CS161

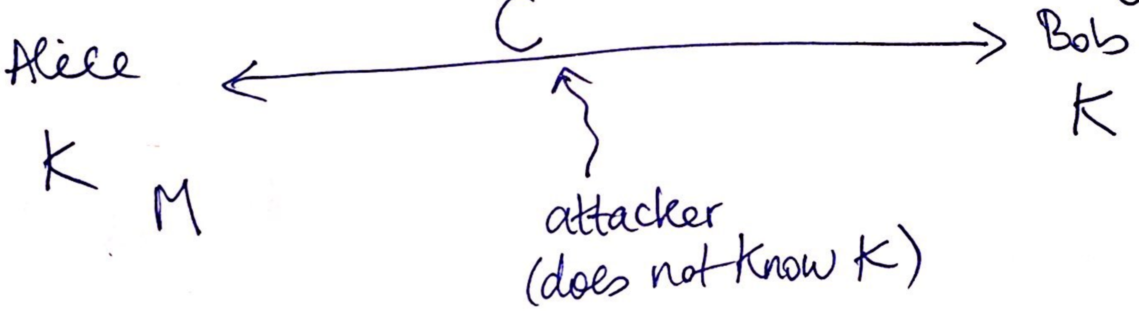
Announcements

Homework 1 was due yesterday.

Project 1 will be released Wednesday morning

Due 2/14/18 (Wednesday)

Syntax of encr scheme   Symetric key encryption



Keygen → K

Enc(K, M) → C
($\uparrow$ msg) ($\uparrow$ ciphertext)

Dec(K, C) → M

Security: C does not reveal any information on M other than length

Correctness:

$\forall K, \forall M, \forall C \leftarrow Enc(K, M),$
($\uparrow$ for all)

$Dec(K, C) \rightarrow M$

Kerchoff's principle: attacker knows enc algorithm but not keys

Possible definition $Pr[A(C) \rightarrow M] = negl$

bad:

Idea: attacker does not learn any partial information about M, any $f(M)$ other than length

One-time pad (OTP)

Alice

same length: $K = k_1 \dots k_n$, $M = M_1 \dots M_n$

Bob

$K = k_1 \dots k_n$

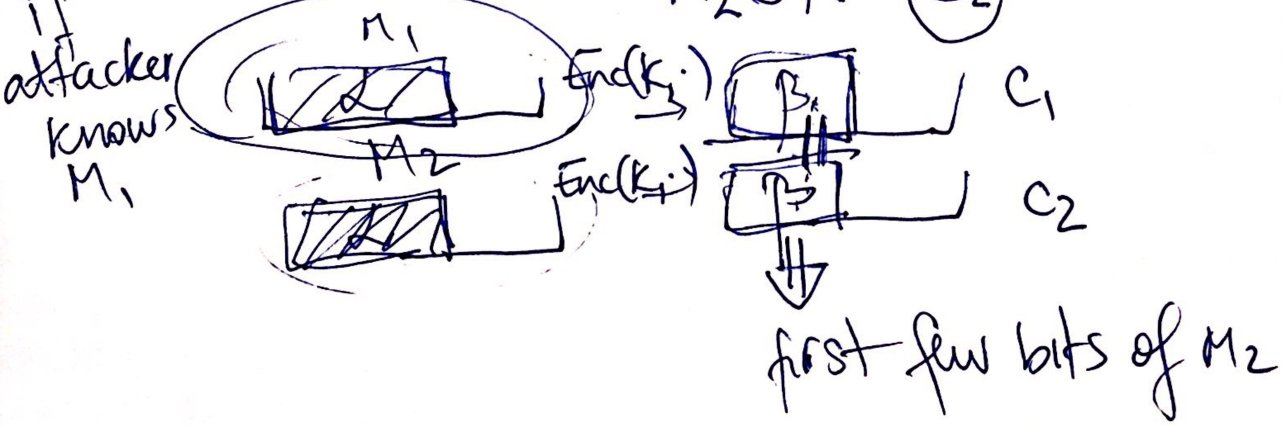
Keygen → choose n random bits: K

$$Enc(K, M) = K \oplus M = C$$

$$Dec(K, C) = K \oplus C = \underbrace{K \oplus (K \oplus M) = M}_{\text{correctness}}$$

Warning: only one message encrypted with key, else not secure

$$M_1 \otimes K = C_1, \quad M_2 \otimes K = C_2 \Rightarrow C_1 \otimes C_2 = M_1 \otimes M_2$$



Security game: IND-KPA

IND → indistinguishability under

K → known

PA → plaintext attack

Adv — Ch

m_0, m_1 (plaintext) →

Keygen() → K.

b random ← {0,1}

m_b

$C = Enc(K, m_b)$

← C

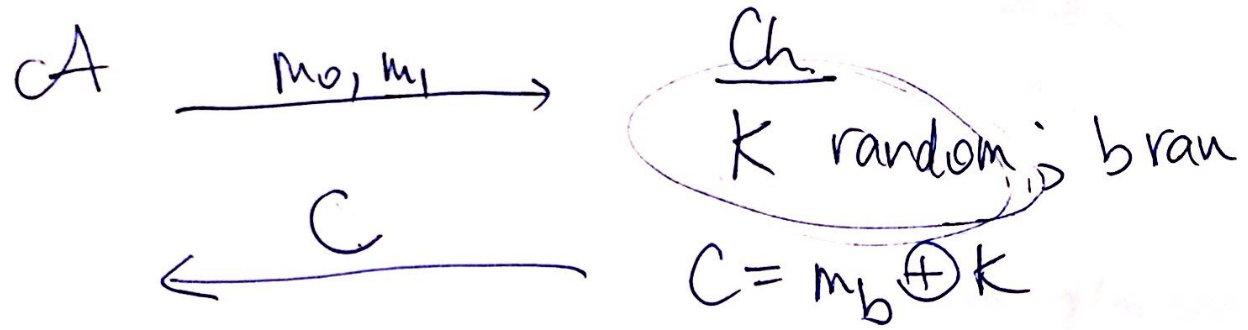
∀ (poly) time attackers,

$$Pr[\mathcal{A}(C) \rightarrow b] \leq \frac{1}{2} + negl$$

negl ↓ $\frac{1}{2^{128}}$

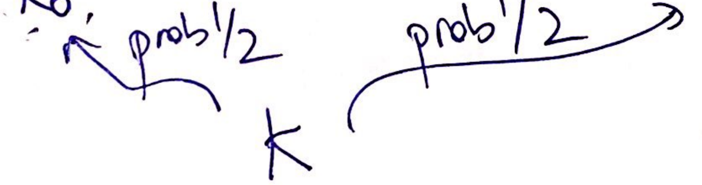
OTP is IND-KPA secure

Show: $\Pr[\mathcal{A}(C) \to b] = 1/2$



Given C, the message m_b could have been m_0 or m_1 with same probabilty.

$C = m_0 \otimes \underbrace{(m_0 \otimes C)}_{K_0}$ $\qquad$ $C = m_1 \otimes \underbrace{(m_1 \otimes C)}_{K_1}$



$\Pr[\mathcal{A}(C) \to b] = 1/2$

Limitations: - only use once
- message size is $\leq$ key size

$\Rightarrow$ Symmetric-key encryption fixes these.

Block ciphers — reuse the key for multiple encryptions

Alice — K

Bob — K.

$$\text{Block cipher } E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

(n ↙ block cipher size)

$$E_K : \{0,1\}^n \rightarrow \{0,1\}^n$$

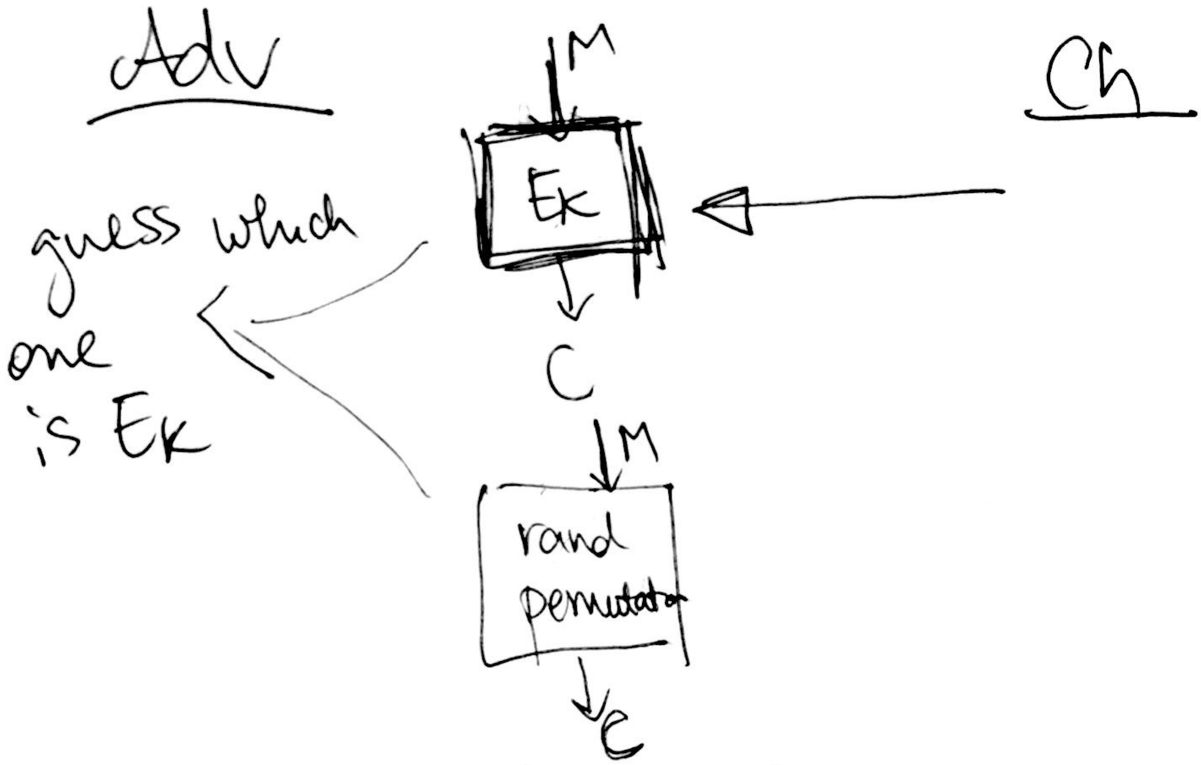
$$E_K(M) = C \; ; \; D_K(C) = M$$

1) E is a permutation (one-to-one / bijection)

$$E(K, M) \rightarrow C$$

(scrambles)

2) Security: E_k "behaves like" a ~~per~~ random permutation



$$\Pr[Adv(\square, \square) = \text{guesses correctly which is } E_k] \leq \frac{1}{2} + \text{negl}$$

Symmetric-Key Cryptography

CS 161: Computer Security

Prof. Raluca Ada Popa

Jan 30, 2018

Announcements

- Project 1 out this week, due 2 weeks from release date

Special guests

- Alice 
- Bob 
- The attacker (Eve - “eavesdropper”, Malice)  
- Sometimes Chris too

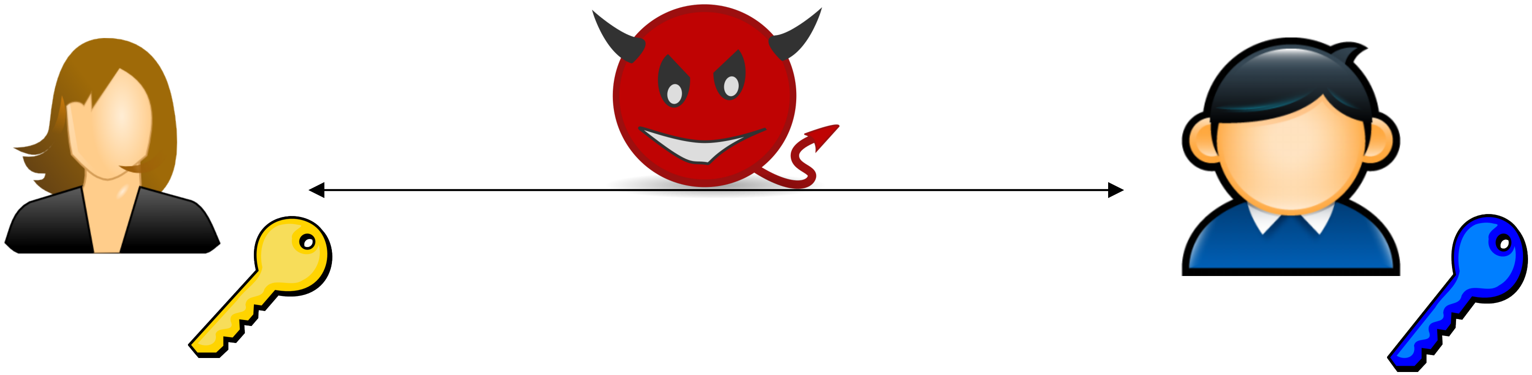
Cryptography

- Narrow definition: secure communication over insecure communication channels
- Broad definition: a way to provide formal guarantees in the presence of an attacker

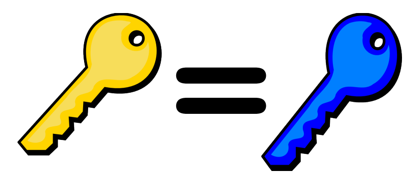
Three main goals

- Confidentiality: preventing adversaries from reading our private data,
- Integrity: preventing attackers from altering some data,
- Authenticity: determining who created a given document

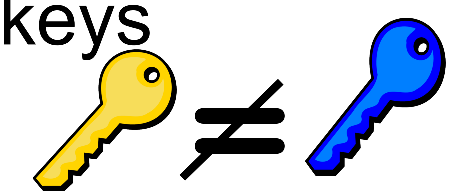
Modern Cryptography



- Symmetric-key cryptography
  - The same secret key is used by both endpoints of a communication



- Public-key (asymmetric-key) cryptography
  - Sender and receiver use different keys



Today: Symmetric-key Cryptography

Whiteboard & notes:

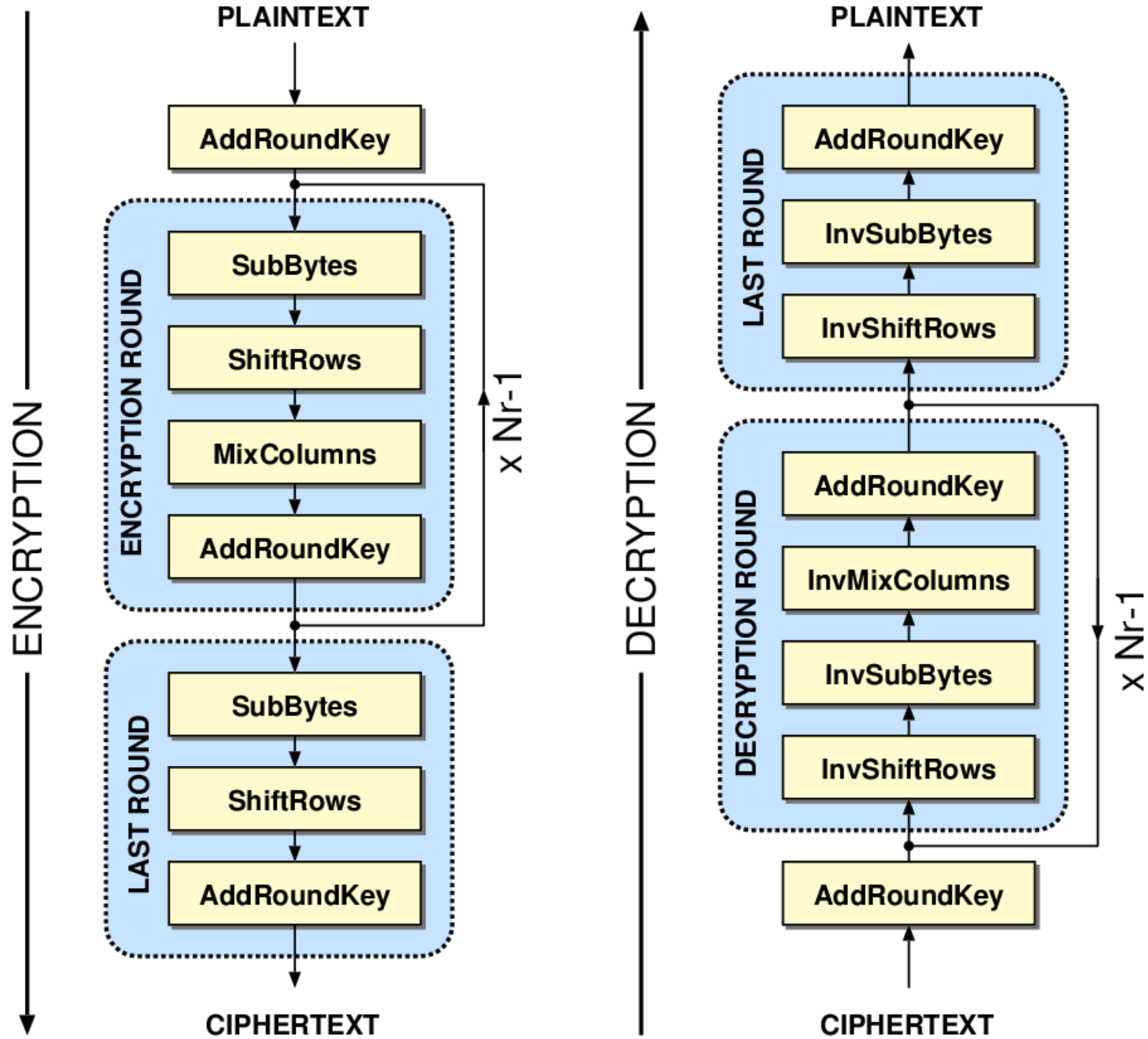
- Symmetric encryption definition
- Security definition
- One time pad (OTP)
- Block cipher

Advanced Encryption Standard (AES)

- Block cipher developed in 1998 by Joan Daemen and Vincent Rijmen
- Recommended by US National Institute for Standard and Technology (NIST)
- Block length $n = 128$, key length $k = 256$

AES ALGORITHM

Just giving you a sense, no need to understand why its so

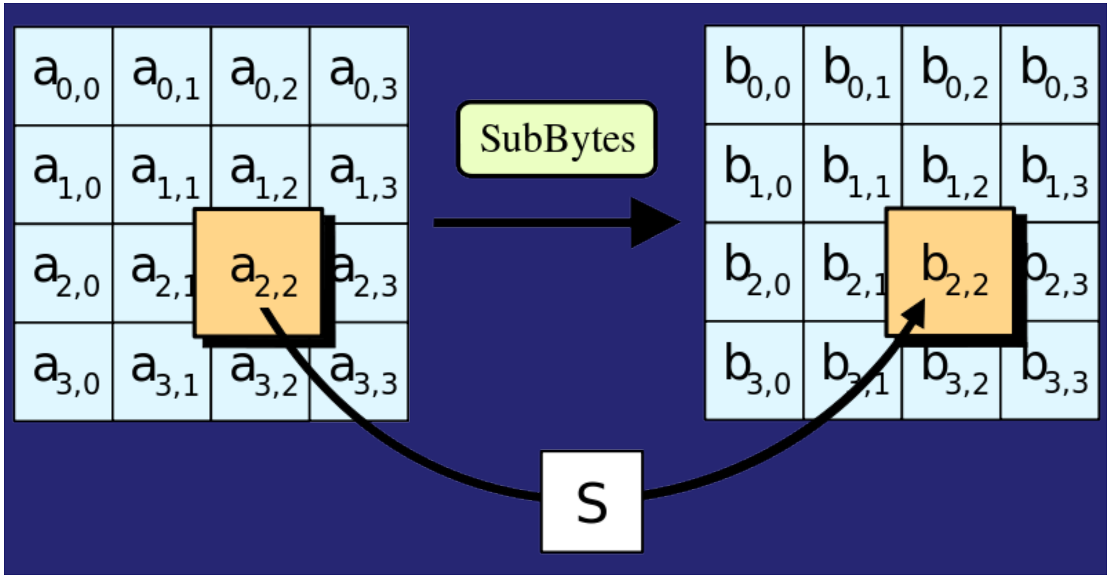


- 14 cycles of repetition for 256-bit keys.

AES slides, credit Kevin Orr

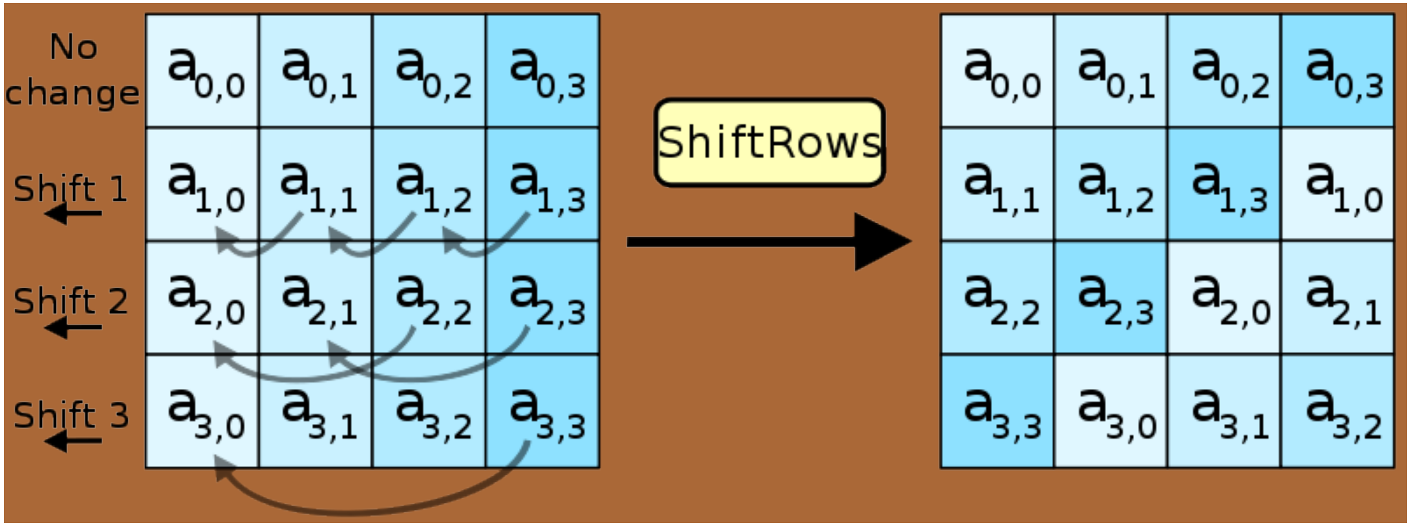
Algorithm Steps - Sub bytes

- each byte in the *state* matrix is replaced with a SubByte using an 8-bit substitution box
- $b_{ij} = S(a_{ij})$



Shift Rows

- Cyclically shifts the bytes in each row by a certain offset
- The number of places each byte is shifted differs for each row



Why secure?

- Not provably secure
- By “educated” belief/assumption: it stood the test of time and of much cryptanalysis (field studying attacks on encryption schemes)
- Various techniques to boost confidence in its security
- If we were to even have something probably secure, P is not NP

Uses

- Government Standard
  - AES is standardized as Federal Information Processing Standard 197 (FIPS 197) by NIST
  - To protect classified information
- Industry
  - SSL / TLS
  - SSH
  - WinZip
  - BitLocker
  - Mozilla Thunderbird
  - Skype

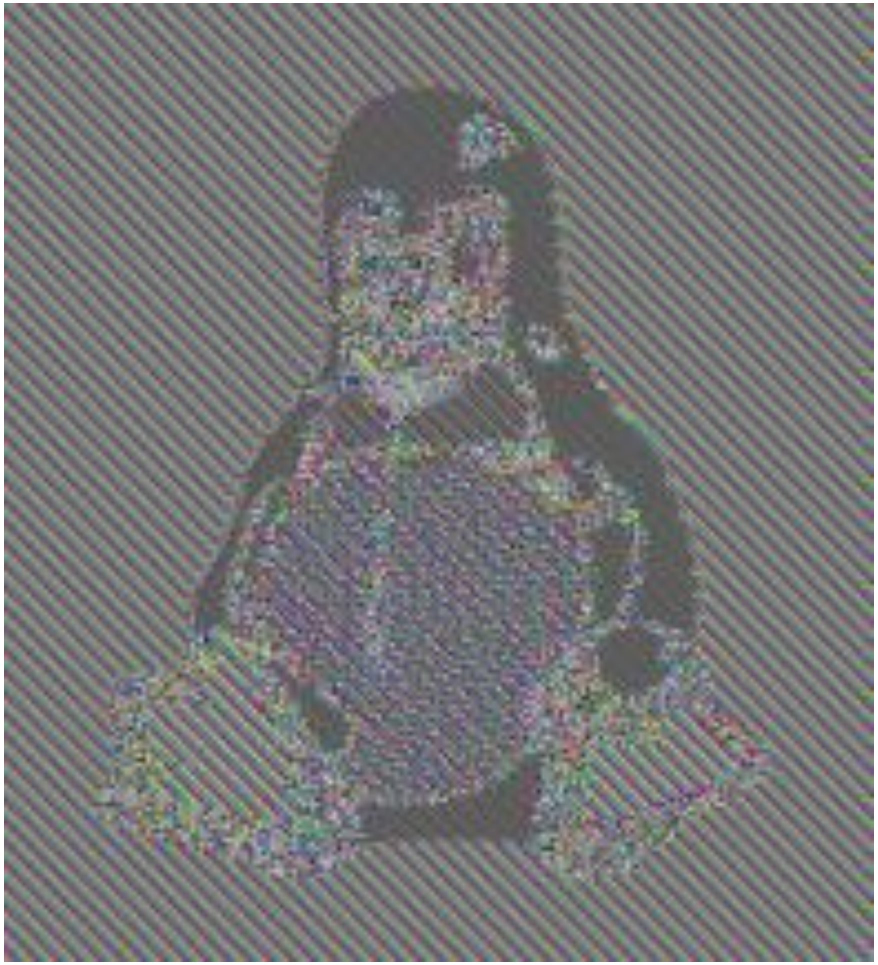
But used as part of symmetric-key encryption or other crypto tools

Symmetric-key encryption from block ciphers

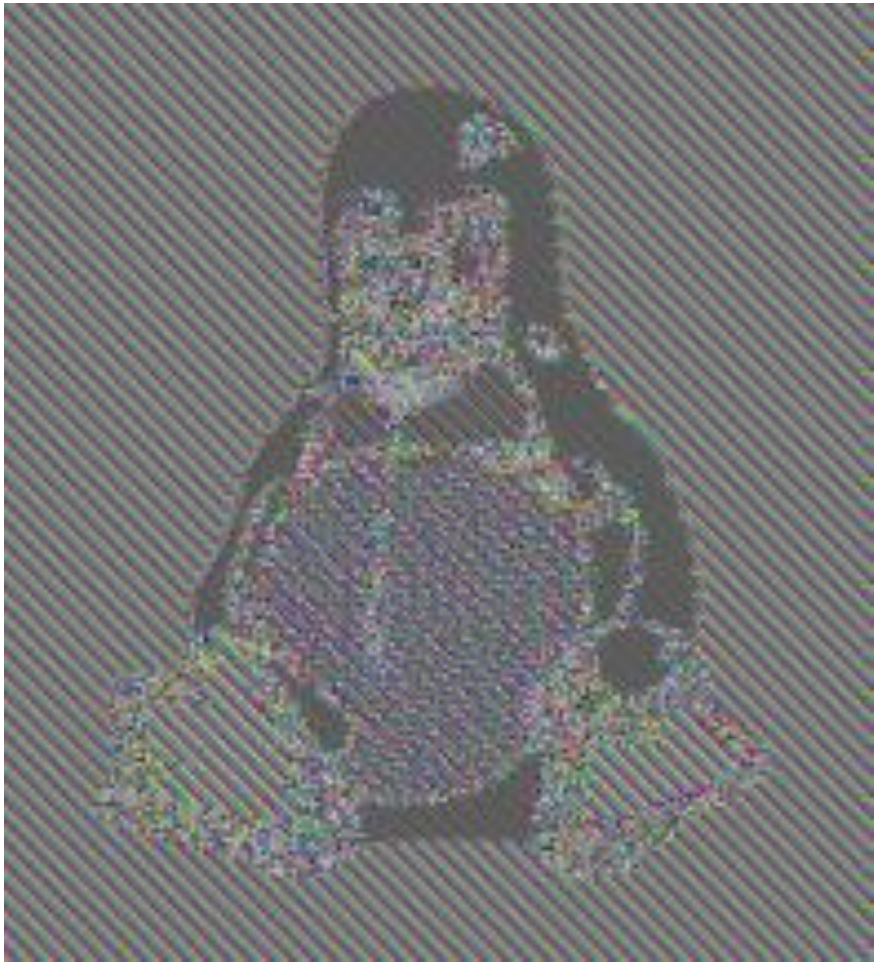
Why block ciphers not enough for encryption by themselves?

- Can only encrypt messages of a certain size
- If message is encrypted twice, attacker knows it is the same message

Original image



Eack block encrypted with a block cipher



Later (identical) message again encrypted

Symmetric key encryption scheme

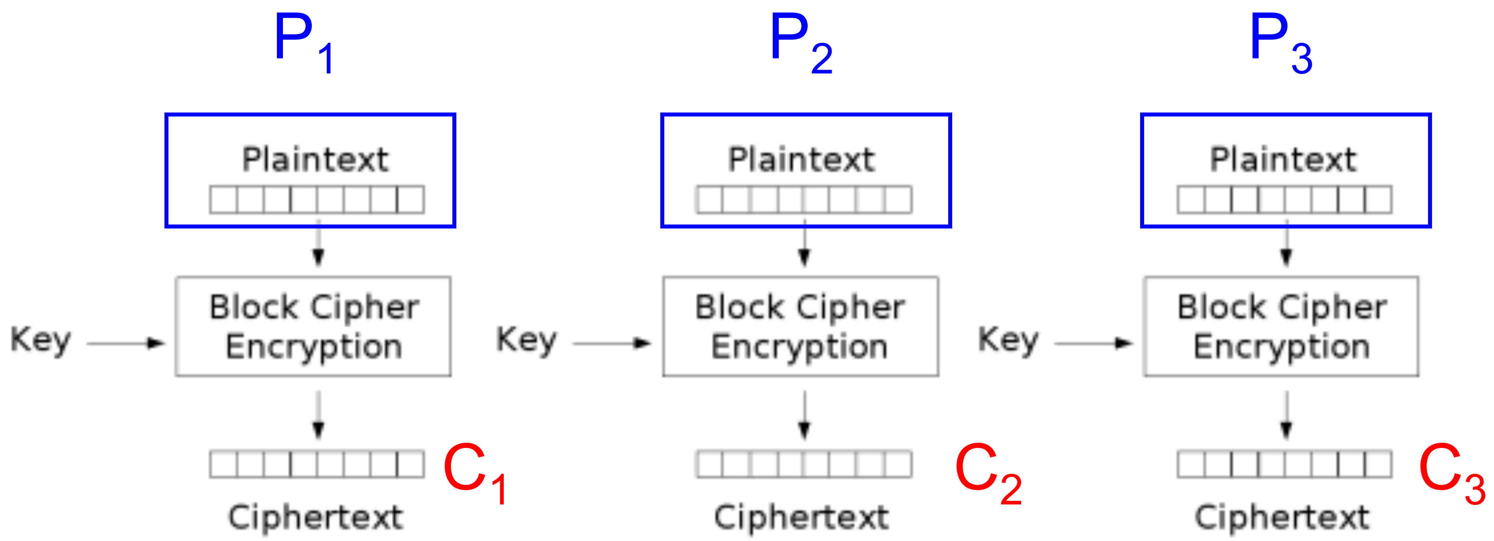
- Can be reused (unlike OTP)
- Builds on block ciphers:
  - Can be used to encrypt long messages
  - Wants to hide that same block is encrypted twice
- Uses block ciphers in certain modes of operation

Electronic Code Book (ECB)

- Split message M 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



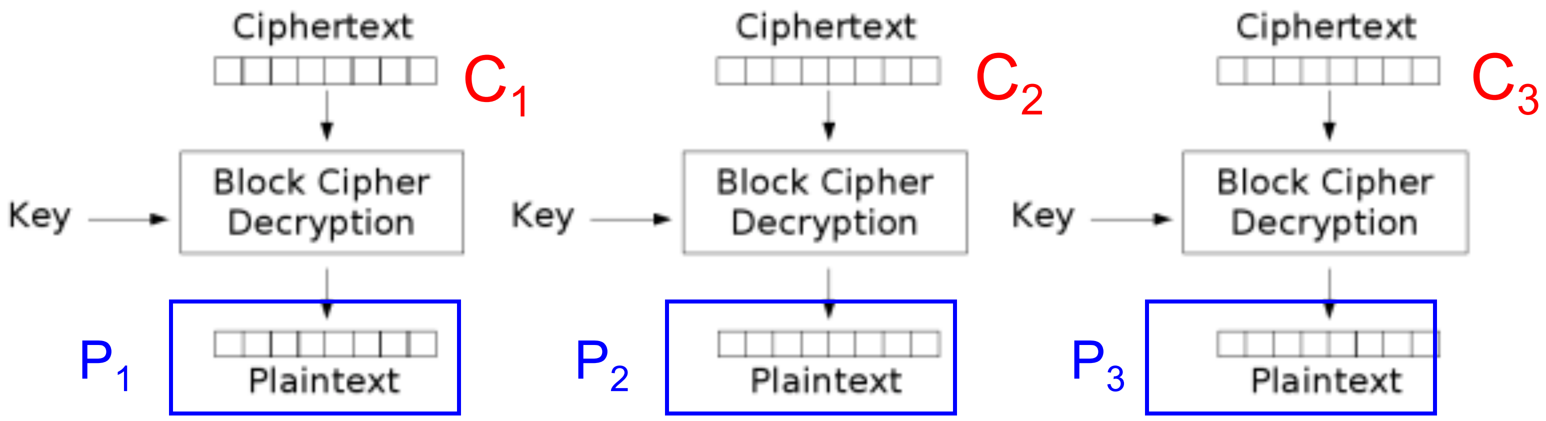
Electronic Codebook (ECB) mode encryption

KeyGen = key gen of block cipher

break message M into P1|P2|…|Pn
Enc(K, P1|P2|..|Pn) = (C1, C2,..., Pn)

Dec(K, (C1,C2,..,Pn)) = (P1, P2, .., Pn)

Decryption



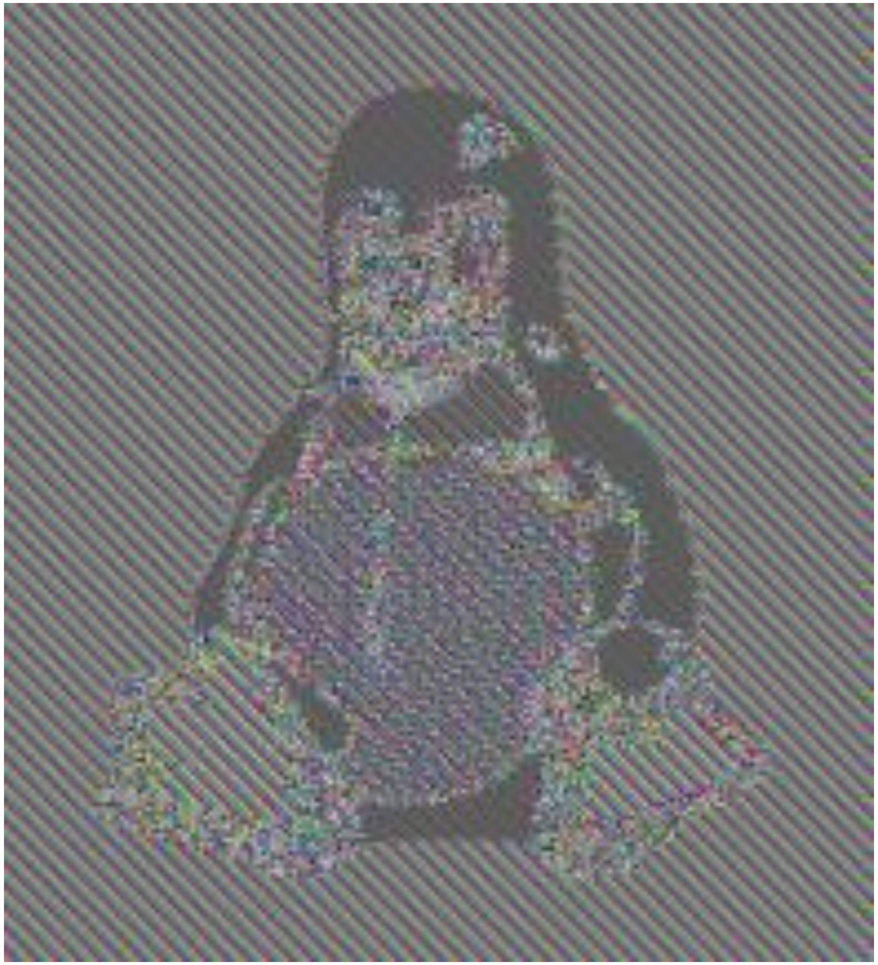
Electronic Codebook (ECB) mode decryption

What is the problem with ECB?

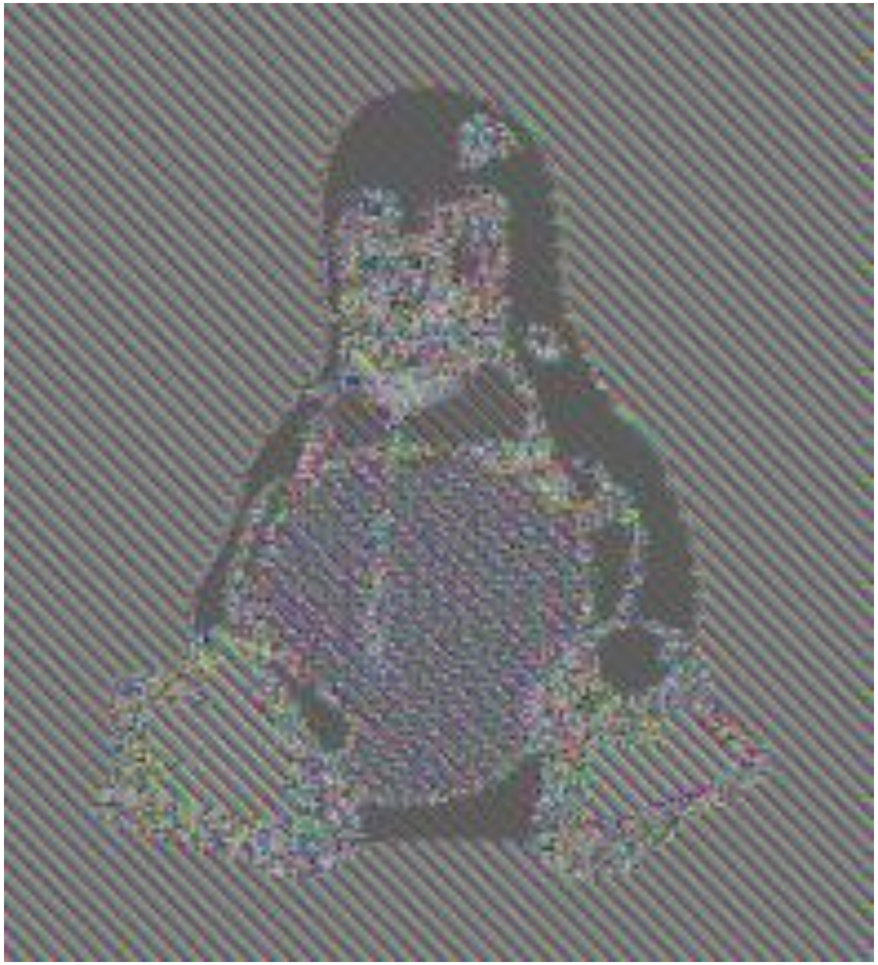
Does this achieve IND-KPA?

No, attacker can tell if $P_i = P_j$

Original image

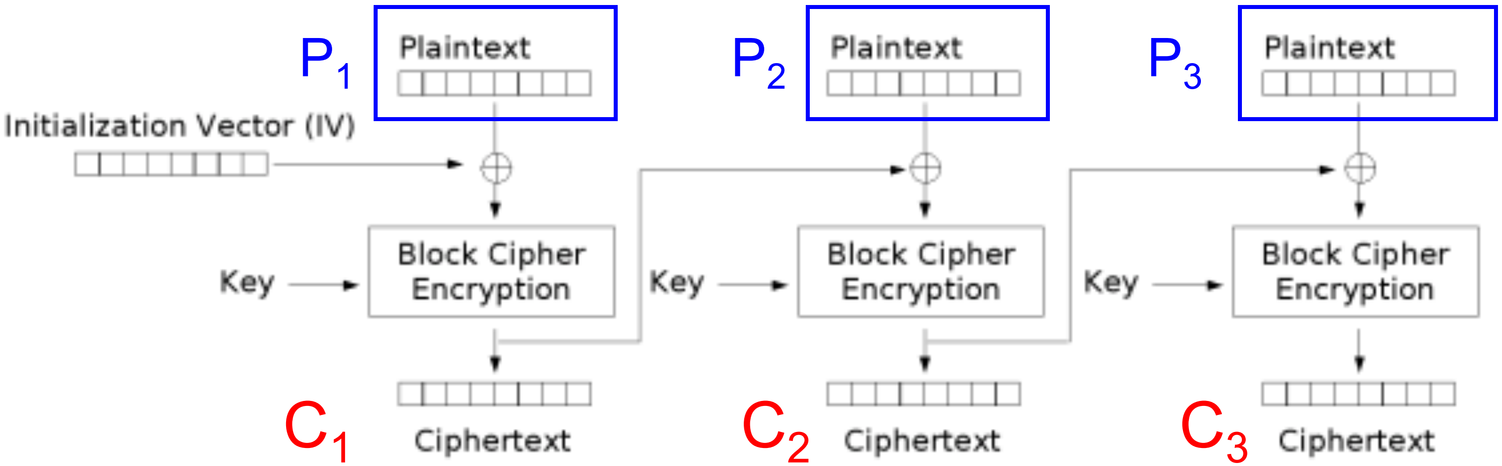


Encrypted with ECB



Later (identical) message again encrypted with ECB

CBC: Encryption

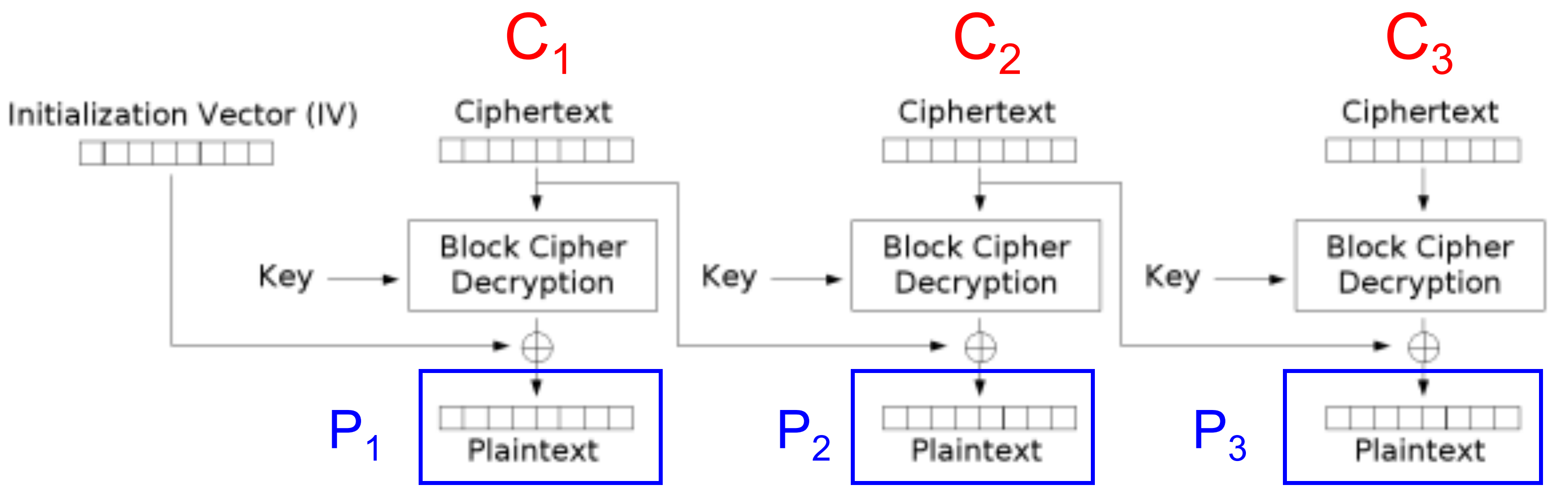


Cipher Block Chaining (CBC) mode encryption

IV may not repeat for messages with same P_1, choose it at random; not secret, part of ciphertext

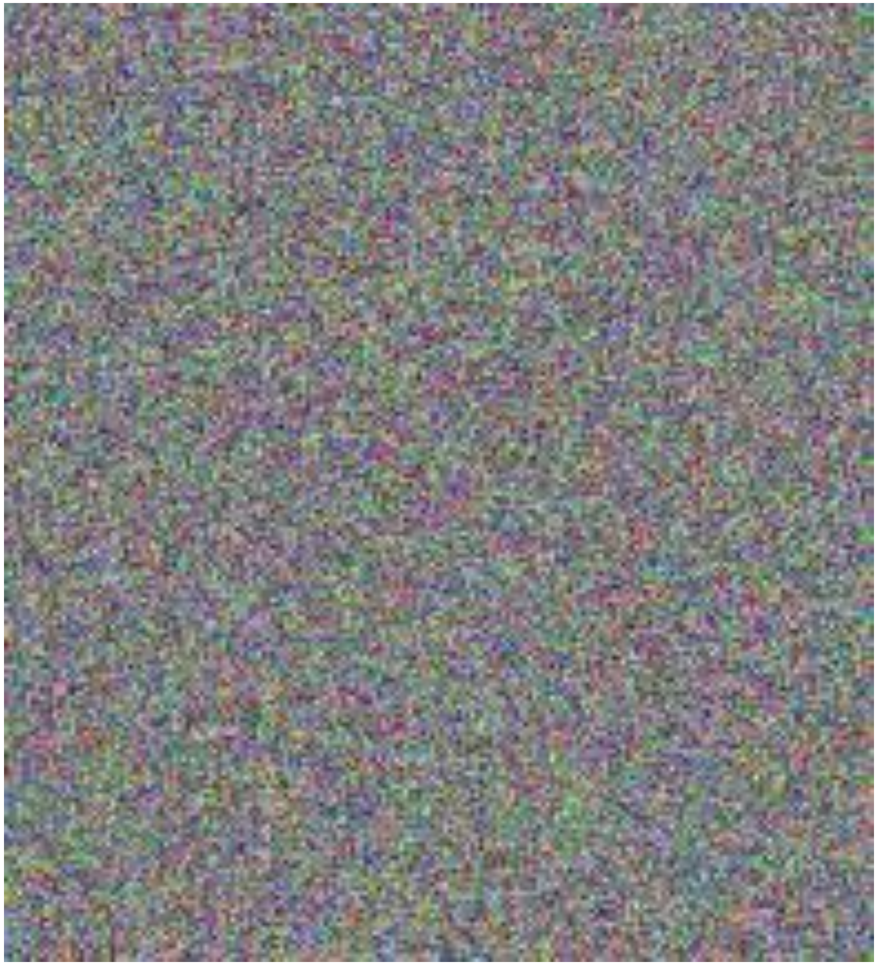
```
break message M into P1|P2|…|Pn
Enc(K, P1|P2|..|Pn) = (IV, C1, C2,..., Pn)

Dec(K, (IV,C1,C2,..,Pn)) = (P1, P2, .., Pn)
```

CBC: Decryption



Cipher Block Chaining (CBC) mode decryption

Original image



Encrypted with CBC

CBC

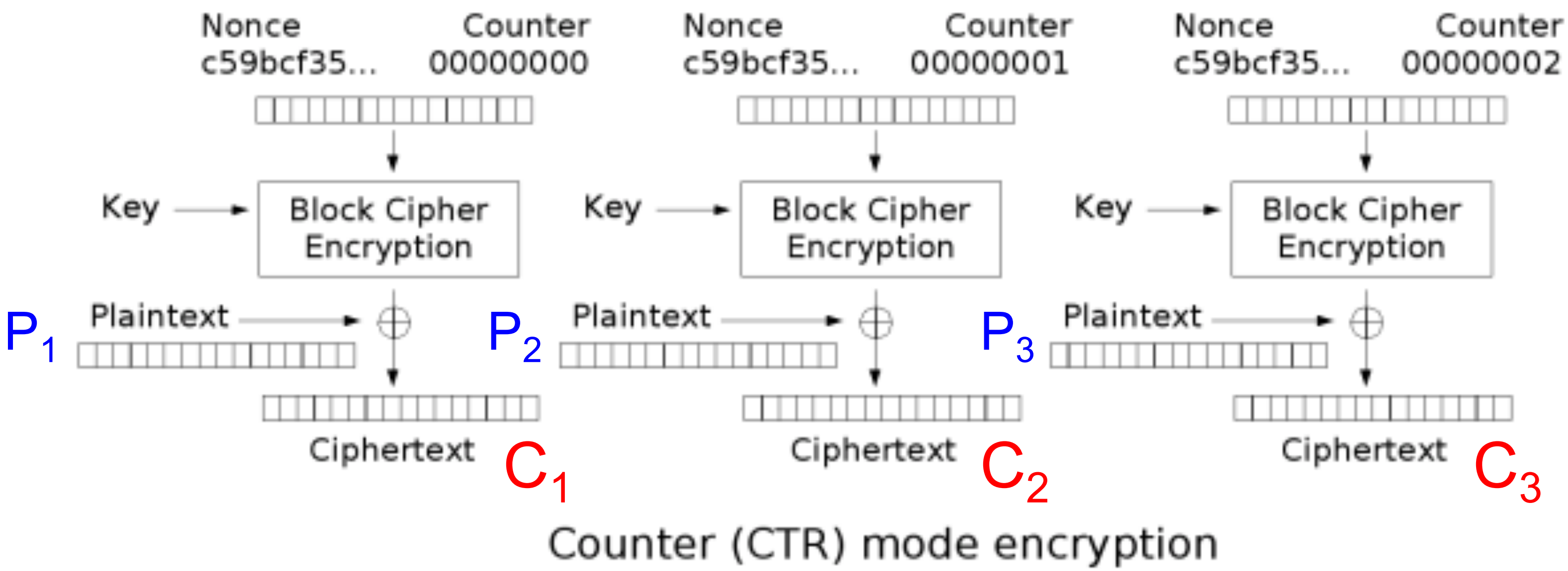
Popular, still widely used
Achieves IND-KPA, and more (IND-CPA)

Caveat: sequential encryption, hard to parallelize

CTR mode gaining popularity

CTR: Encryption

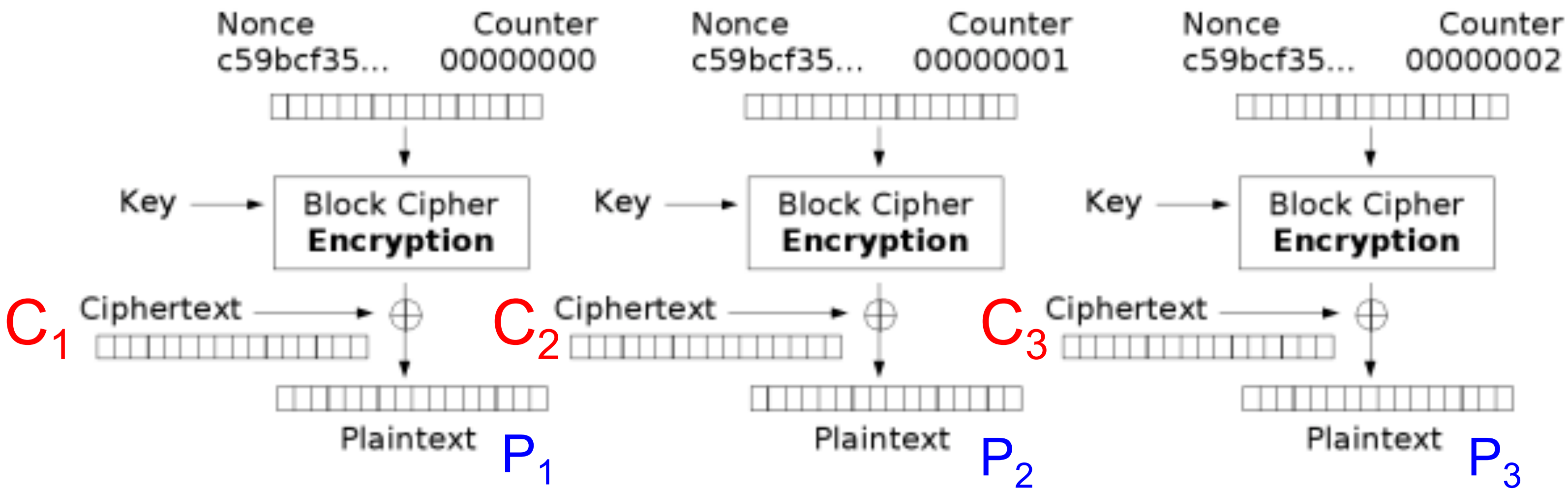
`Enc(K, P1|P2|P3) = (nonce, C1, C2, C3)`



Nonce is similar to IV for CBC, one should not use the same nonce for two messages; choose it at random

CTR: Decryption

```
Dec(K, (nonce,C1,C2,C3)) = (P1, P2, P3)
```



Counter (CTR) mode decryption

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

CBC vs CTR

Security: Both IND-KPA, and even IND-CPA
If you ever reuse the same nonce, CBC might leak some information about the initial plaintext blocks up to a first difference between two messages. CTR can leak information about various blocks in the message.

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

Summary

- Encryption protects confidentiality
- IND-KPA is a security game expressing message indistinguishability
- OTP is secure if used only once
- Block ciphers help build symmetric-key encryption schemes with reusable sizes and arbitrary message lengths by chaining them in cipher modes