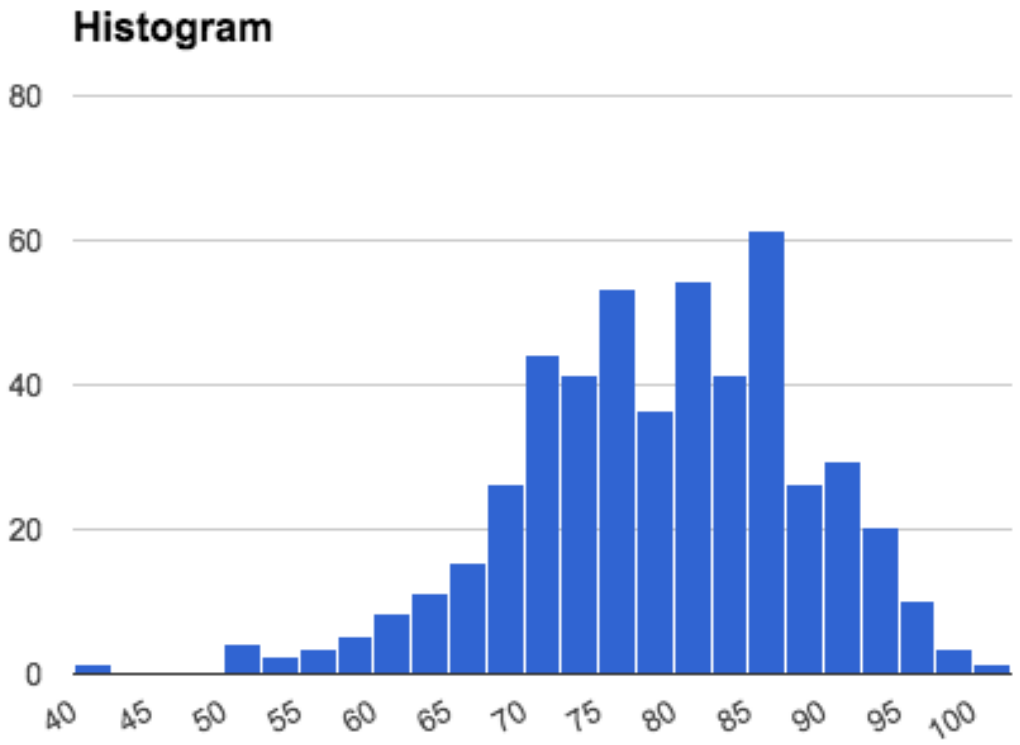
Block ciphers

CS 161: Computer Security

Prof. Raluca Ada Popa

February 26, 2016

Announcements



MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
40.0	79.5	100.0	78.85	9.41

Last time

- Syntax of encryption: Keygen, Enc, Dec
- Security definition for known plaintext attack:
  - attacker provides two messages m0, m1
  - attacker receives one encrypted
  - must guess which was encrypted
- Recall one-time pad:
  - provides strong security, but can only be used once

Today: block ciphers

- Building blocks for symmetric-key encryption schemes that can be **reused**

Block cipher

A function $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$. Once we fix the key K, we get

$E_K : \{0,1\}^n \rightarrow \{0,1\}^n$ defined by $E_K(M) = E(K,M)$.

Three properties:

- Correctness:
  - $E_K(M)$ is a permutation (bijective function)
- Efficiency
- Security

Efficiency

- Can compute $E_K(M)$ efficiently (polynomial-time)
- Can compute $D_K(C)$ efficiently, the inverse of E_K

$$D_K(E_K(M)) = M$$

Security

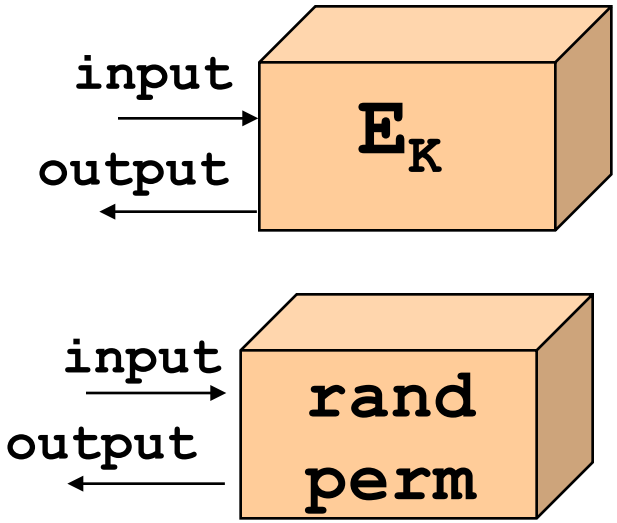
For an unknown key K, E_K “behaves” like a random permutation

For all polynomial-time attackers, for a randomly chosen key K, the attacker **cannot distinguish** E_K from a random permutation

Block cipher: security game

- Attacker is given two boxes, one for E_K and one for a random permutation
- Attacker does not know which is which
- Attacker can give inputs to each box, look at the output
- Attacker must guess which is E_K

??? Which is E_K???



Security game

For all polynomial-time attackers,

Pr[attacker wins game] <= ½+negl

Example block cipher: AES (Advanced Encryption Standard)

- Joan Daemen & Vincent Rijmen, 1997
- Block size 128 bits
- Key can be 128, 192, or 256 bits (today use 256)
- You don't need to understand how it works for this class
  - Just to get a sense of it: basically it has multiple rounds during which it combines bits of plaintext with bits of the key, substitution steps where bits are replaced with other bits from a lookup table, bits are shifted, bits are mixed, etc.
- Not provably secure, but was not broken so far, so people **assume it is a secure block cipher**

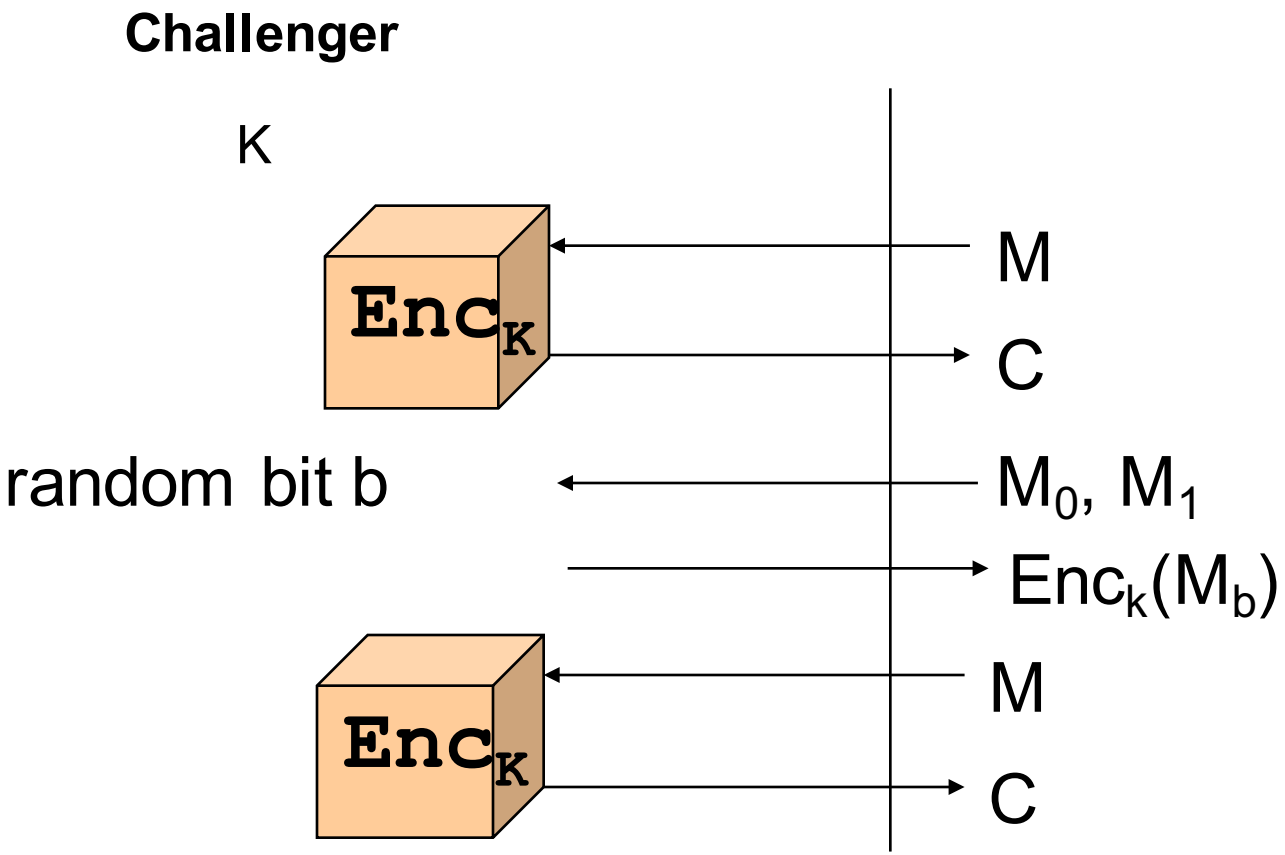
Block ciphers as encryption

How to use them as encryption?

First idea:

- $Enc(K, M) = E_K(M)$
- $Dec(K, C) = D_K(C)$

Desired security: indistinguishability under chosen plaintext attack (IND-CPA)



Here is my guess: b'

IND-CPA

An encryption scheme is IND-CPA if

for all polynomial-time adversaries

$$\Pr[\text{Adv wins game}] \leq \tfrac{1}{2} + \text{negligible}$$

Note that IND-CPA requires that the encryption scheme is randomized

(An encryption scheme is deterministic if it outputs the same ciphertext when encrypting the same plaintext; a randomized scheme does not have this property)

Difference from known-plaintext attack from last time

- The extra queries to Enc_K
- The attacker gets to see encryptions for ciphertexts of its choice
- Why is IND-CPA a stronger security?
  - The attacker is given more capabilities so the IND-CPA scheme resists a more powerful attacker

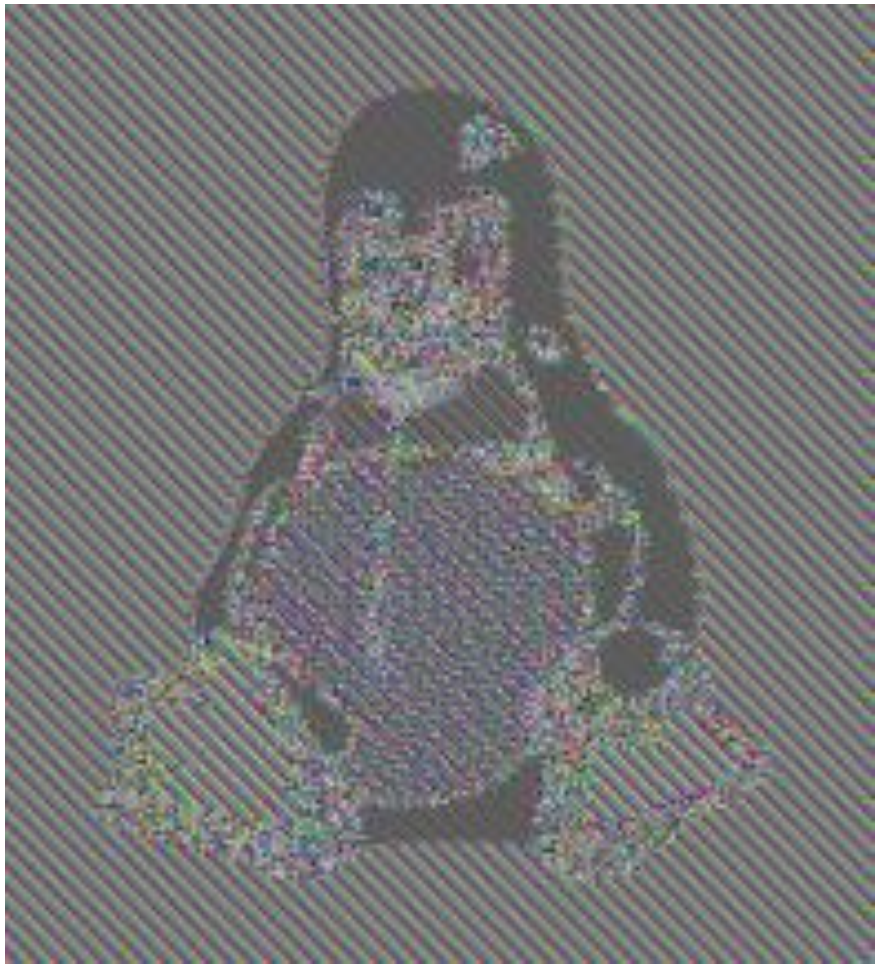
Are block ciphers IND-CPA?

Recall: $E_K : \{0,1\}^n \rightarrow \{0,1\}^n$ is a permutation (bijective)

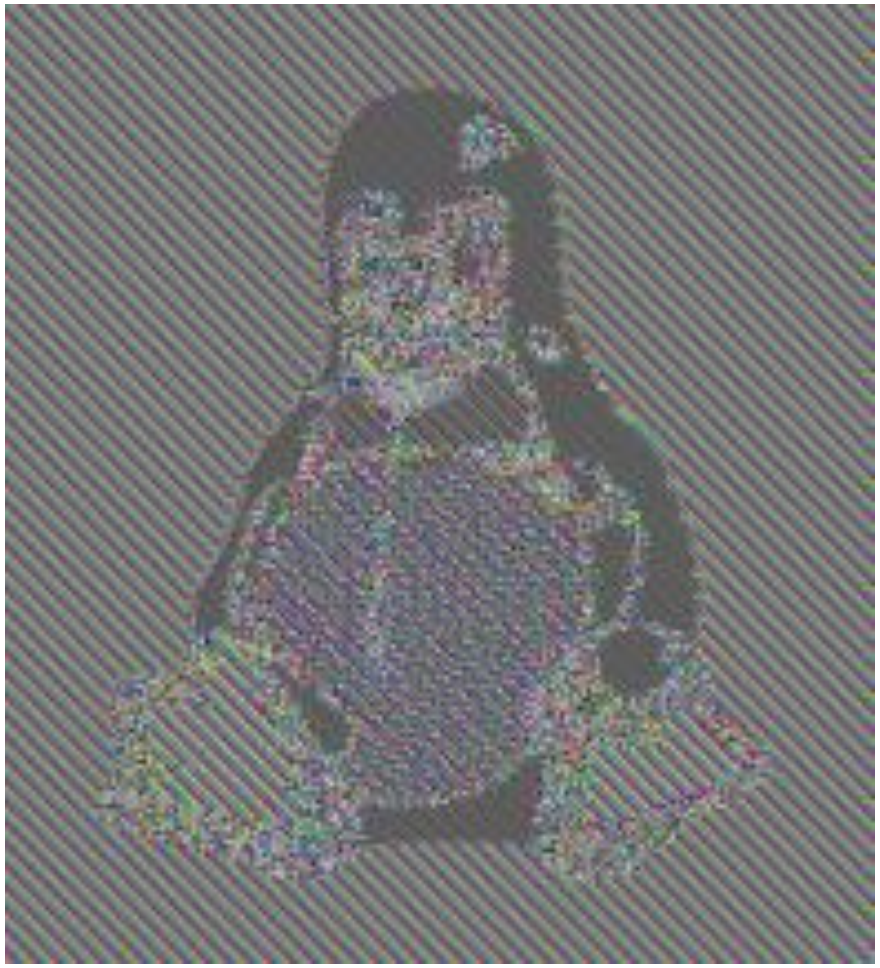
Are block ciphers secure under chosen-plaintext attack?

- No, because they are deterministic
- Here is an attacker that wins the IND-CPA game:
  - Adv asks for encryptions of “bread”, receives C_{br}
  - Then, Adv provides (M_0 = bread, M_1 = honey)
  - Adv receives C
  - If $C=C_{br}$, Adv says bit was 0 (for “bread”), else Adv says says bit was 1 (for “honey”)
  - Chance of winning is 1

Original image



Eack block encrypted with a block cipher



Later (identical) message again encrypted

Another insufficiency of block ciphers:

- Can only encrypt a block!
- Blocks have a certain size n so the plaintext can only be as long
- What do we do for longer strings?

Modes of operation

Chain block ciphers **in certain modes of operation**

- Certain output from one block feeds into next block

Need some **initial randomness IV** **(initialization vector)**

Why? To prevent the encryption scheme from being deterministic

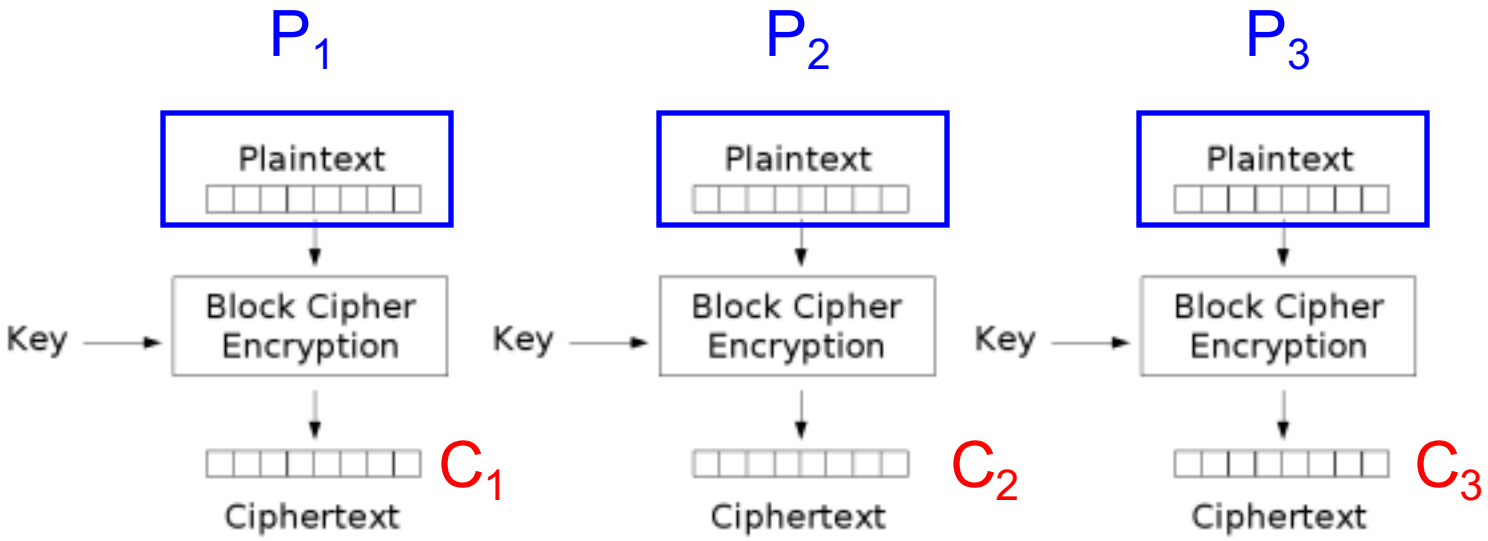
How would you chain a block cipher to encrypt long strings?

Electronic Code Book (ECB)

- Split message in blocks P_1, P_2, …
- Each block is a value which is substituted, like a codebook
- Each block is encoded independently of the other blocks

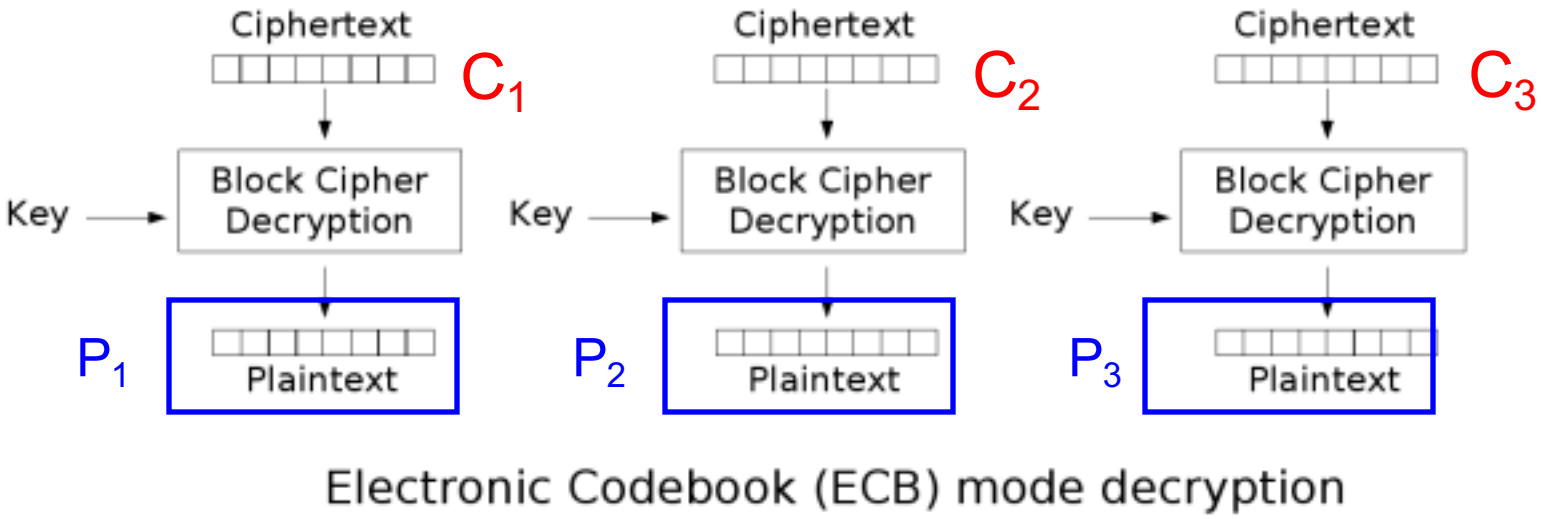
$$C_i = EK(Pi)$$

Encryption



Electronic Codebook (ECB) mode encryption

Decryption

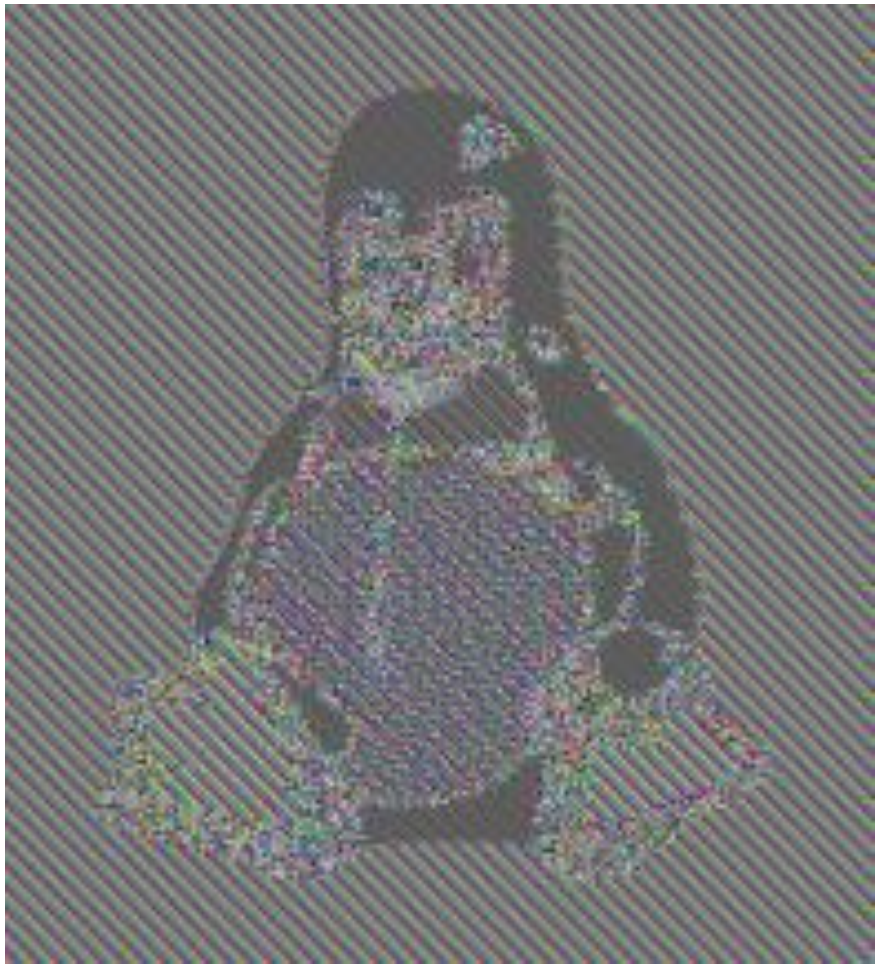


Electronic Codebook (ECB) mode decryption

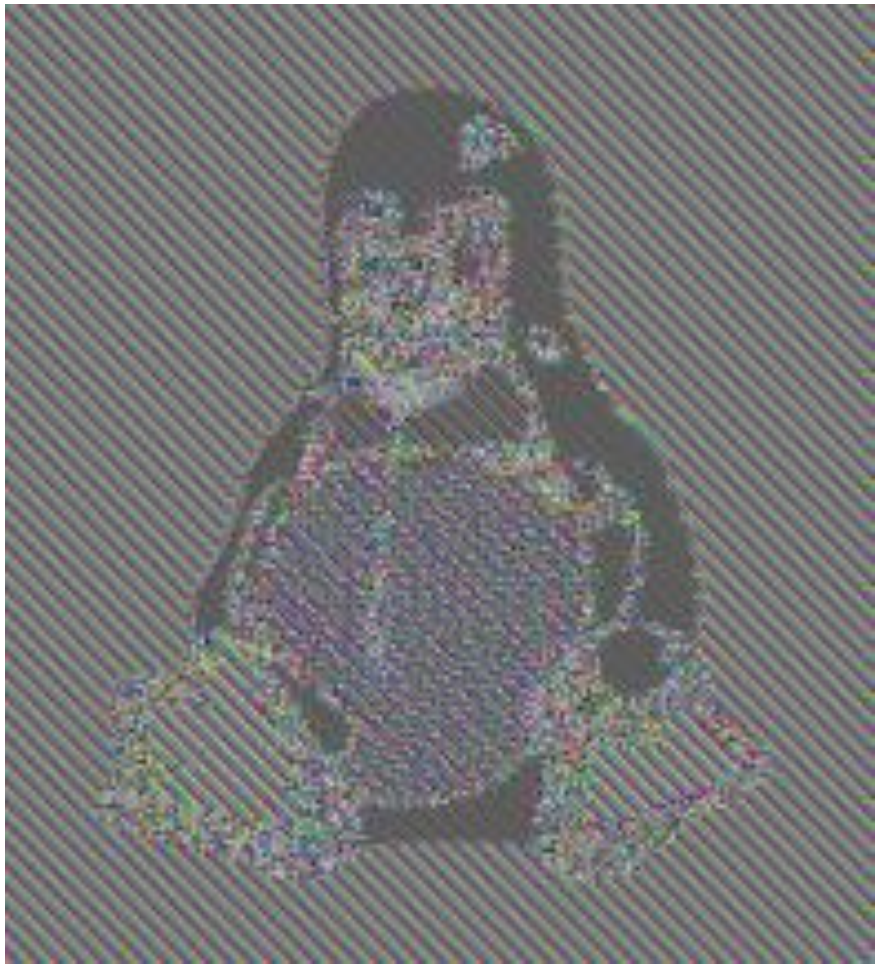
What is the problem with ECB?

Deterministic per block

Original image



Encrypted with ECB

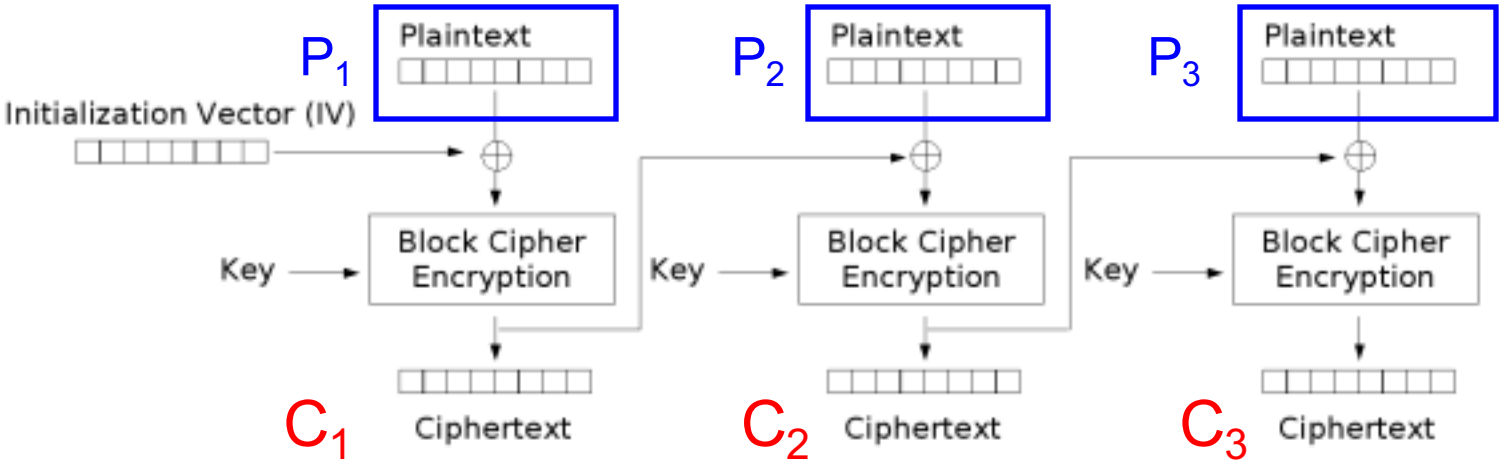


Later (identical) message again encrypted with ECB

CBC: Encryption

Enc(K, plaintext):

- If n is the block size of the block cipher, split the plaintext in blocks of size n: P_1, P_2, P_3,..
- Choose a random IV
- Now compute this:



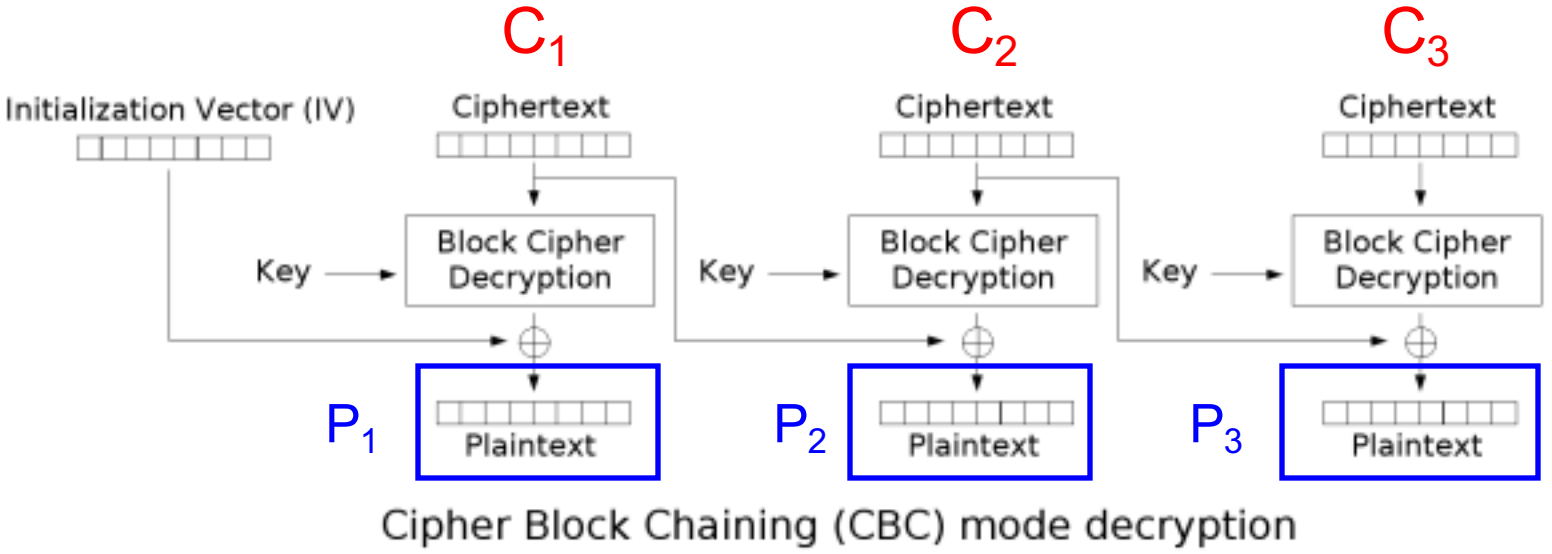
Cipher Block Chaining (CBC) mode encryption

- The final ciphertext is (IV, C_1, C_2, C_3)

CBC: Decryption

Dec(K, ciphertext):

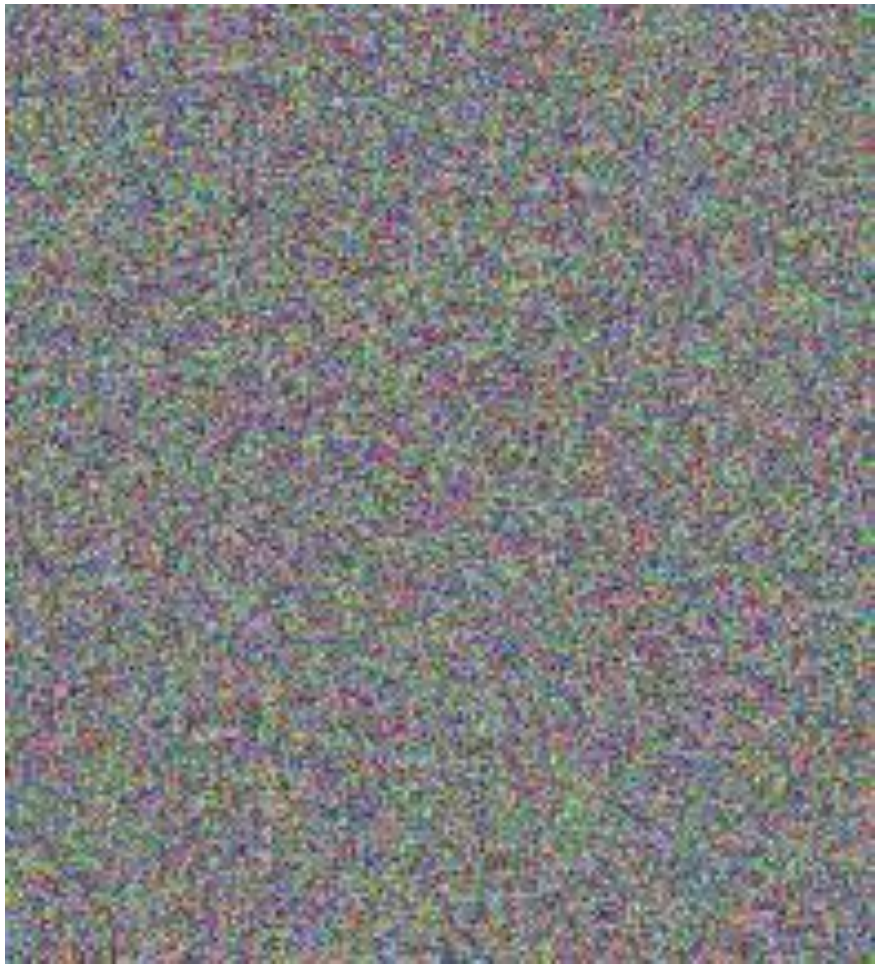
- Take IV out of the ciphertext
- If n is the block size of the block cipher, split the ciphertext in blocks of size n: C_1, C_2, C_3,..
- Now compute this:



Cipher Block Chaining (CBC) mode decryption

- Output the plaintext as the concatenation of P_1, P_2, P_3, ...

Original image



Encrypted with CBC

CBC

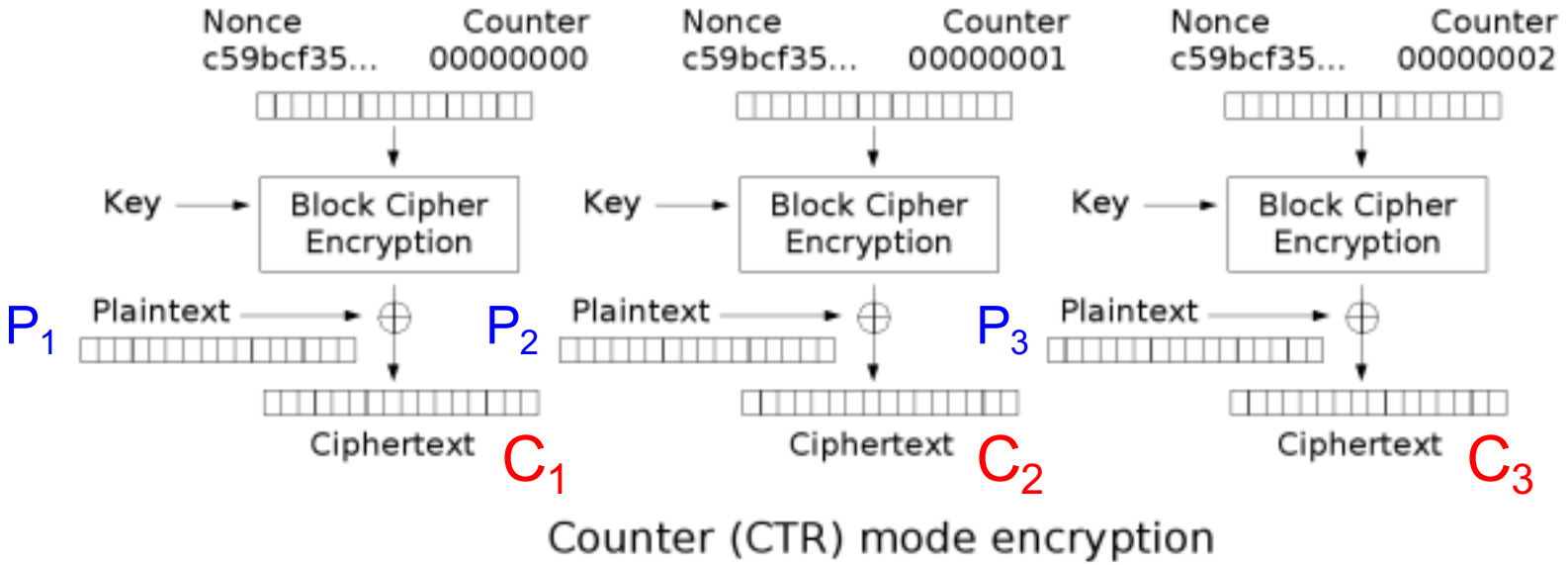
Popular, still widely used

Caveat: sequential encryption, hard to parallelize

CTR mode gaining popularity

CTR: Encryption

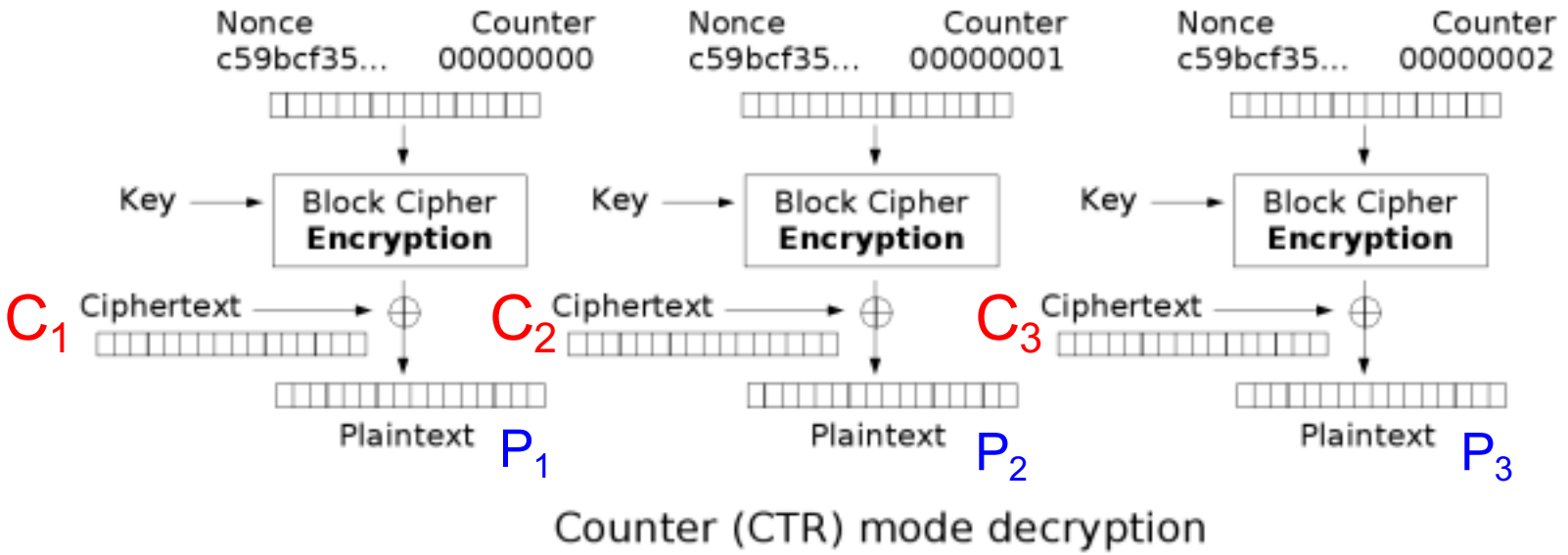
(Nonce = Same as IV)



Important that nonce does not repeat across different encryptions

Choose at random

CTR: Decryption



Counter (CTR) mode decryption

Note, CTR decryption uses block cipher's *encryption*, not decryption

CBC vs CTR

Security: If no reuse of nonce**, both are IND-CPA.**

If you ever reuse the same nonce, CTR leaks more information than CBC. Consider two plaintexts with blocks P1, P2, P3 and P1', P2', P3'. Consider P1=P1', P2 not equal to P2', and P3=P3'. When using the same IV for encrypting these two plaintexts, the attacker can see that P1=P1' for both, and that P3=P3' for CTR, but not for CBC.

Speed: Both modes require the same amount of computation, but CTR is parallelizable

Stream ciphers

Stream ciphers

- Another way to construct encryption schemes
- Similar in spirit to one-time pad: it XORs the plaintext with some random bits
- But random bits are not the key (as in one-time pad) but are output of a pseudorandom generator PRG

Pseudorandom Generator (PRG)

- Given a seed, it outputs a sequence of random bits

  PRG(seed) -> random bits

- It can output arbitrarily many random bits

PRG security

- Can PRG(K) be truly random?

No. Consider key length k. Have 2^k possible initial states of PRG. Deterministic from then on.

- A secure PRG suffices to “look” random to an attacker (no attacker can distinguish it from a random sequence)

Stream cipher

Enc(K, M):

- Choose a random value IV
- Enc(K,M) = PRG(K, IV) XOR M

Can encrypt any message length because PRG can produce any number of random bits

Example of PRG: using block cipher in CTR mode

If you want m random bits, and a block cipher with E_k has n bits, apply the block cipher ceil(m/n) times and concatenate the result:

PRG(K, IV) = E_k(IV, 1), E_k(IV, 2), E_k(IV, 3) … E_k(IV, ceil(m/n))

Example of stream cipher: using block cipher in CTR

Enc(K, M):

- Choose IV at random
- Compute PRG(K, IV) xor M, where PRG is defined as before and it has size of M

Summary

- Desirable security: IND-CPA
- Block ciphers have weaker security than IND-CPA
- Block ciphers can be used to build IND-CPA secure encryption schemes by chaining in careful ways
- Stream ciphers provide another way to encrypt, inspired from one-time pads