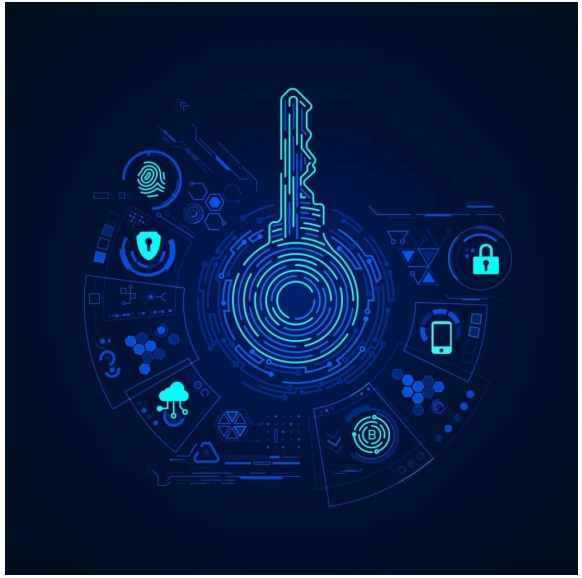
# Homomorphic encryption

Clara Cidrás Fernández - Jofre Coll Vila - Cristian González Trillo

# 1) Homomorphic encryption, ¿qué es?

- **Técnica criptográfica que permite operaciones sobre datos cifrados sin necesidad de descifrarlos**
- Ofrece **seguridad** y **privacidad**
- **Origen en los años 80**, en el 2009 primer esquema totalmente homomórfico
- Supuso un **gran avance para diversos sectores**



# 1) Homomorphic encryption, ¿qué es?

- Permite **realizar operaciones** (suma, multiplicación, etc.) **directamente sobre datos cifrados**
- **Los resultados**, una vez descifrados, **son los mismos que si se hubieran operado sobre los datos originales**.
- **Protege la privacidad**: los datos nunca se ven en claro durante el proceso.
- **Ideal para procesar información sensible sin exponerla**.

# 2) Tipos de cifrado homomórfico

- **PHE** (Partially Homomorphic Encryption)

  Permite **una sola operación** (suma o multiplicación) sobre los datos cifrados.

  **Limitación:** Sólo una operación, **no adecuada para cálculos complejos**.

- **SHE** (Somewhat Homomorphic Encryption)

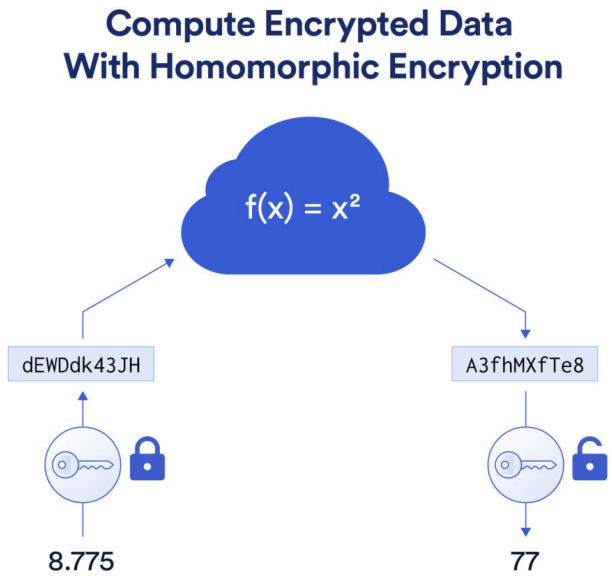
  Permite **sumas y multiplicaciones**, pero solo un número limitado debido al "ruido".

  **Limitación:** Limitado por el crecimiento del **ruido**, requiere planificación.

- **FHE** (Fully Homomorphic Encryption)

  Permite **cualquier operación** sobre datos cifrados sin necesidad de descifrarlos.

  **Limitación:** **Alta complejidad computacional**, mayor consumo de recursos y necesidad de hardware especializado.



# PHE fórmula

### Generación de claves

- Elegimos dos números primos grandes $p$ y $q$.
- Calculamos $n = p \times q$.
- Calculamos $\lambda = \operatorname{lcm}(p-1, q-1)$ (mínimo común múltiplo).
- Elegimos un número $g \in \mathbb{Z}_{n^2}^*$ tal que $g$ genera un subgrupo de orden múltiplo de $n$.
- Calculamos:

$$\mu = \left(L\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n$$

donde $L(x) = \frac{x-1}{n}$.

- La clave pública es $(n, g)$.
- La clave privada es $(\lambda, \mu)$.

# PHE fórmula

## Cifrado de un mensaje

Para cifrar un mensaje $m \in \mathbb{Z}_n$:

- Elegimos un valor aleatorio $r \in \mathbb{Z}_n^*$.
- Calculamos el cifrado:

$$c = g^m \times r^n \bmod n^2$$

---

## Descifrado

Para descifrar un cifrado $c$:

$$m = L\left(c^\lambda \bmod n^2\right) \times \mu \bmod n$$

# PHE fórmula

## Propiedad homomórfica aditiva

El esquema de Paillier cumple que:

- Si tenemos dos mensajes cifrados $c_1 = \text{Enc}(m_1)$ y $c_2 = \text{Enc}(m_2)$,
- Entonces:

$$c_1 \times c_2 \bmod n^2 = \text{Enc}(m_1 + m_2 \bmod n)$$

Es decir, el producto de dos cifrados equivale a un cifrado de la suma de los mensajes originales.

# Ejemplo práctico

Supongamos:

- $p = 3, q = 5$.
- Entonces $n = 3 \times 5 = 15$.
- $n^2 = 225$.

Calculamos:

- $\lambda = \text{lcm}(2, 4) = 4$.

Elegimos $g = 16$.

Calculamos:

- $g^\lambda \bmod n^2 = 16^4 \bmod 225 = 1$.
- $L(1) = \frac{1-1}{15} = 0$.
- (Para este ejemplo, ajustamos $g$ para evitar $L(1) = 0$; usamos $g = 31$).

Usamos:

- $g = 31$.

Calculamos:

- $g^\lambda \bmod n^2 = 31^4 \bmod 225 = 61$.
- $L(61) = \frac{61-1}{15} = 4$.
- $\mu = 4^{-1} \bmod 15 = 4$ (ya que $4 \times 4 = 16 \equiv 1 \mod 15$).

# Ejemplo práctico

Supongamos:

- $p = 3, q = 5$.
- Entonces $n = 3 \times 5 = 15$.
- $n^2 = 225$.

Calculamos:

- $\lambda = \text{lcm}(2, 4) = 4$.

Elegimos $g = 16$.

Calculamos:

- $g^\lambda \bmod n^2 = 16^4 \bmod 225 = 1$.
- $L(1) = \frac{1-1}{15} = 0$.
- (Para este ejemplo, ajustamos $g$ para evitar $L(1) = 0$; usamos $g = 31$).

Usamos:

- $g = 31$.

Calculamos:

- $g^\lambda \bmod n^2 = 31^4 \bmod 225 = 61$.
- $L(61) = \frac{61-1}{15} = 4$.
- $\mu = 4^{-1} \bmod 15 = 4$ (ya que $4 \times 4 = 16 \equiv 1 \mod 15$).

# Ejemplo práctico

### Cifrado

Queremos cifrar:

- $m_1 = 7$
- $m_2 = 8$

Elegimos valores aleatorios:

- $r_1 = 2, r_2 = 3$

Calculamos:

$$c_1 = 31^7 \times 2^{15} \bmod 225$$

$$c_2 = 31^8 \times 3^{15} \bmod 225$$

- $c_1 = 34$
- $c_2 = 57$

# Ejemplo práctico

### Operación homomórfica

Multiplicamos los cifrados:

$$c' = c_1 \times c_2 \bmod 225$$

$$c' = 34 \times 57 \bmod 225 = 138 \bmod 225 = 138$$

El cifrado $c'$ corresponde a $m_1 + m_2 = 7 + 8 = 15$.

# Ejemplo práctico

### Descifrado

Aplicamos el descifrado sobre $c'$:

$$m' = L\left(c'^{\lambda} \bmod 225\right) \times \mu \bmod 15$$

$$m' = L\left(138^{4} \bmod 225\right) \times 4 \bmod 15$$

(Se calculan las potencias y operaciones módulo)

Resultado:

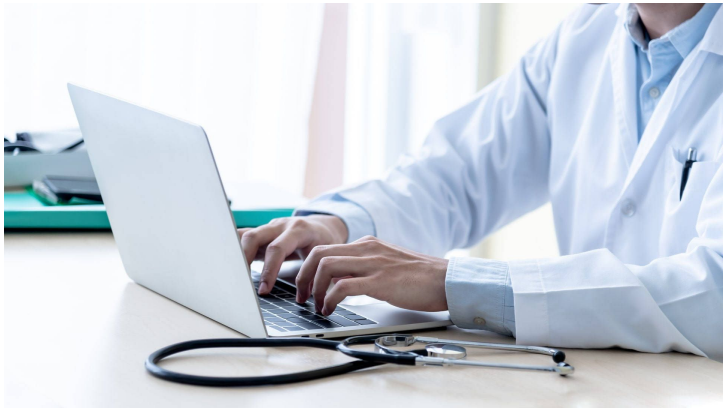
- Recuperamos $m' = 15$.

# 3) Aplicaciones

1. Sanidad
2. Finanzas
3. Cloud computing
4. IA y machine learning
5. Gobiernos y defensa

# 3.1) Aplicaciones - SANIDAD

- Análisis de historiales médicos sin descifrar datos.
- Facilita la colaboración entre hospitales y centros de investigación.
- Cumple normativas estrictas de privacidad como GDPR o HIPAA.
- Protege datos sensibles frente a ciberataques.



# 3.2) Aplicaciones - FINANZAS

- Auditorías y evaluaciones de riesgo sobre datos cifrados
- Prevención de fraudes sin comprometer la privacidad de los clientes
- Análisis de grandes volúmenes de transacciones de manera segura
- Mejora la confianza entre instituciones financieras y clientes

# 3.3) Aplicaciones - CLOUD COMPUTING

- El proveedor de la nube nunca accede a la información real
- Protege los datos frente a brechas de seguridad o accesos no autorizados
- Facilita el cumplimiento de normativas de protección de datos

# 3.4) Aplicaciones - IA Y MACHINE LEARNING

- Modelos de IA pueden entrenarse sin acceso a datos originales
- Fundamental para proteger datos sensibles en sectores como salud o banca
- Permite aprendizaje colaborativo entre entidades sin compartir datos
- Ayuda a cumplir regulaciones de protección de datos

# 3.5) Aplicaciones - GOBIERNOS Y DEFENSA

- Intercambio seguro de inteligencia entre agencias o países aliados
- Reducción del riesgo de fugas de datos críticos
- Protección de información estratégica en entornos de alta amenaza

# 4) Problemas y retos

- **Sobrecarga computacional**

  El uso de cifrado homomórfico, especialmente el FHE, requiere una **gran cantidad de recursos computacionales,** significativamente mayor que las técnicas tradicionales.

- **Gestión del ruido**

  Las operaciones en datos cifrados introducen "ruido", que **afecta la precisión del cifrado y su capacidad para ser descifrado correctamente**.

- **Requerimientos de hardware**

  Las **CPUs tradicionales no son suficientemente rápidas** para procesar operaciones homomórficas de manera eficiente. Aunque los aceleradores por GPU y las investigaciones están avanzando, aún **es necesario un hardware especializado**.

- **Complejidad de distribución**

  La **distribución segura de claves criptográficas es un reto clave** en sistemas con cifrado homomórfico. Dividir las claves en partes y requerir consenso para operaciones agrega complejidad.

# 5) Estandarización

Actualmente, la estandarización de la encriptación homomórfica se encuentra en contínuo desarrollo. El mayor esfuerzo de estandarización lo está llevando a cabo la organización HomomorphicEncryption.org, impulsada por la comunidad, el sector académico y el industrial.

Esta organización está desarrollando el HES (Homomorphic Encryption Standard), que fue el resultado de la “Second Standardization Workshop” en 2018.

En 2019 redactaron un DRAFT para actualizar el HES, el cual aún se encuentra pendiente de votación.

La última reunión de esta organización ha tenido lugar el 23 de Marzo de 2025. Lo que demuestra que sigue activa y continúa desarrollando el proyecto.

# 6) Futuro de la encriptación homomórfica

La encriptación homomórfica debe **pasar de un aspecto teórico a uno más práctico**. Deben **continuar los avances en estandarización**, rendimiento y el **desarrollo de bibliotecas y ecosistemas que faciliten el uso** de la encriptación homomórfica.

Se deben **marcar como objetivos** aquellos **problemas y retos que se han expuesto en este documento**, tales como la sobrecarga computacional, ya que solventándolos, **la encriptación homomórfica puede brindar un gran abanico de posibilidades a los sectores industriales que presentan altos requerimientos de seguridad y privacidad**.

# 7) Preguntas debate

- **Dado el alto coste computacional del FHE, ¿en qué tipos de aplicaciones creéis que es más realista o prioritario su uso en el corto plazo?**

- **En escenarios donde la privacidad es crítica (como salud o finanzas), ¿creéis que merece la pena sacrificar eficiencia por privacidad, o debemos encontrar un equilibrio?**

- **¿Deberían leyes como el GDPR fomentar o incluso exigir el uso de cifrado homomórfico en ciertos contextos?**

- **¿Cómo cambiaría el negocio de la nube si los datos se procesaran siempre cifrados?**

- **¿Puede el cifrado homomórfico ser usado con fines maliciosos, por ejemplo para ocultar actividades ilegales? ¿Cómo podríamos evitarlo?**