**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA**

**Escuela de Ingeniería Informática**

---

# Práctica 3:
# Cifrado Asimétrico

---

**Asignatura:** Seguridad de la Información

**Curso:** 4° de Grado en Ingeniería Informática

**Autor:**

Nicolás Rey Alonso

nicolas.rey101@alu.ulpgc.es

**Fecha:** Febrero 2026

*"La criptografía es la ciencia y el arte de proteger la información."*

# Capítulo 1

# Demostración de la Peligrosidad del Modo ECB

## 1.1. Fundamento teórico: El problema del modo ECB

El modo ECB (*Electronic Codebook*) es uno de los modos de operación más simples e inseguros. Su funcionamiento es:

$$C_i = E(K, P_i) \tag{1.1}$$

Donde cada bloque de texto plano $P_i$ se cifra independientemente con la misma clave $K$.

### 1.1.1. Problemas de seguridad

1. **Determinismo:** Bloques idénticos producen criptogramas idénticos
2. **Revelación de patrones:** Patrones en el texto plano se revelan en el cifrado
3. **Análisis de frecuencia:** Posible análisis estadístico
4. **Imagen de Tux:** El ejemplo clásico demuestra visualmente estas debilidades

## 1.2. Experimento: Cifrado de imágenes

### 1.2.1. Procedimiento

Se utilizó una imagen de colores sólidos (PNG de 20-50 KB) para demostrar los problemas de ECB:

1. Conversión de la imagen a formato PGM
2. Separación de cabecera y datos binarios
3. Cifrado de los datos con AES-256-ECB
4. Reconstrucción de la imagen cifrada
5. Conversión de vuelta al formato original
6. Comparación con modo CBC

### 1.2.2. Comandos utilizados

Listing 1.1: Cifrado ECB de imágenes

```
# Convertir a PGM
convert imagen.png imagen.pgm

# Separar cabecera y datos
head -n 3 imagen.pgm > cabecera.txt
tail -n +4 imagen.pgm > datos.bin

# Cifrado ECB (INSEGURO)
openssl enc -aes-256-ecb -in datos.bin -out datos_ecb.bin \
    -K 0123456789ABCDEF... -nosalt

# Reconstruir y convertir
cat cabecera.txt datos_ecb.bin > imagen_cifrada.pgm
convert imagen_cifrada.pgm imagen_cifrada.png
```

### 1.2.3. Comparación de modos

Cuadro 1.1: Comparación de modos de operación

| Modo | Resultado | Seguridad |
|---|---|---|
| ECB | *Se distinguen patrones de la imagen original* | Crítica |
| CBC | *Imagen completamente aleatoria* | Segura |

## 1.3. Resultados obtenidos

### 1.3.1. Imágenes generadas

Se generaron las siguientes imágenes:

- **Original:** Imagen de colores sólidos clara
- **ECB con AES-256:** Los colores originales son vagamente visibles
- **ECB con 3DES:** Patrones más pronunciados debido a bloques de 64 bits
- **CBC con AES-256:** Imagen completamente aleatoria

## 1.4. Conclusiones sobre seguridad

- **NUNCA usar ECB:** Incluso con claves fuertes, revela patrones
- **Usar CBC, CTR, GCM:** Estos modos son seguros
- **Tamaño de bloque:** Bloques mayores (128 bits) ocultan mejor los patrones
- **IV aleatorio:** Esencial en modos que requieren IV

# Capítulo 2

# Conclusiones

## 2.1. Resumen de aprendizajes

A través de esta práctica se han consolidado los siguientes conceptos:

1. **Algoritmos de resumen:** Importancia de la sensibilidad y tamaño del hash
2. **Cifrado simétrico:** Uso correcto de algoritmos y modos de operación
3. **Derivación de claves:** PBKDF2 como estándar seguro
4. **Modos seguros vs. inseguros:** Errores comunes en criptografía

## 2.2. Recomendaciones de seguridad

Para aplicaciones reales:

1. **Resúmenes:** Usar SHA-256 o superior (SHA-3)
2. **Cifrado simétrico:** Preferir AES con modo CBC, CTR o GCM
3. **Derivación de claves:** PBKDF2 con mínimo 100,000 iteraciones
4. **Sal:** Mínimo 16 bytes de sal aleatoria
5. **IV:** Aleatorio y único para cada cifrado en modos que lo requieran

## 2.3. Problemas encontrados y soluciones

- **Provider Legacy:** Fue necesario activarlo para usar algoritmos antiguos (MD5, RC4)

- **Formatos binarios:** Importancia de usar `-binary` en OpenSSL
- **Imágenes comprimidas:** No funcionan bien con el experimento ECB

## 2.4. Trabajos futuros

1. Estudiar cifrado asimétrico (RSA, ECDSA)
2. Implementar funciones hash criptográficas personalizadas
3. Analizar vulnerabilidades de padding (Padding Oracle)
4. Estudiar Side-channel attacks en criptografía

# Bibliografía


[1] OpenSSL Documentation, *https://www.openssl.org/docs/man3.2/*, 2024.

[2] RSA Laboratories, *PKCS #5: Password-Based Cryptography Specification*, RFC 2898, 2000.

[3] NIST Federal Information Processing Standards Publication 197, *Specification for the Advanced Encryption Standard (AES)*, 2001.

[4] Kelm, R., Turan, M. S., *PBKDF2 Implementation in OpenSSL*, 2023.

[5] Wikipedia, *Block cipher mode of operation*, https://en.wikipedia.org/wiki/Block_cipher_mode_of_operat 2024.

[6] Wikipedia, *Electronic Codebook (ECB) - Tux the Linux Penguin*, https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#ECB, 2024.

[7] Rivest, R., *The MD5 Message-Digest Algorithm*, RFC 1321, 1992.

[8] FIPs 180-4: Secure Hash Standard, National Institute of Standards and Technology, 2015.

# Apéndice A

# Comandos OpenSSL Utilizados

## A.1. Comandos de hash

Listing A.1: Comandos para resúmenes

```
# Listar algoritmos disponibles
openssl dgst -list

# Diferentes formatos
openssl dgst -md5 archivo.txt
openssl dgst -sha256 -c archivo.txt          # Con separadores
openssl dgst -sha256 -binary archivo.txt     # Binario
```

## A.2. Comandos de cifrado

Listing A.2: Comandos para cifrado simétrico

```
# Listar cifradores disponibles
openssl enc -list

# Cifrado básico
openssl enc -aes-256-cbc -in archivo.txt -out archivo.bin \
    -pass pass:"contraseña"

# Con información de derivación
openssl enc -aes-256-cbc -in archivo.txt -out archivo.bin \
    -pass pass:"contraseña" -pbkdf2 -p

```

```
# Descifrado
openssl enc -aes-256-cbc -d -in archivo.bin -out archivo.txt \
    -pass pass:"contraseña" -pbkdf2
```

## A.3. Comandos para manipulación de archivos

Listing A.3: Comandos útiles

```
# Ver información del archivo
file archivo.bin
hexdump -C archivo.bin | head -n 5

# Eliminar primeros N bytes
dd if=entrada.bin of=salida.bin bs=1 skip=16

# Conocer tamaño
wc -c archivo.bin
```

# Apéndice B

# Configuración de ImageMagick

## B.1. Instalación

En macOS:

```
brew install imagemagick
brew install ghostscript
```

En Linux:

```
apt install imagemagick
apt install ghostscript
```

## B.2. Conversiones útiles

Listing B.1: Conversiones de imagen

```
# PNG a PGM
convert imagen.png imagen.pgm

# PGM a PNG
convert imagen.pgm imagen.png

# Crear imagen de prueba
convert -size 200x200 xc:red -fill blue \
    -draw 'rectangle 0,0 100,200' test.png
```