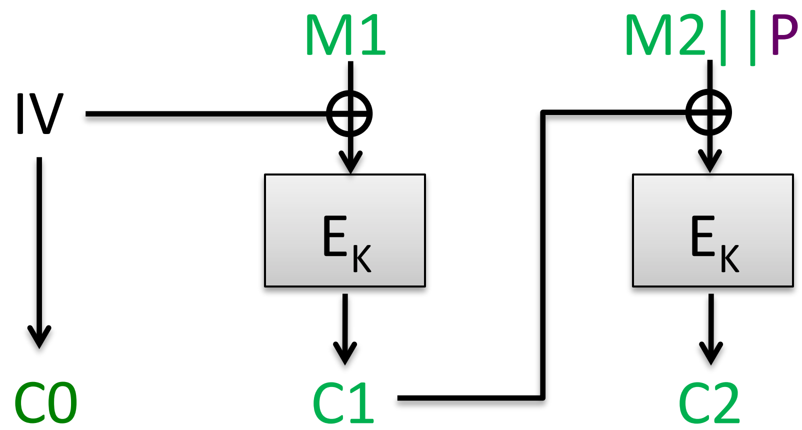


Padding for CBC mode

- CBC mode handles messages with length a multiple of n bits
- We use padding to make it work for arbitrary encryption schemes

- Padding checks often give rise to padding oracle attacks

Simple situation: pad by 1 byte



Assume that M1||M2 has length 2n-8 bits

P is one byte of padding that must equal 0x00

C0 , C1 , C2 →

← ok

Adversary obtains Ciphertext C0,C1,C2

C0, C1⊕1 , C2 →

← error

<u>Dec(K, C')</u>

M1'||M2'||P' = CBC-Dec(K,C')

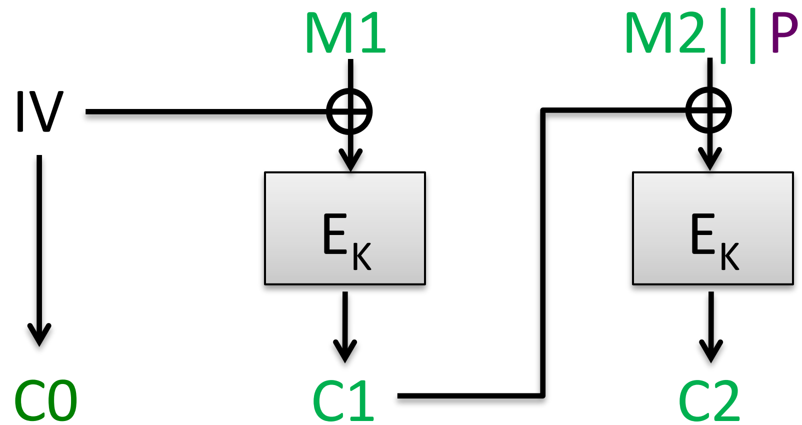
If P' ≠ 0x00 then

  Return error

Else

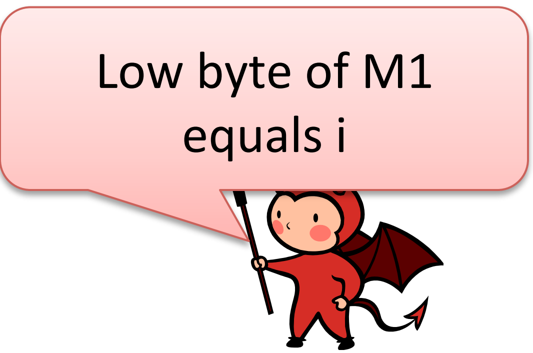
  Return ok

Simple situation: pad by 1 byte



Assume that M1||M2 has length 2n-8 bits

P is one byte of padding that must equal 0x00



R, C0 , C1 →
← error
R , C0 ⊕ 1 , C1 →
← error
R , C0 ⊕ 2 , C1 →
← error
…
R , C0 ⊕ i , C1 →
← ok

Adversary obtains ciphertext C = C0,C1,C2
Let R be arbitrary n bits

<u>Dec(K, C')</u>
M1'||M2'||P' = CBC-Dec(K,C')
If P' ≠ 0x00 then
    Return error
Else
    Return ok

PKCS #7 Padding

$$\text{PKCS\#7-Pad}(M) \quad = \quad M \,||\, \underbrace{P \,||\, \ldots \,||\, P}_{\text{P repetitions of byte encoding number of bytes padded}}$$

Possible paddings:

01
02 02
03 03 03
04 04 04 04
…
FF FF FF FF … FF

For block length of 16 bytes, never need more than 16 bytes of padding (10 10 … 10)

Decryption

(assuming at most one block of padding)

```
Dec( K, C )
M1 || … || Mn = CBC-Dec(K,C)
P = RemoveLastByte(Mn)
while i < int(P):
      P' = RemoveLastByte(Mn)
      If P' != P then
            Return error
Return ok
```

PKCS #7 padding oracles

Low byte of M1 equals i xor 01

Adversary obtains ciphertext C = C0,C1,C2

Let R be arbitrary n bits

R, C0 , C1 →

← error

R , C0 ⊕ 1 , C1 →

← error

R , C0 ⊕ 2 , C1 →

← error

…

R , C0 ⊕ i , C1 →

← ok

<u>Dec(K, C)</u>

```
M1 || … || Mn = CBC-Dec(K,C)
P = RemoveLastByte(Mn)
while i < int(P):
    P' = RemoveLastByte(Mn)
    If P' != P then
        Return error
Return ok
```

PKCS #7 padding oracles

Second lowest byte of M1 equals i xor 02

R, C0 , C1

error

R , C0 ⊕ 1||j , C1

error

R , C0 ⊕ 2||j , C1

error

…

R , C0 ⊕ i||j , C1

ok

Adversary obtains ciphertext C = C0,C1,C2
Let R be arbitrary n bits

Dec(K, C)
M1 || … || Mn = CBC-Dec(K,C)
P = RemoveLastByte(Mn)
while i < int(P):
    P' = RemoveLastByte(Mn)
    If P' != P then
        Return error
Return ok

Set j = i

Chosen ciphertext attacks against CBC

Attack	Description	Year
Vaudenay	10’s of chosen ciphertexts, recovers message bits from a ciphertext. Called “padding oracle attack”	2001
Canvel et al.	Shows how to use Vaudenay’s ideas against TLS	2003
Degabriele, Paterson	Breaks IPsec encryption-only mode	2006
Albrecht et al.	Plaintext recovery against SSH	2009
Duong, Rizzo	Breaking ASP.net encryption	2011
Jager, Somorovsky	XML encryption standard	2011
Duong, Rizzo	“Beast” attacks against TLS	2011

None of these modes are secure for encryption

- ECB is obviously insecure
- CTR mode and CBC mode fail in presence of active attacks
  - Cookie example
  - Padding oracle attacks
- Next lecture: adding authentication mechanisms