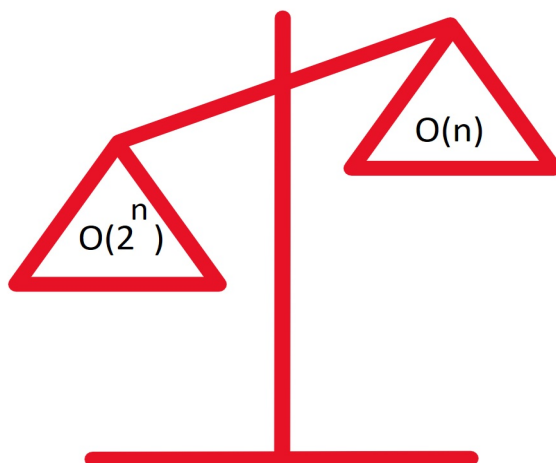


# Criptografía post-cuántica basada en retículos

Lección 1: Minicurso Mar del Plata, Noviembre 2025

## Syllabus

- Criptografía asimétrica y asimétrica
- Principio de Kerckhoff
- Complejidad computacional y Security Level
- El objetivo CPA-IND



# Índice

<b>1.3</b>	<b>Criptografia: confidencialidad y otros usos modernos</b>	<b>3</b>
<b>1.4</b>	<b>Criptografia simetrica y asimetrica : primitivas</b>	<b>4</b>
1.4.1	Symmetric Cryptography and primitives . . . . .	4
1.4.2	Asymmetric Cryptography also called Public Key Cryptography . . . . .	6
<b>1.5</b>	<b>Security Level and Computationally infeasible</b>	<b>8</b>
1.5.1	NIST–Standardized Post-Quantum Digital Signatures . . . . .	9
<b>1.6</b>	<b>CPA-IND and Probabilistic Encryption (Non deterministic)</b>	<b>10</b>
1.6.1	Kerckhoffs’s principle and models COA, KPA, CPA and CCA . . . . .	11
	<b>Referencias</b>	<b>12</b>

## 1.3 Criptografía: confidencialidad y otros usos modernos

The art and science of keeping messages secure is **cryptography**, and it is practiced by **cryptographers**. **Cryptanalysts** are practitioners of **cryptanalysis**, the art and science of breaking ciphertext; that is, seeing through the disguise. The branch of mathematics encompassing both cryptography and cryptanalysis is **cryptology** and its practitioners are **cryptologists**. Modern cryptologists are generally trained in theoretical mathematics **they have to be**.

[Schneier15, Chapter 1, page 1]

### **Authentication, Integrity, and Nonrepudiation**

In addition to providing confidentiality, cryptography is often asked to do other jobs:

- **Authentication.** It should be possible for the receiver of a message to ascertain its origin; an intruder should not be able to masquerade as someone else.
- **Integrity.** It should be possible for the receiver of a message to verify that it has not been modified in transit; an intruder should not be able to substitute a false message for a legitimate one.
- **Nonrepudiation.** A sender should not be able to falsely deny later that he sent a message.

[Schneier15, page 2]

## 1.4 Criptografia simetrica y asimetrica : primitivas

A cryptographic primitive is a fundamental building block used in cryptographic protocols and algorithms. These primitives provide basic functionalities and are combined to implement more complex cryptographic operations.

### 1.4.1 Symmetric Cryptography and primitives

Figure 1.4.1.1: Symmetric Key-Gen

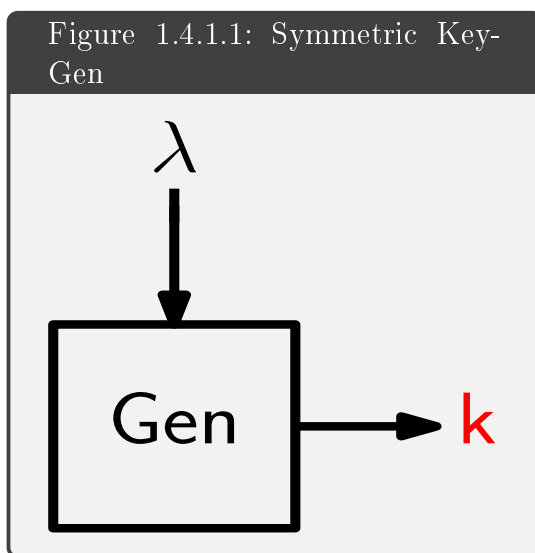
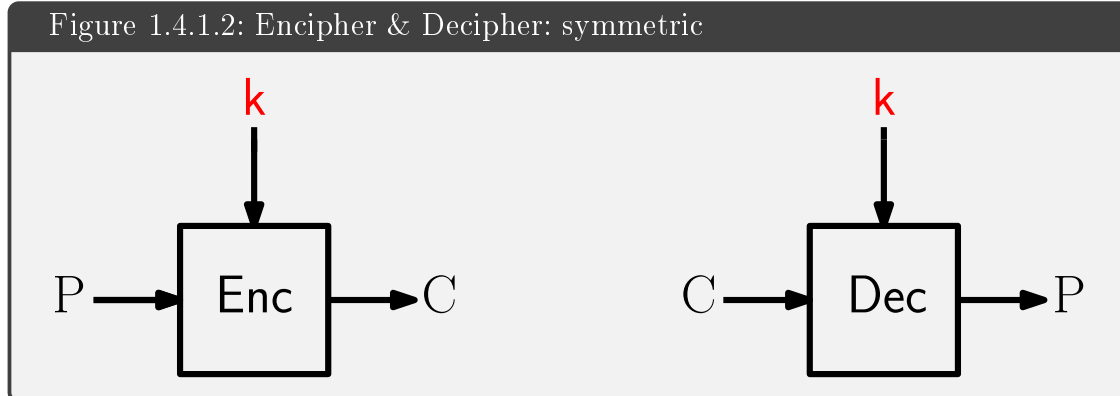


Figure 1.4.1.2: Encipher & Decipher: symmetric



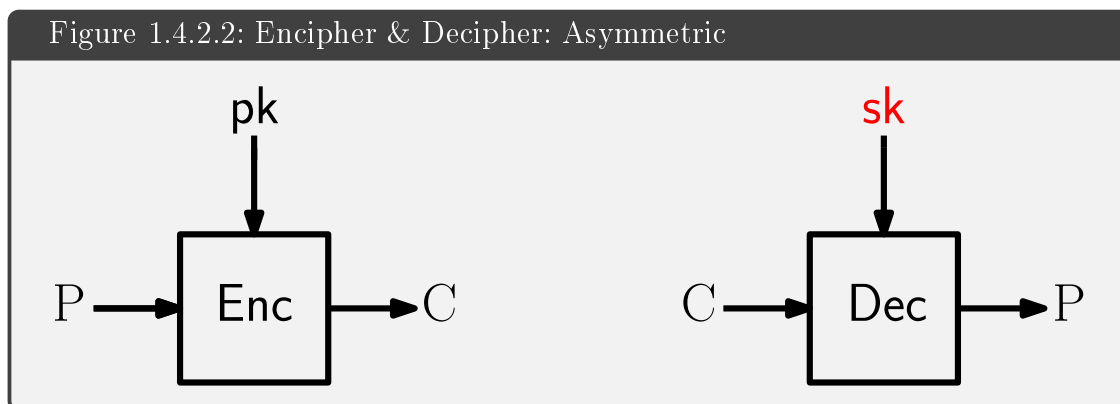
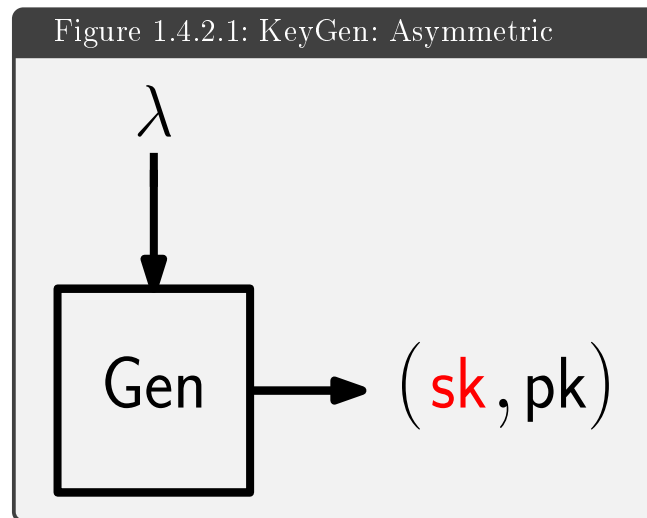
Most important symmetric primitives :

- **Block Ciphers:** Encrypt data in fixed-size blocks..
  - DES (Data Encryption Standard)
  - [AES](#) (Advanced Encryption Standard), [KeyGen NIST SP 800-133](#)
  - 3DES (Triple DES)
  - Blowfish
  - Twofish
  - Serpent
- **Stream Ciphers:** Encrypt data one bit or byte at a time.
  - RC4
  - Salsa20
  - ChaCha20
- **Message Authentication Codes (MACs):** Provide authentication and integrity.
  - HMAC (Hash-based Message Authentication Code)
  - CMAC (Cipher-based Message Authentication Code)
  - GMAC (Galois Message Authentication Code)
- **Key Derivation Functions (KDFs):** Derive one or more secret keys from a secret value.
  - PBKDF2 (Password-Based Key Derivation Function 2)
  - bcrypt
  - scrypt
  - Argon2
- **Random Number Generators:** Generate random numbers for cryptographic use.
  - Cryptographically Secure PRNGs (CSPRNGs)
- **Hash Functions**
  - [SHA-256](#) (Secure Hash Algorithm)
  - SHA-3
  - MD5 (though considered weak)
  - RIPEMD-160

#### NOTE 1.4.1.3

Besides mining blocks Bitcoin also uses SHA-256 for the 24 words of wallets (BIP-39).

### 1.4.2 Asymmetric Cryptography also called Public Key Cryptography



Most important asymmetric primitives :

- **Public Key Cryptography:** Algorithms that use a pair of keys.
  - RSA (Rivest-Shamir-Adleman)
  - ECC (Elliptic Curve Cryptography)
  - DSA (Digital Signature Algorithm)
  - ElGamal Encryption
- **Digital Signatures:** Verify the authenticity and integrity of a message.
  - RSA Signatures
  - ECDSA (Elliptic Curve Digital Signature Algorithm)
  - EdDSA (Edwards-Curve Digital Signature Algorithm)
- **Key Exchange Protocols:** Allow secure key agreement between parties.
  - Diffie-Hellman Key Exchange
  - ECDH (Elliptic Curve Diffie-Hellman)
- **Identity-Based Cryptography:** Uses identity as a public key.
  - Identity-Based Encryption (IBE) systems

## 1.5 Security Level and Computationally infeasible

### Security level $\lambda$

An encryption algorithm has a [security level](#) of  $\lambda$  bits if the best known attack has a computational cost of  $\mathcal{O}(2^\lambda)$  e.g. requires  $\mathcal{O}(2^\lambda)$  steps. This allows us to compare algorithms and is useful when we combine several primitives in a hybrid cryptosystem to understand any weaknesses. The security level is usually written in unary representation as  $1^\lambda$ .

The standard security levels are related to the best known attacks to AES-128, 192, 256, corresponding to NIST levels 1, 3 and 5. Such attacks have complexity 143, 207 and 272 bits, respectively. [NIST : Security \(Evaluation Criteria\)](#)

We will call a task *computationally infeasible* if its cost as measured by either the amount of memory used or the runtime is finite but impossible large.

[DH76, page 646]

To get a clue of the meaning of *computationally infeasible* imagine you are looking for a key  $\mathbf{k}$  of  $m$  bits:

$$\mathbf{k} \in \{0, 1\}^m$$



**1.5.0.1 Brute Force: 30 bits**

```

from timeit import default_timer as timer

#loop with 2**m rounds
m=30
start = timer()
j=0
while j < 2**m:
    # try with key k_j
    print(j)
    j+=1
end = timer()

#print the total time employed to check all keys
print(m, (end - start)/60 , "minutes") # Time in minutes.

```

**Ejercicio 1.5.0.2**

How long it takes for  $m=64$  bits ?

**In most cryptographic functions, the key length is an important security parameter.**

**1.5.1 NIST–Standardized Post-Quantum Digital Signatures**

As of 2024, the National Institute of Standards and Technology (NIST) has standardized the following post-quantum digital signature schemes:

- **CRYSTALS-Dilithium**
  - **Type:** Lattice-based (Module-LWE)
  - **Security Levels:** 1, 3, and 5 (NIST categories)
  - **Status:** Primary standard for general-purpose signatures
  - **Features:** Efficient, well-balanced performance
- **FALCON**
  - **Type:** Lattice-based (NTRU-like, short signatures)
  - **Security Levels:** 1 and 5
  - **Status:** Standardized (for use when smaller signatures are needed)
  - **Features:** Very compact signatures but more complex implementation
- **SPHINCS+**

- **Type:** Hash-based (stateless)
- **Security Levels:** 1, 3, and 5
- **Status:** Standardized as a backup (for long-term security)
- **Features:** Conservative security (based on hash functions), larger signatures

#### NOTE 1.5.1.1

- These algorithms were selected in **July 2022** as part of NIST's PQC Standardization Round 3.
- **CRYSTALS-Dilithium** is the **recommended general-purpose signature scheme**, while **FALCON** is suggested for cases requiring smaller signatures.
- **SPHINCS+** is included as a **hedge against potential future attacks on lattice-based schemes**.

## 1.6 CPA-IND and Probabilistic Encryption (Non deterministic)



Different ciphertexts for the same cleartext  $M$  encrypted with a key  $k$ .

### The CPA indistinguishability experiment $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n)$ :

1. A key  $k$  is generated by running  $\text{Gen}(1^n)$ .
2. The adversary  $\mathcal{A}$  is given input  $1^n$  and oracle access to  $\text{Enc}_k(\cdot)$ , and outputs a pair of messages  $m_0, m_1$  of the same length.
3. A random bit  $b \leftarrow \{0, 1\}$  is chosen, and then a ciphertext  $c \leftarrow \text{Enc}_k(m_b)$  is computed and given to  $\mathcal{A}$ . We call  $c$  the challenge ciphertext.
4. The adversary  $\mathcal{A}$  continues to have oracle access to  $\text{Enc}_k(\cdot)$ , and outputs a bit  $b'$ .
5. The output of the experiment is defined to be 1 if  $b' = b$ , and 0 otherwise. (In case  $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n) = 1$ , we say that  $\mathcal{A}$  succeeded.)

#### 1.6.0.1 CPA-IND secure

The symmetric crypto system  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  is **CPA-IND-secure** (or just CPA-secure) if no adversary  $\mathcal{A}$  can succeed with probability better than  $1/2$ .

### 1.6.1 Kerckhoffs's principle and models COA, KPA, CPA and CCA

#### Kerckhoffs's principle

The enemy knows the system (Shannon's Maxim). This means that security should just depend on the secrecy of the key.

[Kerckhoffs principle](#)

[Attack models](#)

## Referencias

- [Schneier15] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C*, Wiley; 20th Anniversary edition, 2015.
- [DH76] Diffie, W.; Hellman, M. *New directions in cryptography*, (1976). IEEE Transactions on Information Theory. 22 (6): 644-654.
- [QUBIP23] QUBIP: *Transition to Post-Quantum Cryptography*  
QUBIP project is co-funded by the European Union under the Horizon Europe framework programme [grant agreement no. 101119746].  
<https://qubip.eu/>  
<https://github.com/QUBIP>  
[http://www.youtube.com/@qubip\\_eu](http://www.youtube.com/@qubip_eu)