Ciphers and shared-key encryption

Daniel Bosk[1]

KTH EECS

17th January 2026

[1] With some ideas from Douglas Wikström.

Exercise

When you hear ‘cryptography’, what do you think it does?

- The word has its origin in greek[2]:

  χρυπτός (*kryptos*) meaning hidden[3].

  γράφος (*graphos*) meaning writing[4].

- The area has been around for ages.
- We should not confuse it with *steganography*.
- Steganography concerns hiding a message's *existence*.
- Cryptography concerns hiding a message's *contents*.

[2] 'cryptography, n.'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/45374?redirectedFrom=cryptography&.

[3] 'crypto-, comb. form'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/45363.

[4] 'graphy-, comb. form'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/80855.

- The word has its origin in greek[2]:

  κρυπτός (*kryptos*) meaning hidden[3].
  γράφος (*graphos*) meaning writing[4].

- The area has been around for ages.
- We should not confuse it with *steganography*.
- Steganography concerns hiding a message's *existence*.
- Cryptography concerns hiding a message's *contents*.

[2] 'cryptography, n.'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/45374?redirectedFrom=cryptography&.

[3] 'crypto-, comb. form'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/45363.

[4] 'graphy-, comb. form'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/80855.

- The word has its origin in greek[2]:

  κρυπτός (*kryptos*) meaning hidden[3].
  γράφος (*graphos*) meaning writing[4].

- The area has been around for ages.
- We should not confuse it with *steganography*.
- Steganography concerns hiding a message's *existence*.
- Cryptography concerns hiding a message's *contents*.

---

[2] 'cryptography, n.'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/45374?redirectedFrom=cryptography&.

[3] 'crypto-, comb. form'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/45363.

[4] 'graphy-, comb. form'. In: *OED Online*. Hämtad den 5 april 2013. Oxford University Press, Mar. 2013. URL: http://www.oed.com/view/Entry/80855.

‘Cryptography is concerned with the conceptualization, definition, and construction of computing systems that address security concerns.’[5]

[5] Oded Goldreich. *Foundations of cryptography, Vol. 1: Basic tools*. Cambridge: Cambridge Univ. Press, 2001.

- Then it was an art, now it's a science.
- People used 'clever' constructions.
- These were thought to be secure: 'How can anyone figure this out?'
- Well, it turns out that there are always a lot of people with a lot of time and motivation ...

- Then it was an art, now it's a science.
- People used 'clever' constructions.
- These were thought to be secure: 'How can anyone figure this out?'
- Well, it turns out that there are always a lot of people with a lot of time and motivation ...

- Then it was an art, now it's a science.
- People used 'clever' constructions.
- These were thought to be secure: 'How can anyone figure this out?'
- Well, it turns out that there are always a lot of people with a lot of time and motivation ...

Shared-key (symmetric) cryptography Stems from the classical crypto where a key is shared between two users.

Public-key (asymmetric) cryptography This is more modern crypto, from 1970s. Each user has a public and a private key.

Counter-intuitive cryptography More modern, from 1980s and onwards. How to do computations on secret inputs, prove knowledge without revealing of what.

Fully homomorphic encryption More recent (early 2000s) allowing arbitrary computations on encrypted data without decrypting it first. This has implications for cloud computing and data privacy.

Question

Alice and Bob wish to communicate in private. You all know how to do this with friends! How?

Question

Is the following possible?

1. Alice transforms her message m into a garbled string c.
2. Alice sends the garbled string c to Bob.
3. Bob transforms c back to m.

Goal: c hides the content of m.

Question

Is the following possible?

1. Alice transforms her message m into a garbled string c.
2. Alice sends the garbled string c to Bob.
3. Bob transforms c back to m.

Goal: c hides the content of m.

Exercise

What do we need for this to work?

Solution

We have two options:

1. *Either we keep the garbling procedure secret.*
2. *Or we need a secret component in a publicly known garbling.*

Example (Secret language)

'Rövarspråket':

- Each consonant x is replaced by the triplet xox.
- Each vowel is left unchanged.

Definition (Joint secret)

A *joint secret* is a piece of shared information of Alice and Bob that is hard to guess for others.

Example (Joint secret: shared memory)

'Remember how many fish we caught that rainy day?'

Example (Joint secret: random bits)

Alice and Bob agree on a sequence of randomly chosen bits intended for use later.

[A cryptosystem] should not require secrecy, and it should not be a problem if it falls into the enemy hands;

Kerckhoff's Principle

- No security-by-obscurity
- The key should be the only secret

[A cryptosystem] should not require secrecy, and it should not be a problem if it falls into the enemy hands;

Kerckhoff's Principle

- No security-by-obscurity
- The key should be the only secret

Remark

- This doesn't mean we must tell the adversary what we're using.
- But we shouldn't loose any security if we do.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Block-cipher encryption

Input A fixed-sized *key* k, a fixed-sized block of *plaintext* p.

Output A fixed-sized block of *ciphertext* c.

Notation $\mathrm{Enc}_k(p) = c$

Block-cipher decryption

Input A fixed-sized *key* k, a fixed-sized block of *ciphertext* c.

Output A fixed-sized block of *plaintext* p.

Notation $\mathrm{Dec}_k(c) = p$

Block-cipher encryption

Input A fixed-sized *key* k, a fixed-sized block of *plaintext* p.

Output A fixed-sized block of *ciphertext* c.

Notation $\mathrm{Enc}_k(p) = c$

Block-cipher decryption

Input A fixed-sized *key* k, a fixed-sized block of *ciphertext* c.

Output A fixed-sized block of *plaintext* p.

Notation $\mathrm{Dec}_k(c) = p$

Definition (Crypto system)

[6] A *crypto system* is a tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where:

- $\mathcal{M}$ is a finite set of *plaintexts* or messages,
- $\mathcal{C}$ is a finite set of *ciphertexts*,
- $\mathcal{K}$ is the *keyspace*, a finite set of keys,
- $\mathcal{E}$ and $\mathcal{D}$ are the sets of encryption and decryption rules, respectively.

For every $k \in \mathcal{K}$ there is an $\mathrm{Enc}_k \in \mathcal{E}$ and a $\mathrm{Dec}_k \in \mathcal{D}$ such that:

- $\mathrm{Enc}_k \colon \mathcal{M} \to \mathcal{C}$ and $\mathrm{Dec}_k \colon \mathcal{C} \to \mathcal{M}$ are functions, and
- $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$ for all plaintexts $m \in \mathcal{M}$.

[6] Douglas R. Stinson. *Cryptography: Theory and Practice*. 3rd ed. Boca Raton: Chapman & Hall/CRC, 2006.

Definition (Crypto system)

[6] A *crypto system* is a tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where:

- $\mathcal{M}$ is a finite set of *plaintexts* or messages,
- $\mathcal{C}$ is a finite set of *ciphertexts*,
- $\mathcal{K}$ is the *keyspace*, a finite set of keys,
- $\mathcal{E}$ and $\mathcal{D}$ are the sets of encryption and decryption rules, respectively.

For every $k \in \mathcal{K}$ there is an $\mathrm{Enc}_k \in \mathcal{E}$ and a $\mathrm{Dec}_k \in \mathcal{D}$ such that:

- $\mathrm{Enc}_k \colon \mathcal{M} \to \mathcal{C}$ and $\mathrm{Dec}_k \colon \mathcal{C} \to \mathcal{M}$ are functions, and
- $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$ for all plaintexts $m \in \mathcal{M}$.

[6] Douglas R. Stinson. *Cryptography: Theory and Practice*. 3rd ed. Boca Raton: Chapman & Hall/CRC, 2006.

Definition (Shift Cipher)

We define it as follows:

- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{29}$.
- For each $k \in \mathcal{K}$ we define

$$\text{Enc}_k(m) = (m + k) \bmod 29, m \in \mathcal{M}, \text{ och}$$
$$\text{Dec}_k(c) = (c - k) \bmod 29, c \in \mathcal{C}.$$

Example

With key $k = 3$, we encrypt individual letters as follows:

- $\text{Enc}_3(7) = 7 + 3 \bmod 29 = 10$ h$\rightarrow$J
- $\text{Enc}_3(4) = 4 + 3 \bmod 29 = 7$ e$\rightarrow$G

Definition (Shift Cipher)

We define it as follows:

- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{29}$.
- For each $k \in \mathcal{K}$ we define

$$\text{Enc}_k(m) = (m + k) \bmod 29, m \in \mathcal{M}, \text{ och}$$
$$\text{Dec}_k(c) = (c - k) \bmod 29, c \in \mathcal{C}.$$

Example

With key $k = 3$, we encrypt individual letters as follows:

- $\text{Enc}_3(7) = 7 + 3 \bmod 29 = 10$ h→J
- $\text{Enc}_3(4) = 4 + 3 \bmod 29 = 7$ e→G
- $\text{Enc}_3(9) = 9 + 3 \bmod 29 = 12$ j→L

Remark

- The shift cipher is a classical cipher — also know as the Caesar Cipher.
- It's easily broken *by hand*!
- It's used here for illustrative purposes.

Definition (Perfect secrecy)

[7] Consider a cryptosystem $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ with stochastic variables M (message) and C (ciphertext):

- The system has *perfect secrecy* if and only if

$$\Pr(M = m \mid C = c) = \Pr(M = m)$$

for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

Remark

Equivalent to $H(M \mid C) = H(M)$, i.e. ciphertext does not reveal anything about plaintext.

[7] Claude E Shannon. 'Communication theory of secrecy systems'. In: *Bell system technical journal* 28.4 (1949), pp. 656–715.

Definition (Perfect secrecy)

[7] Consider a cryptosystem $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ with stochastic variables M (message) and C (ciphertext):

- The system has *perfect secrecy* if and only if

$$\Pr(M = m \mid C = c) = \Pr(M = m)$$

for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

Remark

Equivalent to $H(M \mid C) = H(M)$, i.e. ciphertext does not reveal anything about plaintext.

[7] Claude E Shannon. 'Communication theory of secrecy systems'. In: *Bell system technical journal* 28.4 (1949), pp. 656–715.

Theorem (Shannon's theorem)

- *Assume cryptosystem* $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *such that* $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$.
- *This provides perfect secrecy if and only if*
  1. *every key* $k \in \mathcal{K}$ *is used with equal probability* $1/|\mathcal{K}|$,
  2. *for every plaintext* $m \in \mathcal{M}$ *and* $c \in \mathcal{C}$ *there is a unique key such that* $\mathrm{Enc}_k(m) = c$.

Theorem (Shannon's theorem)

- *Assume cryptosystem* $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *such that* $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$.
- *This provides perfect secrecy if and only if*
  1. *every key* $k \in \mathcal{K}$ *is used with equal probability* $1/|\mathcal{K}|$,
  2. *for every plaintext* $m \in \mathcal{M}$ *and* $c \in \mathcal{C}$ *there is a unique key such that* $\mathsf{Enc}_k(m) = c$.

Example (One-time Pad)

We construct it as follows:

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$, we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n).$$

- We also define $\mathrm{Dec} = \mathrm{Enc}$.
- The key $k \in \mathcal{K}$ must be chosen uniformly at random for each encryption.

Example (One-time Pad)

We construct it as follows:

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$, we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n).$$

- We also define Dec = Enc.
- The key $k \in \mathcal{K}$ must be chosen uniformly at random for each encryption.

Example (One-time Pad)

We construct it as follows:

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$, we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n).$$

- We also define $\mathrm{Dec} = \mathrm{Enc}$.
- The key $k \in \mathcal{K}$ must be chosen uniformly at random for each encryption.

Example (One-time Pad)

We construct it as follows:

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$, we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n).$$

- We also define $\mathrm{Dec} = \mathrm{Enc}$.
- The key $k \in \mathcal{K}$ must be chosen uniformly at random for each encryption.

Definition (Pseudo-random permutation, PRP)

[8] Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$. We say F is a PRP if:

1. for any $k \in \{0,1\}^s$, F_k is a bijection;
2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

when we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[8] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. 1st ed. Boca Raton: Chapman & Hall/CRC, 2008.

Definition (Pseudo-random permutation, PRP)

[8] Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$. We say F is a PRP if:

1. for any $k \in \{0,1\}^s$, F_k is a bijection;
2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

when we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[8] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. 1st ed. Boca Raton: Chapman & Hall/CRC, 2008.

Definition (Pseudo-random permutation, PRP)

[8] Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$. We say F is a PRP if:

1. for any $k \in \{0,1\}^s$, F_k is a bijection;
2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

when we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[8] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. 1st ed. Boca Raton: Chapman & Hall/CRC, 2008.

Definition (Shift cipher (Caesar))

- Think of letters as integers modulo n (e.g., $n = 26$ or include space).
- Key $k \in \mathbb{Z}_n$.
- $\mathrm{Enc}_k(m) = (m + k) \bmod n$, $\mathrm{Dec}_k(c) = (c - k) \bmod n$.

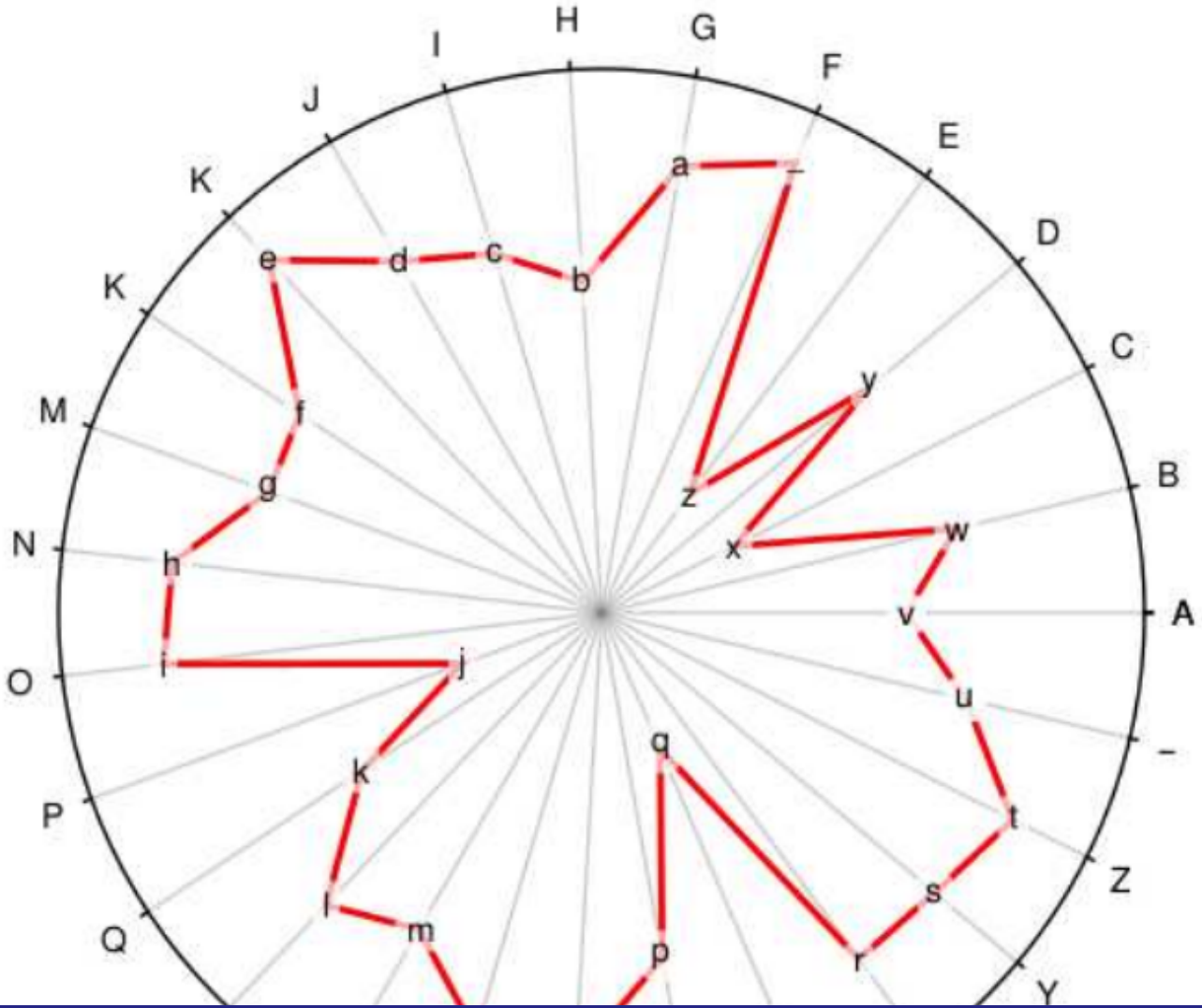
Exercise

You intercept a Caesar-encrypted message, but you do not know the key. How would you try to break it?

Cryptanalysis of Caesar

- Brute force: try all keys (tiny keyspace).
- Statistics: letter frequencies are preserved (just "rotated").
- If we guess one plaintext letter α maps to ciphertext letter β, then $k = \beta - \alpha \bmod n$.

Why classical ciphers fail



Definition (Substitution cipher)

We define it as follows:

- Key: a permutation σ of the alphabet.
- Encryption: $c_i = \sigma(m_i)$.
- Decryption: $m_i = \sigma^{-1}(c_i)$.

Cryptanalysis: structure still leaks

- Single-letter frequencies leak (A/E/T/... are uneven).
- Digrams/trigrams ("th", "the", ...) give more clues.
- A large keyspace is not enough if the scheme leaks structure.

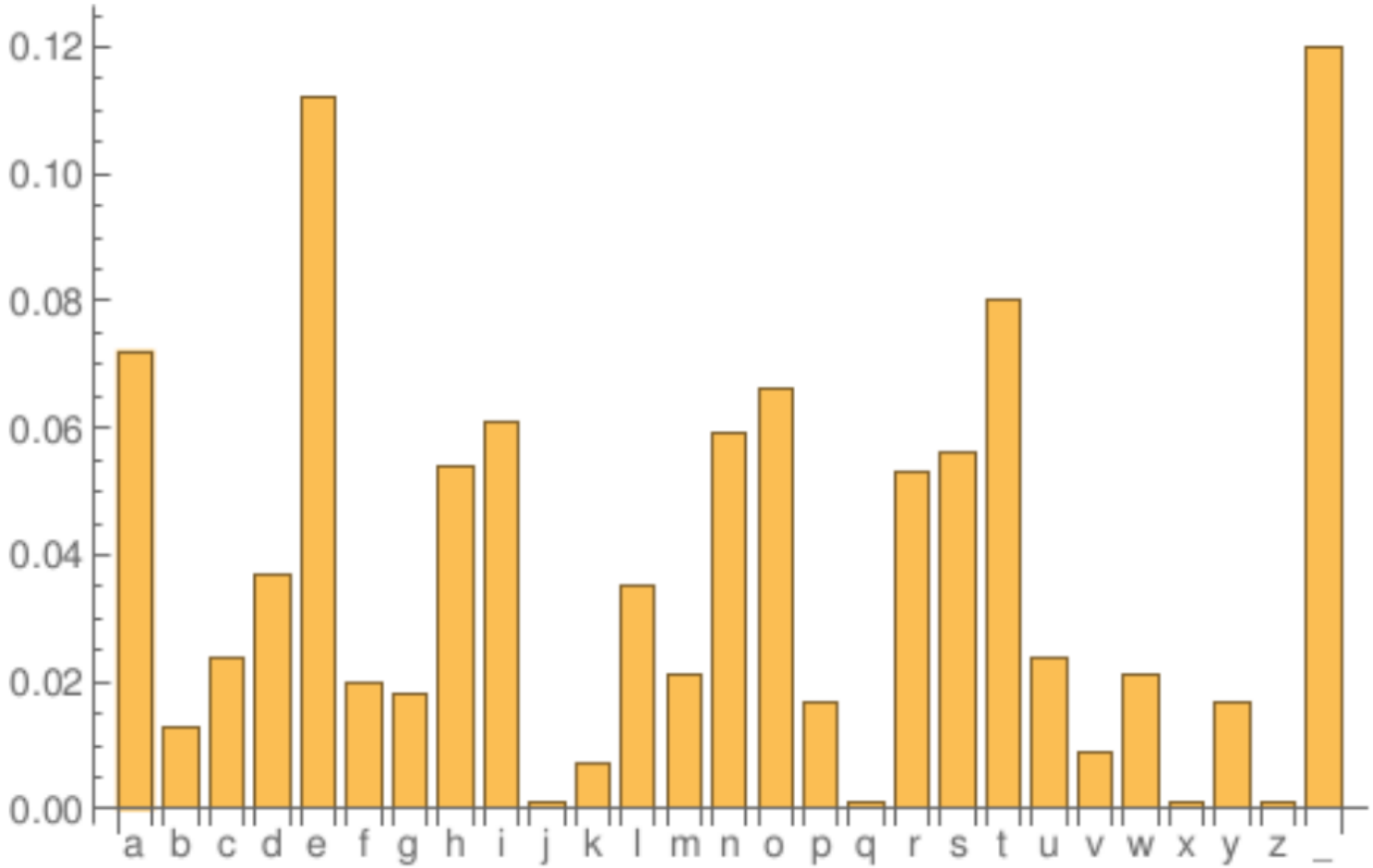


Figure: Frequency analysis of a substitution cipher. Even though

Definition (Vigénère cipher)

We define it as follows:

- Key: a short sequence $k_0, \ldots, k_{\ell-1}$ reused periodically.
- Encryption: $c_i = m_i + k_{i \bmod \ell} \bmod n$.
- Decryption: $m_i = c_i - k_{i \bmod \ell} \bmod n$.

Cryptanalysis of Vigénère (simple version)

- If we guess the key length ℓ, then we split the ciphertext into ℓ columns.
- Each column is a Caesar cipher with its own shift.
- Try plausible ℓ and test for language-like statistics.

Definition (Hill cipher)

We define it as follows:

- Key: an invertible $m \times m$ matrix K over $\mathbb{Z}_n$.
- Plaintext: a vector $\mathbf{p} = (p_1, \ldots, p_m)^T$ of m letters.
- Encryption: $\mathbf{c} = K \cdot \mathbf{p} \bmod n$.
- Decryption: $\mathbf{p} = K^{-1} \cdot \mathbf{c} \bmod n$.

Remark

This hides individual frequencies. This fixes the problem that broke Vigénère.

Definition (Hill cipher)

We define it as follows:

- Key: an invertible $m \times m$ matrix K over $\mathbb{Z}_n$.
- Plaintext: a vector $\mathbf{p} = (p_1, \ldots, p_m)^T$ of m letters.
- Encryption: $\mathbf{c} = K \cdot \mathbf{p} \bmod n$.
- Decryption: $\mathbf{p} = K^{-1} \cdot \mathbf{c} \bmod n$.

Remark

This hides individual frequencies. This fixes the problem that broke Vigénère.

Known-plaintext attack on Hill cipher

- Attacker knows m plaintext-ciphertext pairs: $(\mathbf{p}_1, \mathbf{c}_1), \ldots, (\mathbf{p}_m, \mathbf{c}_m)$.
- Form matrices $P = [\mathbf{p}_1 \cdots \mathbf{p}_m]$ and $C = [\mathbf{c}_1 \cdots \mathbf{c}_m]$.
- Then $C = K \cdot P$, so $K = C \cdot P^{-1} \bmod n$.

Exercise

We broke the classical ciphers using only the ciphertext (frequency analysis). But what other information might an attacker realistically have access to?

Attack models

Ciphertext-only Attacker sees only ciphertexts.

Known-plaintext Attacker knows some plaintext-ciphertext pairs.

Chosen-plaintext Attacker can choose plaintexts and obtain their encryptions.

Chosen-ciphertext Attacker can also choose ciphertexts and obtain their decryptions.

Definition (Perfect secrecy)

- Random variables M (message) and C (ciphertext).
- Perfect secrecy means: $\Pr(M = m \mid C = c) = \Pr(M = m)$ for all m, c.

Remark

Perfect secrecy is possible (one-time pad), but expensive in key material.

Definition (Perfect secrecy)

- Random variables M (message) and C (ciphertext).
- Perfect secrecy means: $\Pr(M = m \mid C = c) = \Pr(M = m)$ for all m, c.

Remark

Perfect secrecy is possible (one-time pad), but expensive in key material.

One-Time Pad (OTP)

- Key is as long as the message, uniformly random, used once.
- Gives perfect secrecy.
- Key distribution and key reuse are the practical obstacles.

Remark

What broke Vigénère was reusing the key!

Key length and brute force

- If the best attack is brute force, security grows with key length.
- Doubling key length squares brute-force work.
- Symmetric keys (e.g., 128–256 bits) can be enough in practice.

Exercise

Alice sends Bob an encrypted message.
An attacker flips a few bits in the ciphertext.
What do you think Bob will notice when decrypting?

Encryption does not give integrity

- Encryption hides content, but does not necessarily detect tampering.
- Bob can often decrypt a modified ciphertext into ‘garbage’ without knowing it was modified.
- Integrity requires additional mechanisms (covered later).

Bit flipping: block ciphers vs stream ciphers

Substitution-permutation network (SPN) block ciphers (AES Joan Daeme
Diffusion spreads the effect. Flipping one ciphertext bit corrupts many plaintext bits unpredictably.

Stream ciphers (OTP, CTR mode) No diffusion across bits. Flipping one ciphertext bit flips exactly one plaintext bit. This is *bit-by-bit malleability*.

Remark

Neither detects tampering! SPN ciphers just make tampering less predictable, not impossible.

Bit flipping: block ciphers vs stream ciphers

SPN block ciphers (AES) Diffusion spreads the effect. Flipping one ciphertext bit corrupts many plaintext bits unpredictably.

Stream ciphers (OTP, CTR mode) No diffusion across bits. Flipping one ciphertext bit flips exactly one plaintext bit. This is *bit-by-bit malleability*.

Remark

Neither detects tampering! SPN ciphers just make tampering less predictable, not impossible.

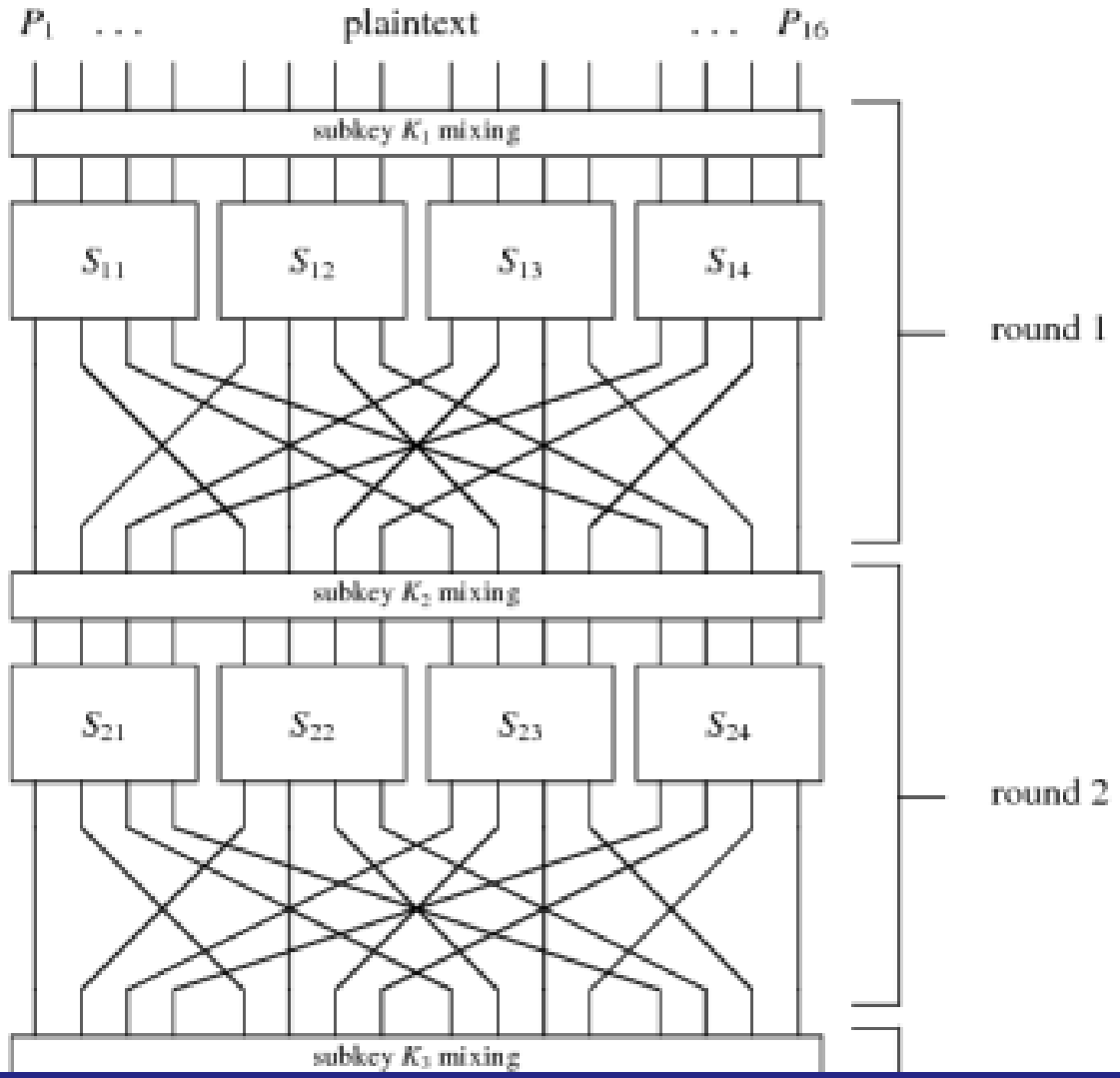
Shannon's principles: confusion and diffusion

Confusion The relationship between key and ciphertext should be complex. Each ciphertext bit should depend on many key bits.

Diffusion The relationship between plaintext and ciphertext should spread information. Changing one plaintext bit should affect many ciphertext bits.

SPNs

- Alternate between substitution (S-boxes) and permutation (P-boxes).
- S-boxes provide confusion: nonlinear mappings.
- P-boxes provide diffusion: rearrange bit positions.
- Repeat for many rounds to achieve security.


P_1 . . . plaintext . . . P_{16}
subkey K_1 mixing
S_{11} S_{12} S_{13} S_{14}
round 1
subkey K_2 mixing
S_{21} S_{22} S_{23} S_{24}
round 2
subkey K_3 mixing

AES at a glance

- Block size: 128 bits.
- Key sizes: 128, 192, or 256 bits.
- Rounds: 10, 12, or 14 (depending on key size).
- Operations per round: SubBytes, ShiftRows, MixColumns, AddRoundKey.

Security level

- n-bit security means $\approx 2^n$ operations to break.
- A 128-bit key gives (at most) 128-bit security.
- For brute force: 2^{128} is astronomically large ($\approx 10^{38}$).

Quantum threat: Grover's algorithm

- Grover's algorithm searches N items in $O(\sqrt{N})$ quantum operations.
- For brute-forcing a key: $2^n \rightarrow 2^{n/2}$ operations.
- A 256-bit key provides only 128-bit security against quantum attack.
- AES-256 is quantum-safe at the 128-bit level.

Block ciphers need a mode of operation

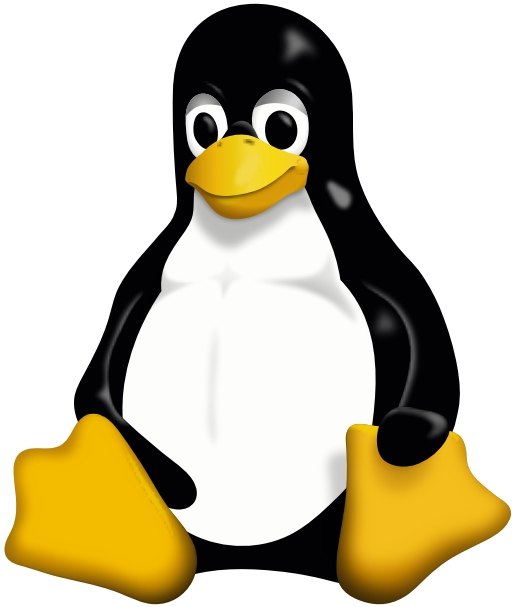
- Real messages are longer than one block.
- A *mode* specifies how to encrypt many blocks.
- A bad mode can leak patterns even if the block cipher is strong.

Exercise

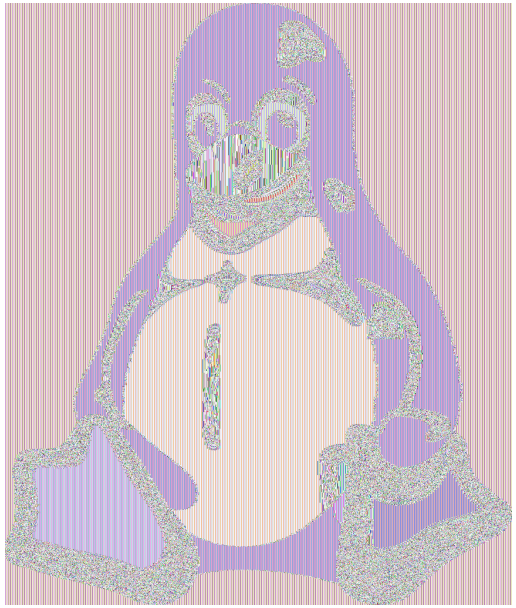
Suppose you encrypt a bitmap image block-by-block.
What would you expect the ciphertext image to look like if you encrypt each block independently with the same key?

Electronic Codebook (ECB) leaks patterns

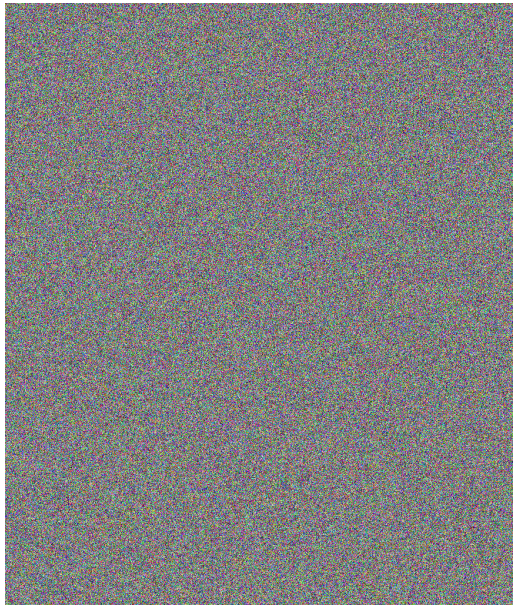
- Each block encrypted independently.
- Equal plaintext blocks $\Rightarrow$ equal ciphertext blocks.
- Structure in the plaintext becomes visible.



(a) Original Tux. Image by: Larry Ewing, Simon Budig, Garrett LeSage.



(b) ECB-encrypted Tux. Image by: RFL890.



(c) CTR-encrypted Tux. Image by: RFL890.

Figure: Tux encrypted using different block modes of operation. When using ECB mode, we can still distinguish Tux. With CTR, the ciphertext looks random.

Cipher Block Chaining (CBC), CTR, Cipher Feedback (CFB), Output Feedback (OFB): what differs?

- How randomness enters (initialization vector (IV)/nonce).
- Whether encryption can be parallelised.
- Error propagation (a bit flip affects how much?).
- Random access: can you decrypt block i without earlier blocks?

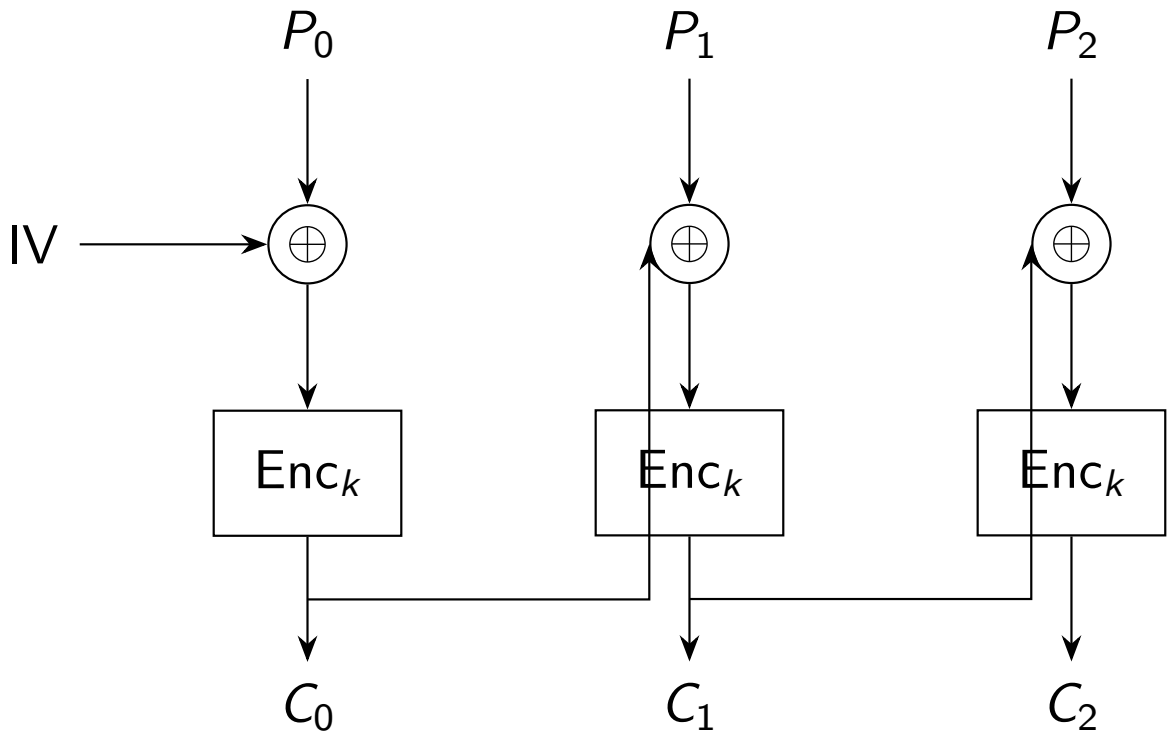


Figure: CBC encryption: each plaintext block is XORed with the previous ciphertext before encryption.

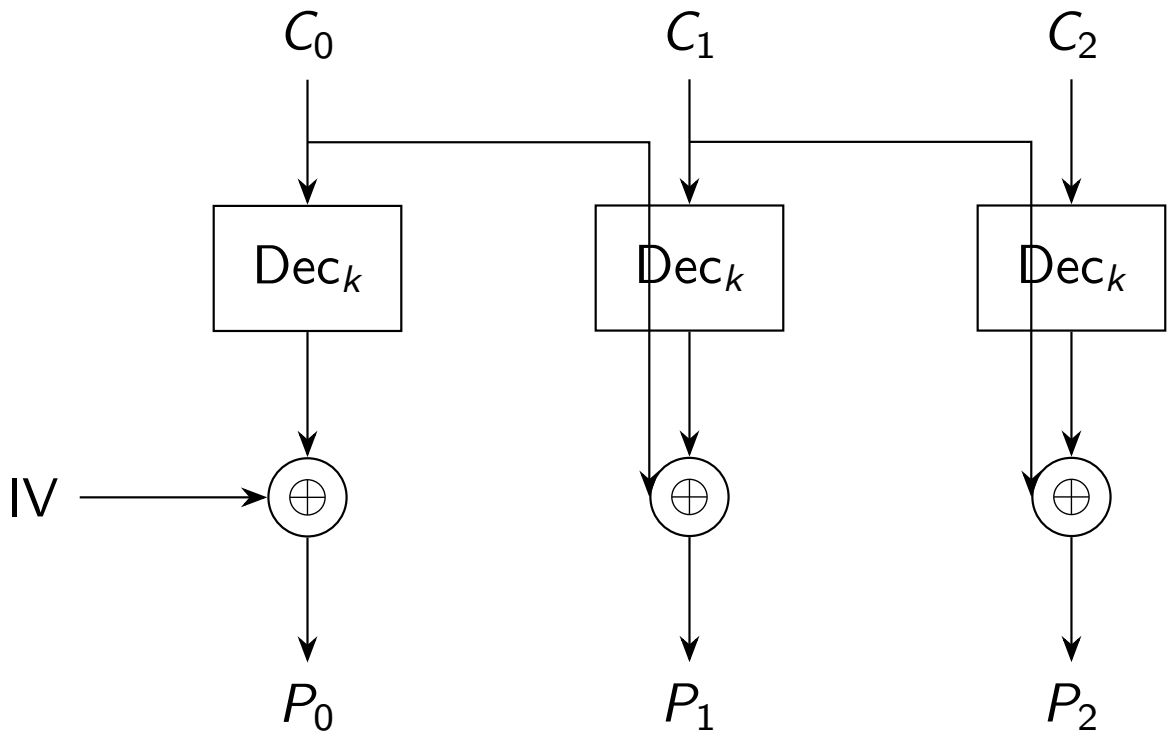


Figure: CBC decryption: each ciphertext block is decrypted, then XORed with the previous ciphertext.

CBC — properties

- Uses a random/unpredictable IV.
- Hides patterns across blocks.
- Encryption is sequential (depends on previous ciphertext).
- Bit flips affect the current block and the next block.

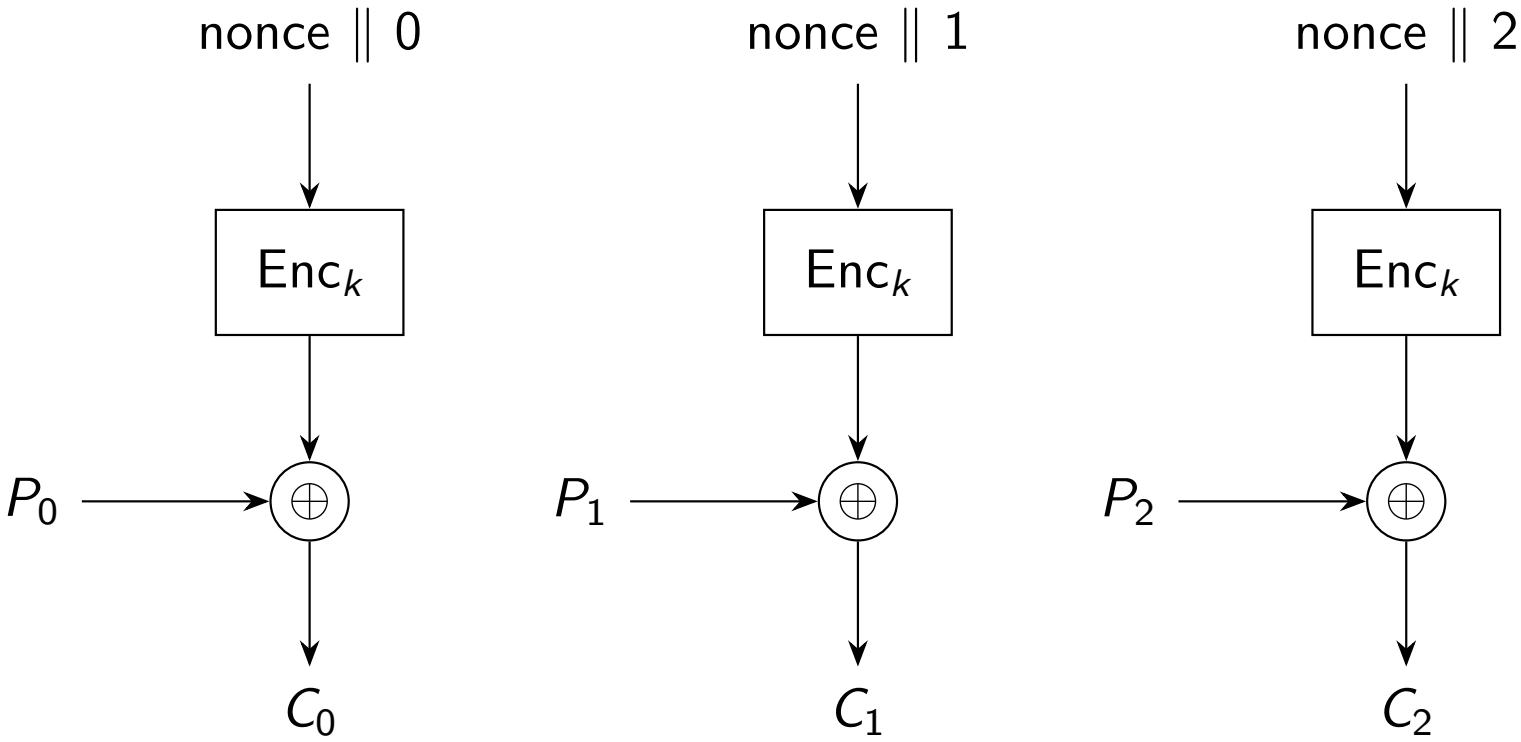


Figure: CTR encryption: the block cipher encrypts counter values to produce a keystream, which is XORed with the plaintext.

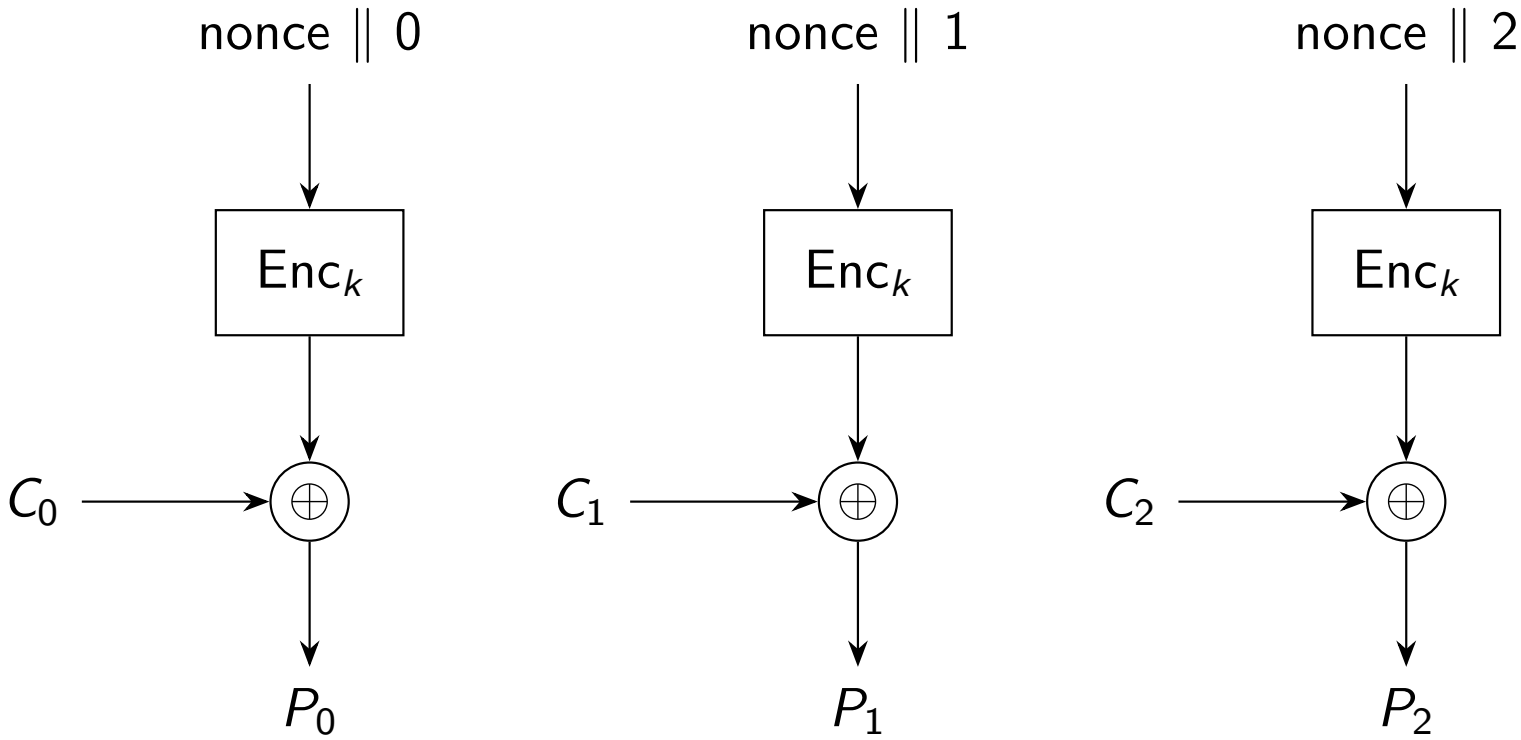


Figure: CTR decryption: identical to encryption—the same keystream is XORed with the ciphertext.

CTR — properties

- Uses a unique nonce (and counter) to create a keystream.
- Parallelisable and supports random access.
- Bit flips in ciphertext flip the corresponding plaintext bits.
- Nonce reuse is catastrophic.

CFB and OFB — properties

- Turn a block cipher into a stream-cipher-like scheme.
- CFB is self-synchronising; OFB is not.
- Both require IV/nonce discipline to avoid reuse.

IV/nonce: what it is and why it matters

- A public value used to randomise encryption.
- Must be unique (and sometimes unpredictable), depending on the mode.
- Reuse can destroy confidentiality.

Exercise

In CTR mode, encryption looks like $c = m \oplus \text{keystream}(k, \text{nonce})$. What do you think happens if the same nonce is reused for two different messages?

CTR nonce reuse: the disaster in one line

- CTR gives: $c = m \oplus \text{keystream}(k, \text{nonce})$.
- If nonce reused: $c_1 \oplus c_2 = m_1 \oplus m_2$.
- This leaks relations between messages and is often enough to recover both.

Exercise

What must be true about IV/nonce values for repeated encryptions under the same key?

Later: authenticated encryption and storage modes

- Authenticated encryption with associated data (AEAD): encryption + integrity in one scheme.
- Galois/Counter Mode (GCM): widely used AEAD mode.
- XEX-based Tweaked-codebook mode with ciphertext Stealing (XTS): disk/storage encryption mode.

Take-away

- Modes matter: ECB is not acceptable for structured data.
- Pick a mode with good properties for your setting (parallelism, random access, error propagation).
- IV/nonce discipline is part of the security definition.

Questions?

Symmetric encryption in one line

- Same key for encryption and decryption.
- Main practical challenge: *key distribution*.