



# Algoritmos de encriptación

Hernández Gutiérrez Carlos Mario

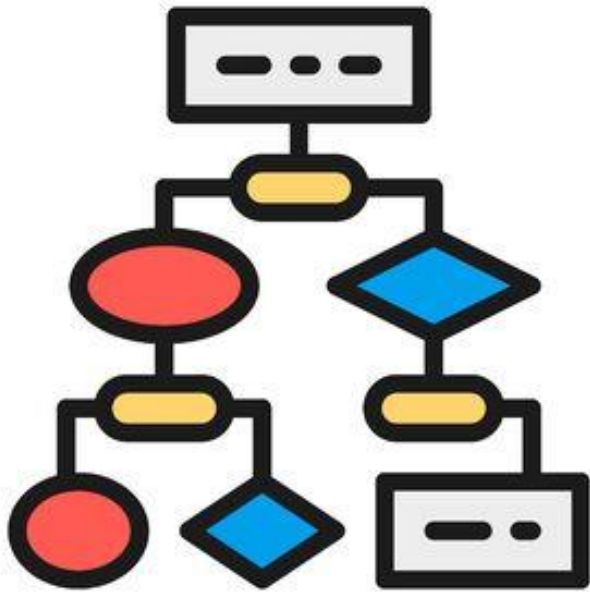
Sistemas Operativos



## Temas por abordar

- ¿Qué es un algoritmo de encriptación?
- Tipos de algoritmos
- Algoritmos simétricos y sus aplicaciones.
- Algoritmos asimétricos y sus aplicaciones.

# ¿Qué es un algoritmo?



Un algoritmo es un conjunto de instrucciones finitas que se encuentran ordenadas y sirven para solucionar un problema en específico.

# ¿Qué es la encriptación?

Es el proceso de transformar información legible, en un formato ilegible mediante el uso de ciertas técnicas con el objetivo de proteger la información.



# ¿Qué es un algoritmo de encriptación?



Es el encargado de codificar la información mediante reglas y modelos matemáticos a fin de tener la información segura.



El mensaje antes de encriptar se llama texto plano y una vez encriptado se le llama texto cifrado.



La información cifrada será desbloqueada por medio de una clave criptográfica



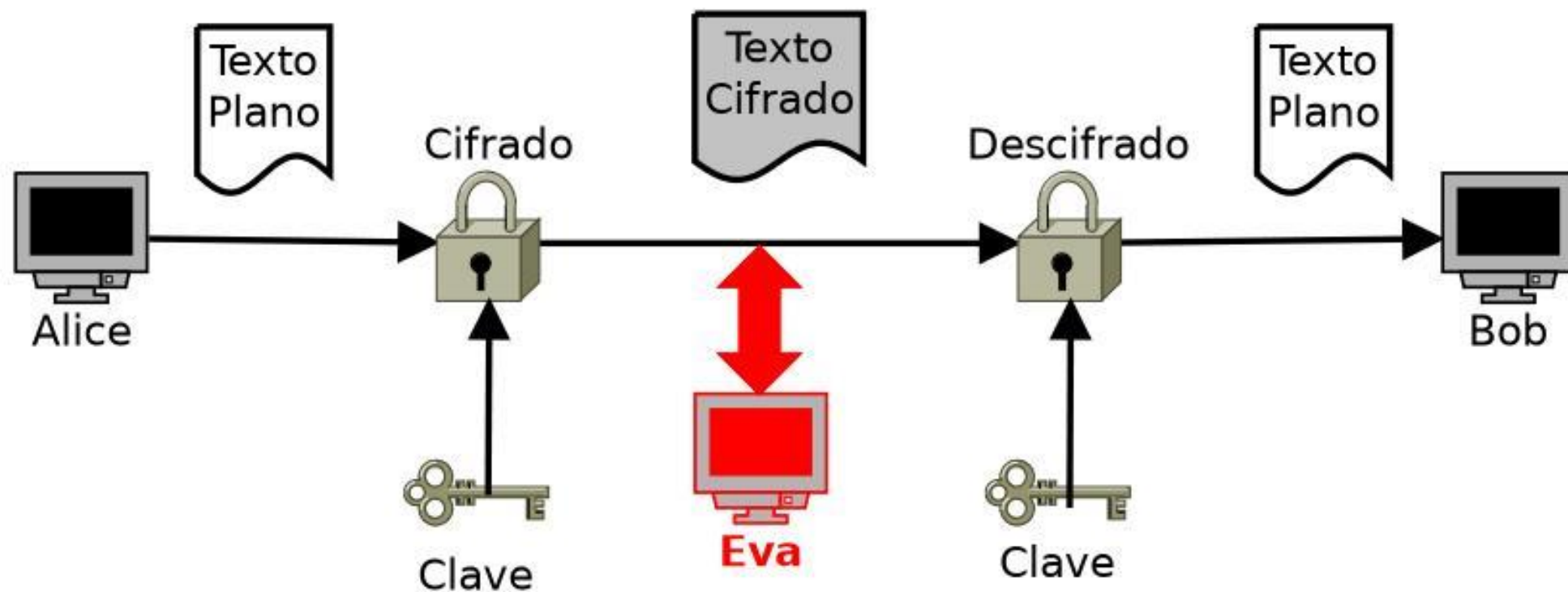
# Clave criptográfica

"Hello" +  = "KZ0KVey8l1c="

Esta clave es una cadena de caracteres que se utiliza para alterar los datos de forma que parezcan aleatorios. Esta clave debe ser privada para que ninguna otra persona pueda acceder a la información.



# Como funciona



# Vulnerabilidades

- Existen diferentes métodos
- Fuerza Bruta
- Ingeniería social
- Ataques de diccionario





# ¿En dónde se encuentran estos algoritmos?

---

- Correo electrónico seguro.
- Seguridad en las transacciones bancarias
- Almacenamiento en la nube
- Comunicaciones privadas
- VPN (redes privadas virtuales)
- Puedes cifrar todo tu disco duro e incluso realizar llamadas por voz cifradas



# Algoritmos simetricos



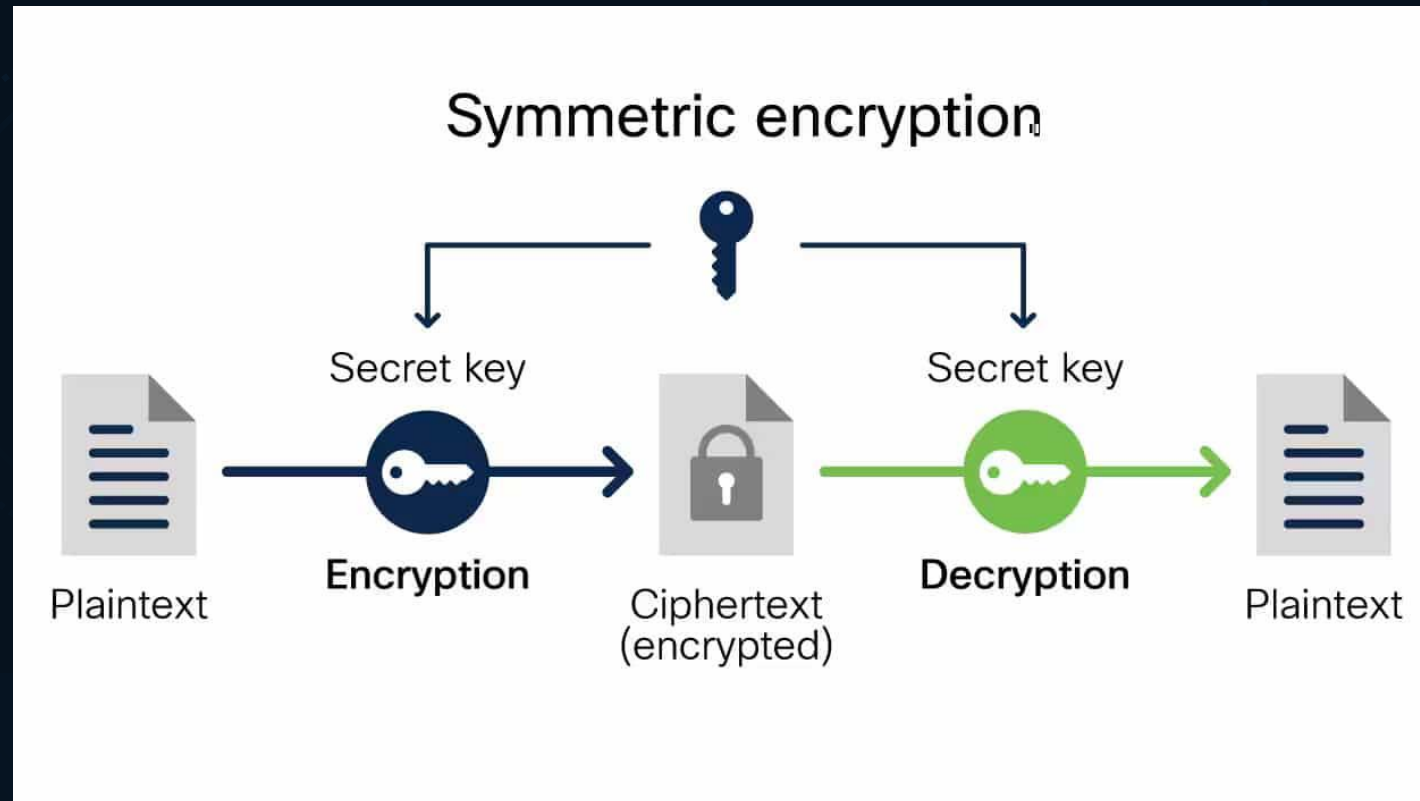
# ¿Qué es el algoritmo simétrico?

Se le conoce también como cifrado de clave privada o compartida.

La clave que se usa en la codificación es la misma que se utiliza en la decodificación

La desventaja al tener solo una llave es que, si una persona no autorizada obtiene la clave, podrá descryptar los mensajes.

Los cifrados de clave simétricos son menos costosos y no requieren tanta potencia informática para encriptar y descryptar contenido.



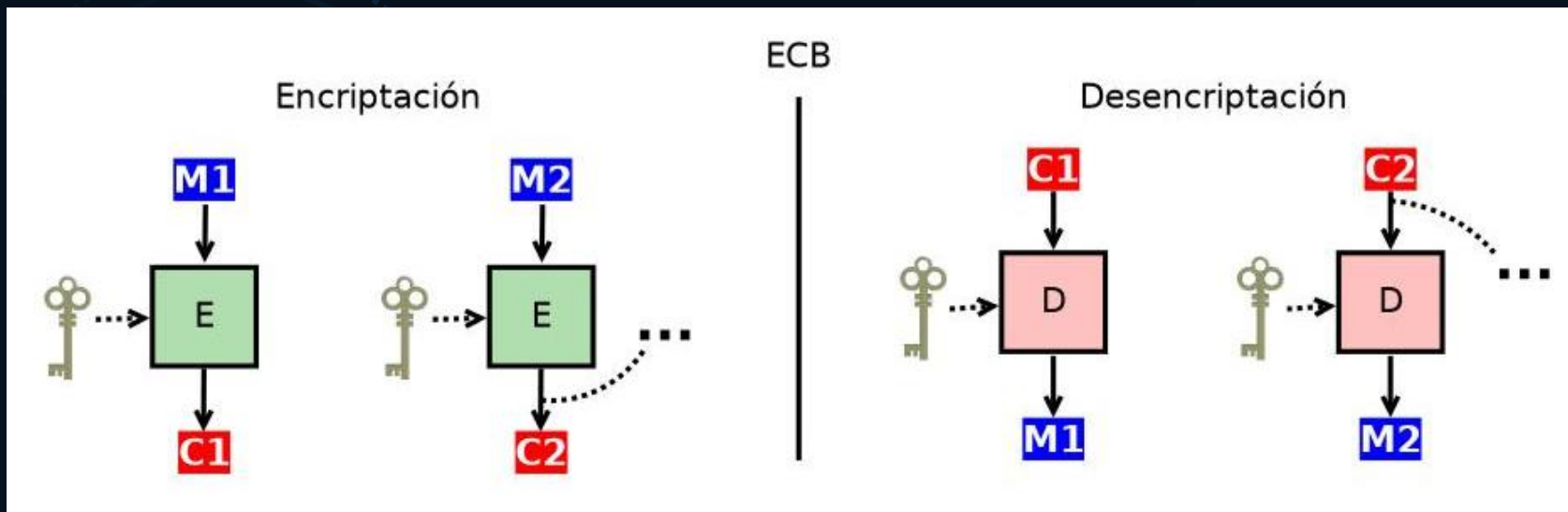
# Algoritmo simétrico de bloques

Toman bloques de tamaño fijo del texto en plano y producen un bloque de tamaño fijo de texto cifrado.

Para la asignación de bloques se realiza sustituciones y permutaciones en el texto plano hasta obtener el texto cifrado.

Entre más grande sea el bloque mejor será su cifrado.

Algoritmo simétrico de bloques: AES (Advanced Encryption Standard)



# AES (Advanced Encryption Standard)

AES por sus siglas en ingles Advanced Encryption Standard es un algoritmo de cifrado por bloques su tamaño fijo de bloque es de 128 bits.

Este algoritmo trabaja con 3 diferentes longitudes de claves:

- Para una clave de 128 bits se aplican 10 rondas de cifrado y nos genera una combinación potencial de  $3.4 \times 10^{38}$  combinaciones.
- Para una clave de 192 bits se aplican 12 rondas y nos genera una combinación potencial de  $6.2 \times 10^{57}$  combinaciones.
- Para una clave de 256 bits las rondas aplicadas son 14 y nos genera una combinación de  $1.1 \times 10^{77}$  combinaciones.



# Como funciona este algoritmo

|    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|
| 03 | 1F | 2A | 1E | 3F | 01 | 7A | 21 | 04 | CF | 7A | 1C |
|----|----|----|----|----|----|----|----|----|----|----|----|

Bloque de 128 bits

|    |    |    |    |
|----|----|----|----|
| AE | 1E | 21 | 1C |
| 03 | 3F | 04 | 33 |
| 1F | 1  | CF | 11 |
| 2A | 7A | 7A | 27 |

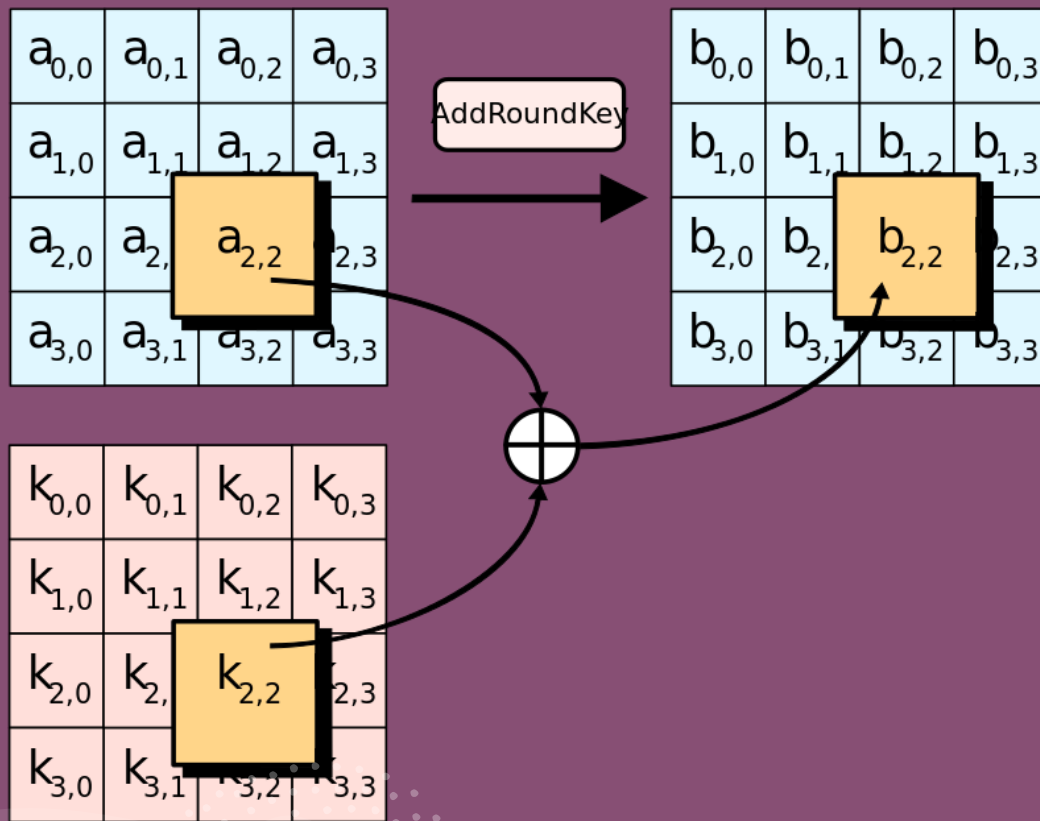
Matriz de Estado

Tenemos nuestro bloque de 128 bits.

Se genera una matriz de estado 4×4, el texto se introducen palabras de 32 bits que se escriben de arriba hacia abajo y de izquierda a derecha.

- Y se ejecutan 3 pasos importantes
- **Paso inicial**
- **Rondas intermedias**
- **Ronda final.**

# Paso inicial

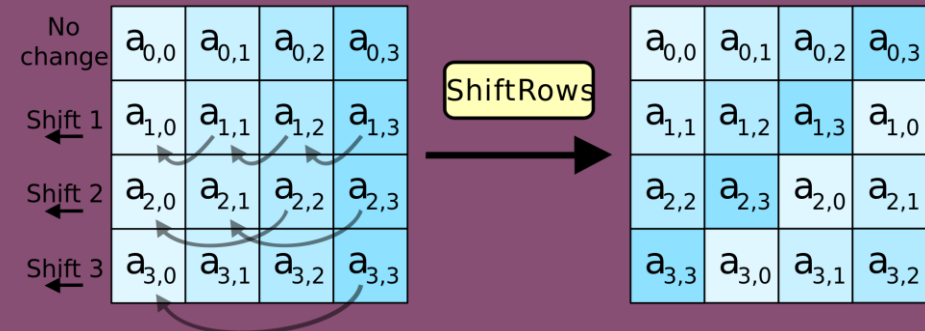
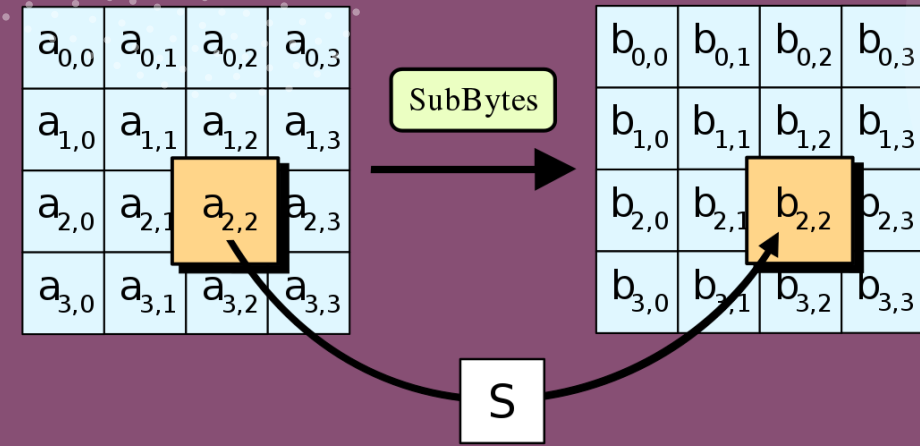


- Comienza con la función llamada **AddRoundKey** donde se hace una operación de X-or exclusivo entre el mensaje a cifrar y la clave inicial.



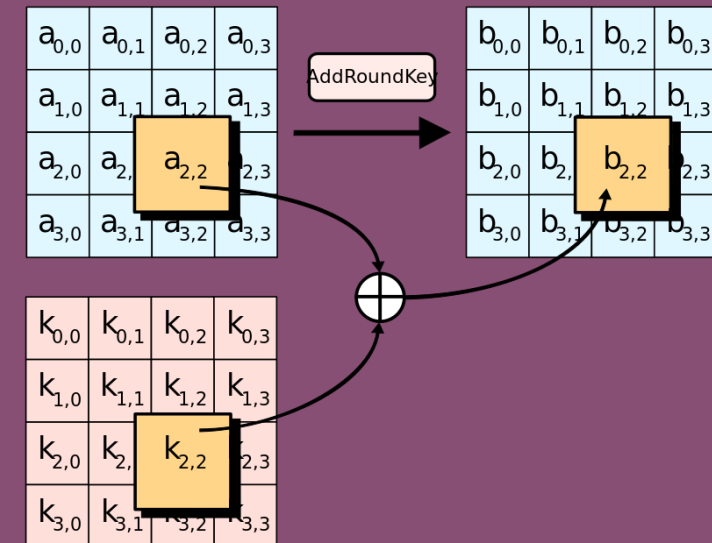
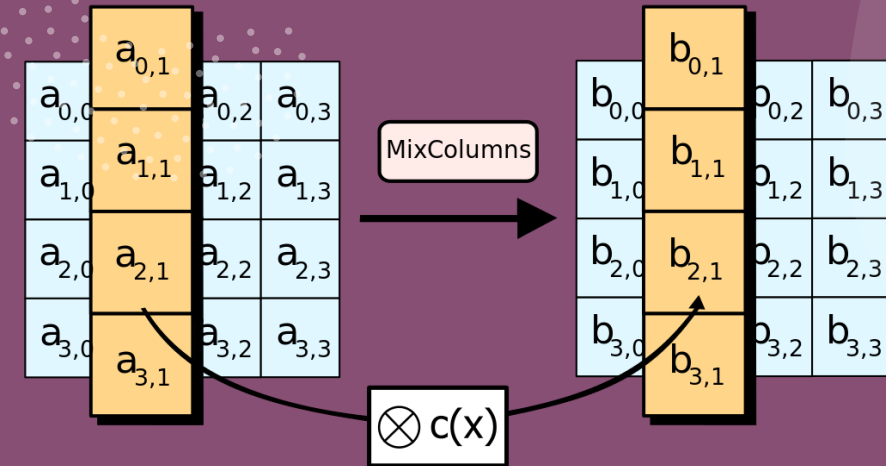
# Rondas intermedias

- **SubBytes:** : Se sustituyen los bytes del bloque de datos por otros bytes, según una tabla de sustitución fija conocida como S-box.
- **ShiftRows:** : Todas las filas del texto se desplazan una posición

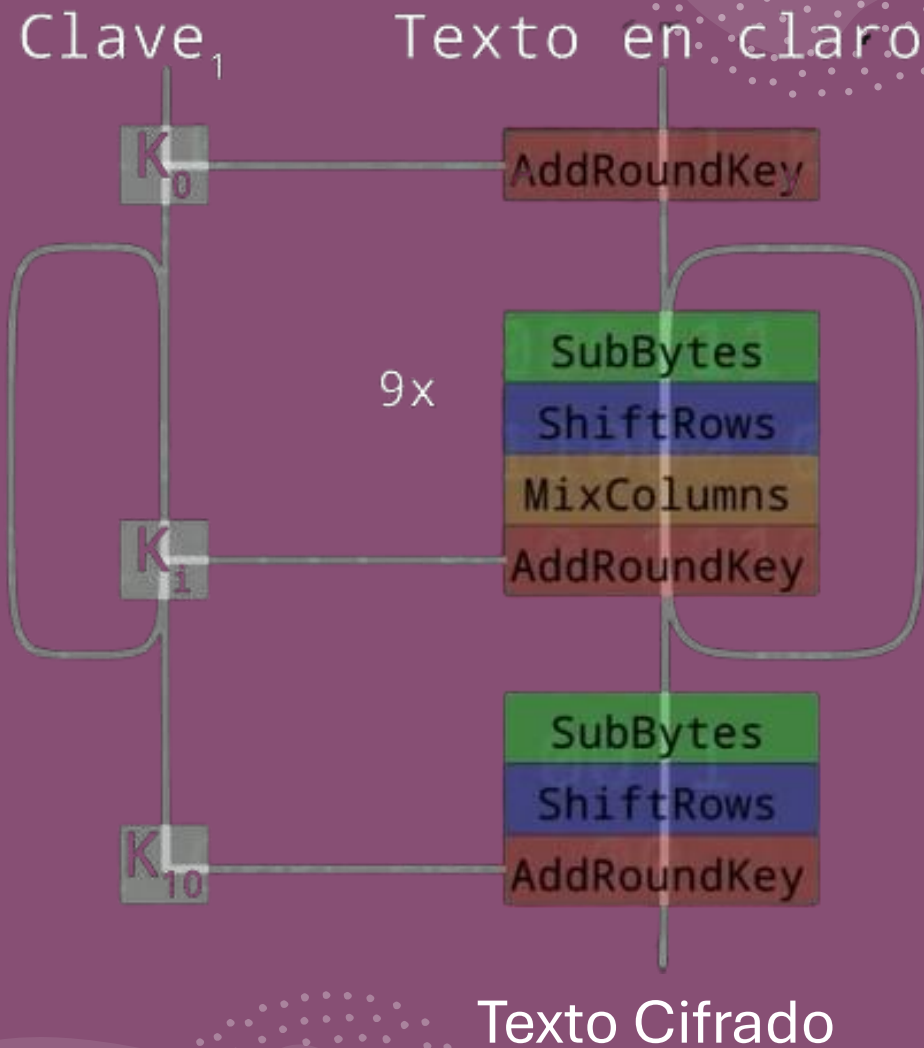


# Rondas intermedias

- **MixColumns:** Se realiza una operación de multiplicación de matrices entre cada columna del bloque de datos y una matriz fija.
- **AddRoundKey:** Se combina el bloque de datos con una subclave derivada de la clave original mediante una operación XOR.



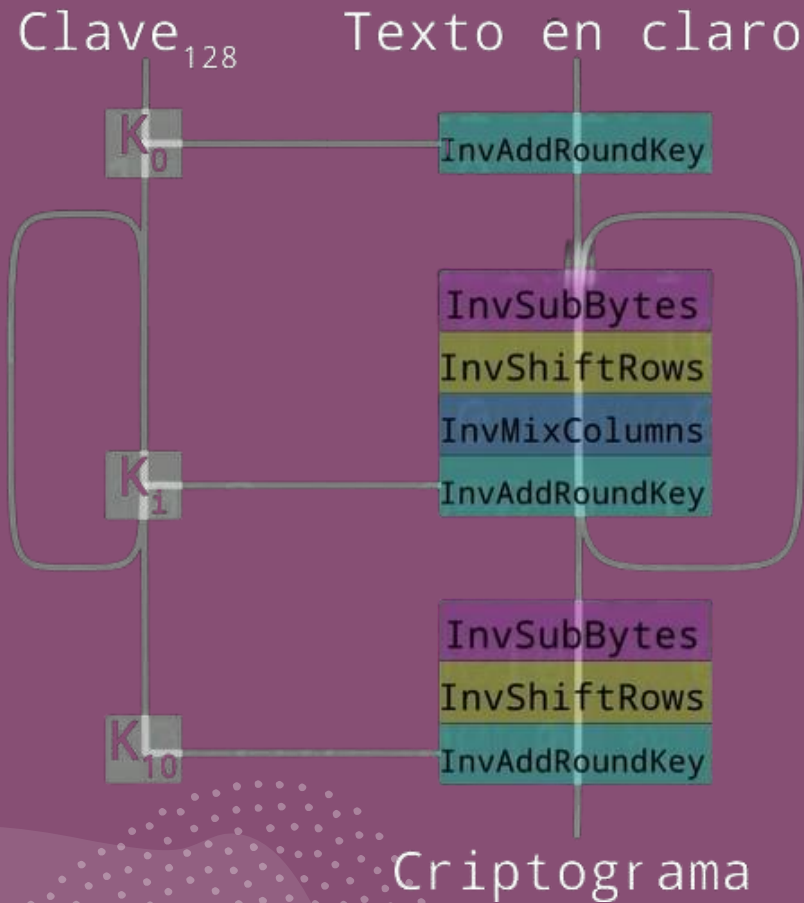
# Ronda final



- Para finaliza en el algoritmo solo se van a repetir las operaciones de **subBytes**, **shiftRows** y **AddRoundkey**.
- Una vez realizado estos pasos se obtiene nuestro Texto cifrado en una matriz de estado final.

# Descifrar el mensaje

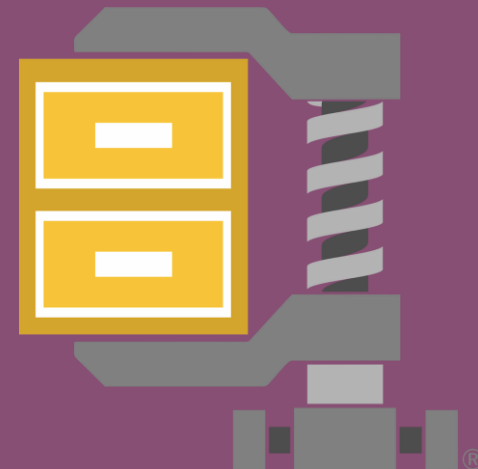
Se utiliza la misma clave para el descifrado. Y los pasos ahora son funciones en orden inverso.



# Aplicaciones

Entre algunas de las aplicaciones más comunes del algoritmo AES se incluyen

- Aplicaciones de mensajería, como Telegram o WhatsApp.
- El programa de compresión de archivos WinZip.



# Algoritmo simétrico de flujo

Estos algoritmos no retienen los datos cifrados en su memoria, sino que los cifra a medida que entran.

Se generan un "flujo de claves" a partir de un numero aleatorio.

El paradigma de este tipo de algoritmos es el One Time Pad.

Algoritmo simétrico de flujo : **ChaCha20**

# ChaCha20

Es un algoritmo de cifrado simétrico que soporta claves de 128 y 256 bits.

El número 20 es porque el algoritmo realiza 20 rondas de funciones no lineales para poder cifrar información.

ChaCha20 está diseñado para ser soportado por dispositivos que no posean una capacidad de procesamiento tan grande como lo son dispositivos IoT, teléfonos celulares, relojes inteligentes, etc.

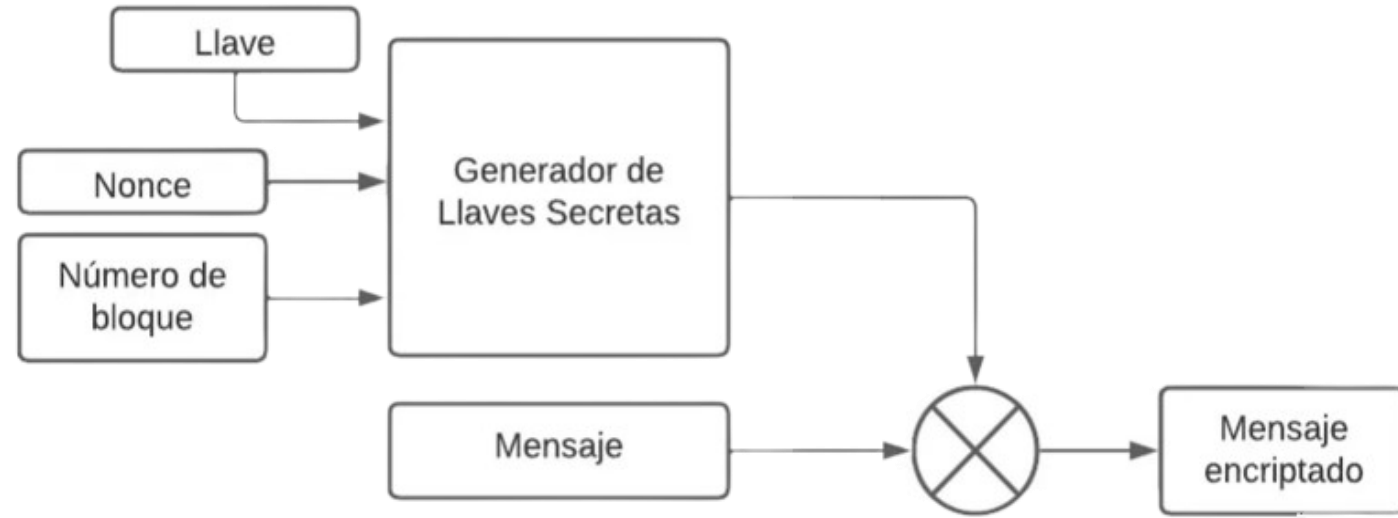


# ¿Como funciona?

## Generador de llaves secretas:

1. Una llave o clave de 256 bits que será la que usaremos para encriptar y desencriptar el mensaje.
2. Una variable llamada “**nonce**” (number used once) de tamaño 96 bits el cual es un número aleatorio arbitrario.
3. Usa un numero de bloque o contador de 32 bits y es un valor que se incrementa cada vez que se cifra un nuevo bloque, inicia en 1.

El mensaje es encriptado por medio de aplicar una operación XOR de los bits del mensaje con la llave secreta. Y así obtenemos el mensaje encriptado.





|          |          |       |       |
|----------|----------|-------|-------|
| expa     | nd 3     | 2-by  | te k  |
| K        | K        | K     | K     |
| K        | K        | K     | K     |
| Contador | Contador | Nonce | Nonce |

## Funcionamiento del algoritmo

- La primera fila de esta matriz se llena con una cadena que siempre es constante, la cual es “expand 32-byte k” de tamaño 128 bits.
- Las siguientes 2 filas se llenan con la clave que el usuario ingrese.
- La última fila se llena con 2 variables, el contador o número de bloque y el nonce en formato.

|   |   |   |   |
|---|---|---|---|
| A | A | A | A |
| B | B | B | B |
| C | C | C | C |
| D | D | D | D |

Se entra a un proceso llamado “**Quarter-round**”, el cual mezcla toda la información de la matriz para obtener las llaves secretas con las que trabajará.

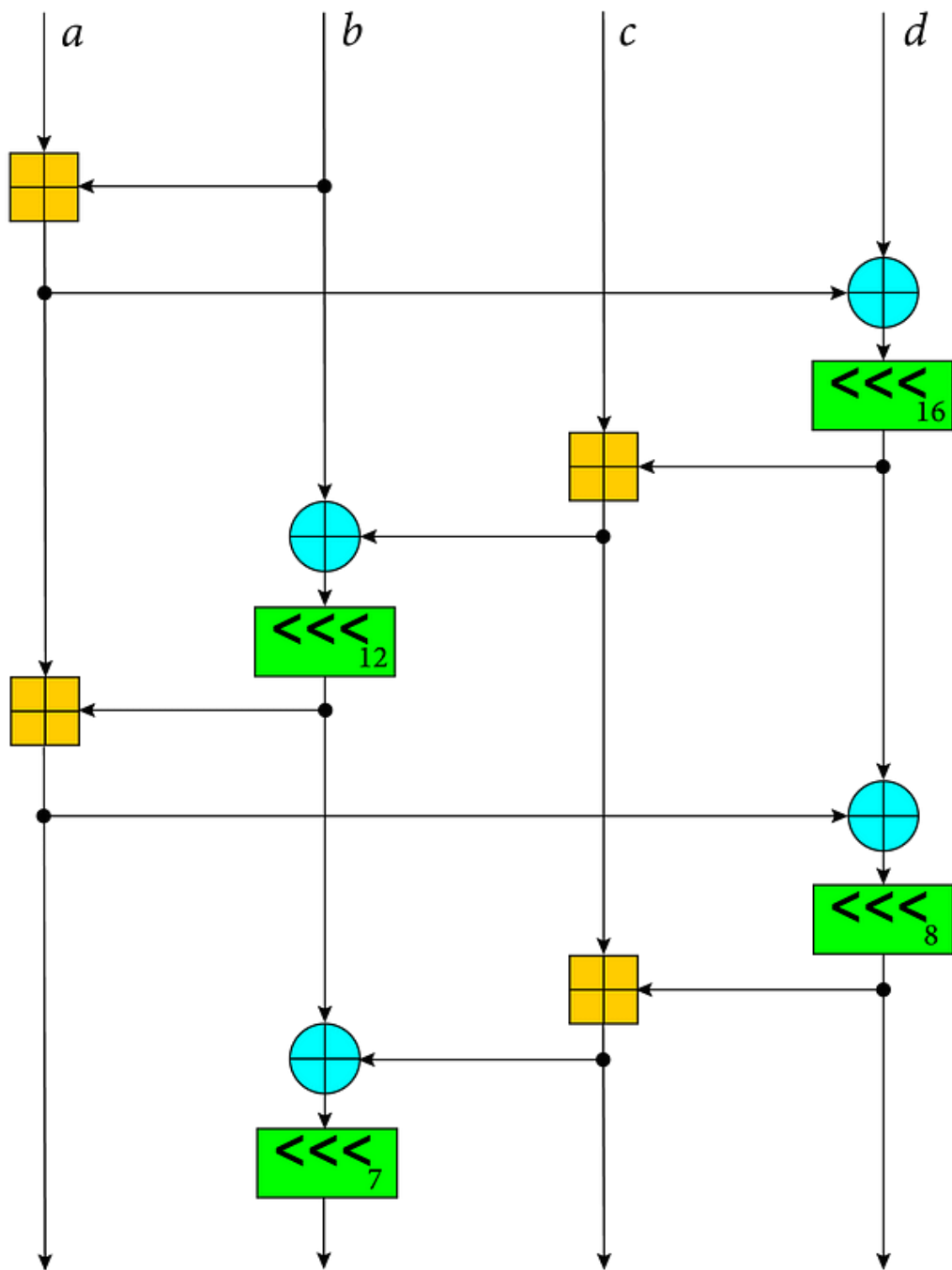
La matriz se va a dividir de 2 formas, por sus columnas o en diagonales. En 20 rondas, 10 para las columnas y 10 para las diagonales.

|   |   |   |   |
|---|---|---|---|
| A | A | A | A |
| B | B | B | B |
| C | C | C | C |
| D | D | D | D |


Sigue una serie de operaciones binarias entre las celdas las cuales son las siguientes.

- Suma binaria.
- Operación XOR.
- Desplazamiento de bits a la izquierda.





|   |
|---|
| A |
| B |
| C |
| D |


$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} + \begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix}$$



- Esta serie de operaciones se repiten un total de 20 veces.
- Se toma la copia de la matriz original que se tenía guarda y se le suma a la matriz mezclada para imposibilitar el obtener la matriz original sin la clave principal.
- Y obtenemos nuestro mensaje encriptado.

# Algoritmos asimétricos



# ¿Qué es el algoritmo asimétrico?



Fue inventada en 1975 por Whitfield Diffie y Martin Hellman

También es conocido como clave pública y hay dos claves: una clave se utiliza para la encriptación y otra distinta para la desencriptación.

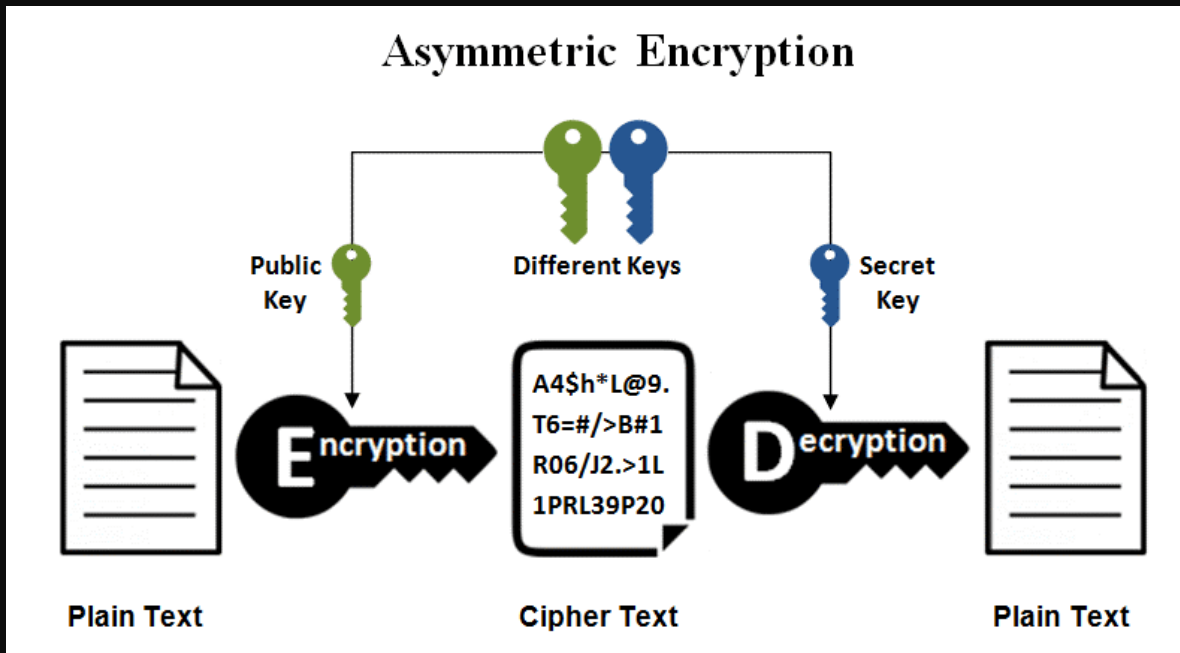
La encriptación asimétrica consume mayor tiempo de procesamiento y es más cara de producir.

Algunos algoritmos asimétricos son:

- RSA(Rivest-Shamir-Adleman)
- Curva elíptica ECC



# ¿Cómo funciona este algoritmo?



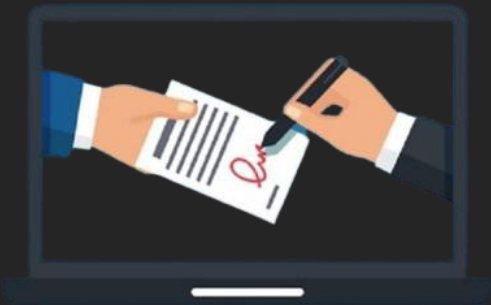
## Cifrado y descifrado:

- Para cifrar un mensaje, se utiliza una clave pública que está disponible para cualquier persona.
- El mensaje cifrado solo puede descifrarse con la clave privada del destinatario.

## Firma digital:

- Para firmar digitalmente un mensaje, se utiliza la clave privada del remitente, y la firma puede ser verificada por cualquier persona con la clave pública del remitente.





La estructura del funcionamiento del cifrado asimétrico funciona de esta forma:

- **Mensaje + clave pública** = Mensaje cifrado
- **Mensaje encriptado + clave privada** = Mensaje descifrado
- **Mensaje + clave privada** = Firma digital.
- **Mensaje firmado + clave pública** = Autenticación del Mensaje



# RSA(Rivest-Shamir-Adleman)

- RSA es un algoritmo de cifrado de clave pública que lleva el nombre de sus inventores: Ron Rivest, Adi Shamir y Leonard Adleman.
- La seguridad de este algoritmo radica en el problema de la factorización de números enteros primos para ofrecer un cifrado de 1024 bits y una longitud de clave de hasta 2048 bits.

# Generación de claves

Se eligen dos números primos grandes,  $p$  y  $q$ , de forma aleatoria. Que sean de tamaño igual o superior a 512 bits.

2 3 5 7 11  
13 17 19 23 29

Se calcula el producto  $n = p * q$



# Generación de claves

- Se calcula  $\phi$  que es la función de Euler con la formula :

$$\Phi_n = (p-1) * (q-1)$$

- Se tiene que escoger un valor de clave publica "e" que este entre el rango:



$$1 < e < \Phi(n)$$

# Generación de claves

- Se calcula “d” con el algoritmo extendido de Euclides:

$$d_A = \text{inv}(e_A, \Phi(n_A))$$

- La clave pública será el par (e, n)
- La clave privada será (d, n)



# Cifrado del mensaje

- Convertimos el mensaje que queremos cifrar en un número entero “m”, utilizando algún esquema de codificación como UTF-8.
- El texto cifrado será “c” y se calcula elevando m a la potencia del exponente de cifrado “e” y luego tomando ese resultado con el módulo de n

$$c = (m^e) \bmod n.$$

{UTF-8}



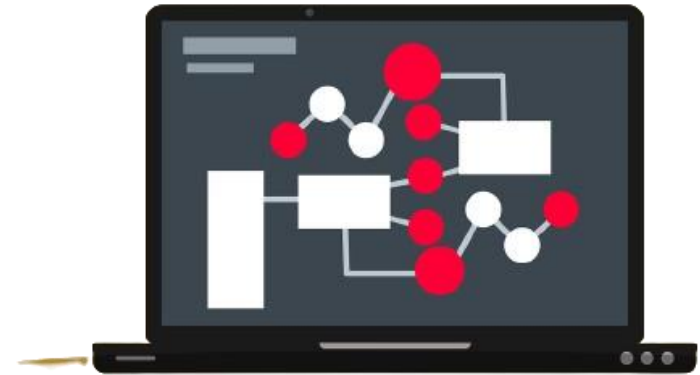
# Descifrado del mensaje

- Se recibe el texto cifrado “C”.
- El mensaje original se calcula elevando “c” a la potencia del exponente de descifrado “d” y tomando el resultado módulo n.

$$m = (c^d) \bmod n$$

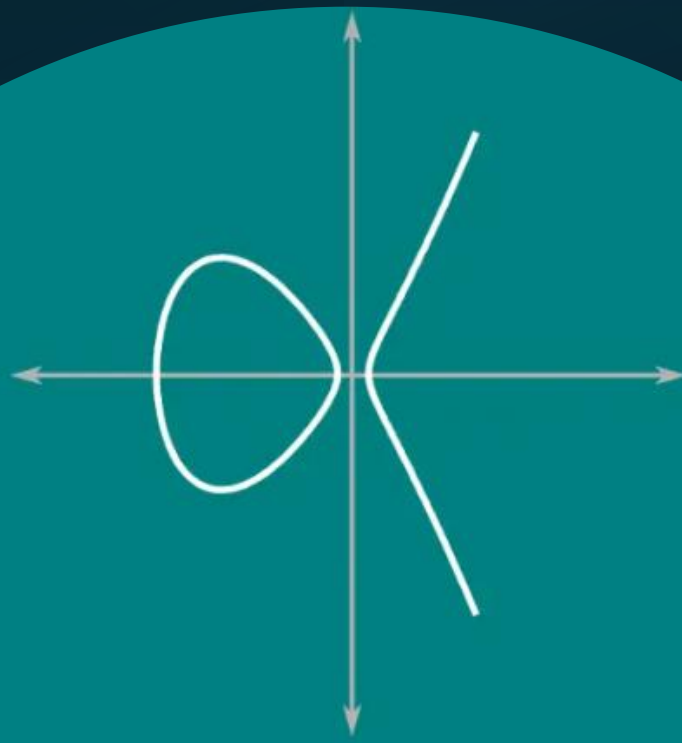


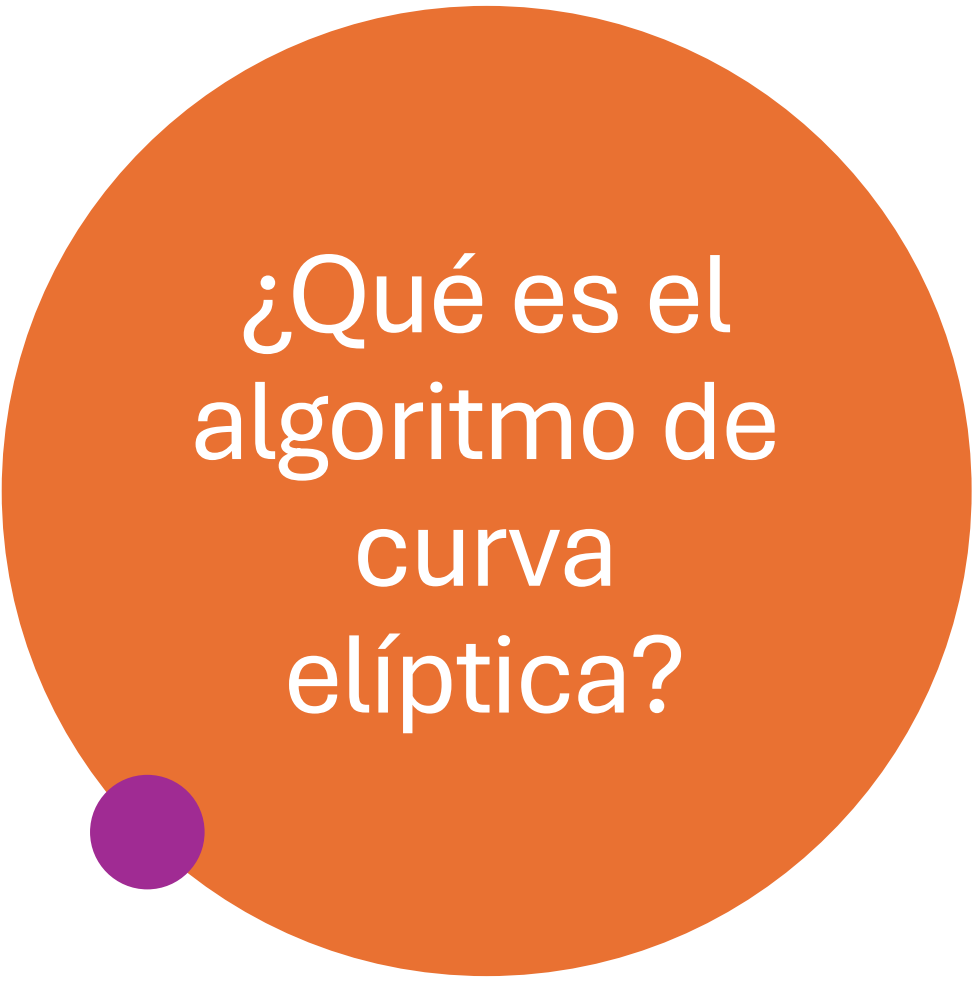
- Se necesitarían 1.500 años de cálculo para descifrar una versión de apenas 768 bits, si tenemos en cuenta que el algoritmo ofrece un cifrado de 1024 bits hasta 2048 bits.





# Algoritmo de curva elíptica ECC

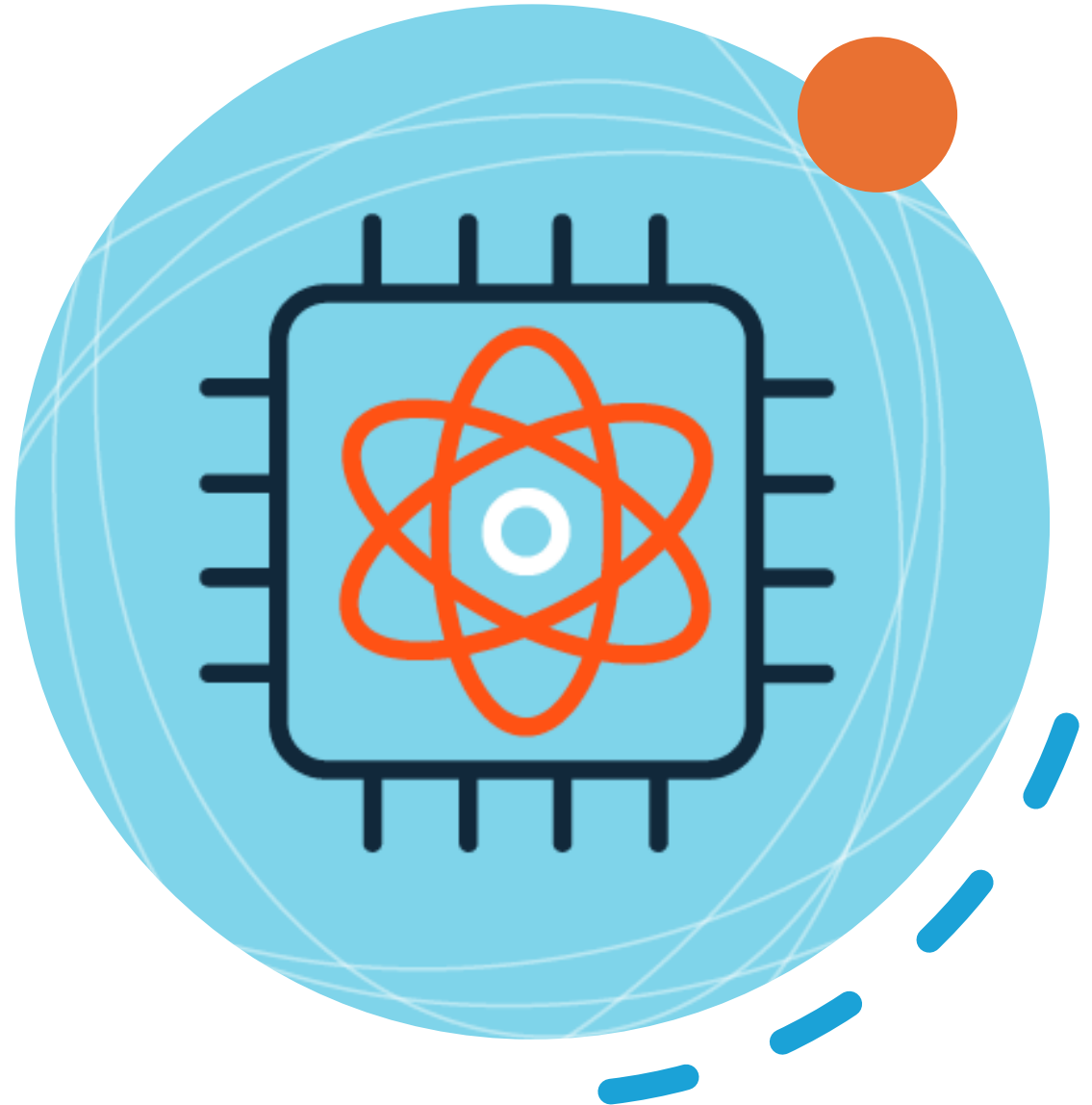




# ¿Qué es el algoritmo de curva elíptica?

- Este método fue propuesto en 1985 por Neal Koblitz y Victor S. Miller.
- Es un conjunto de procedimientos matemáticos diseñados para trabajar con curvas elípticas sobre cuerpos finitos.
- Estos algoritmos aprovechan propiedades especiales de las curvas elípticas para realizar operaciones criptográficas de manera eficiente y segura.

- Los métodos basados en curvas elípticas tienen su fortaleza en que por ahora no existen técnicas computacionales eficientes que permitan resolver el logaritmo elíptico.
- Se sabe que un ordenador cuántico podría romper el diseño, pero todavía no existe uno con la capacidad de hacerlo.



- Este algoritmo tiene una ventaja al anterior mencionado RSA y es que son más rápidos y necesitan números más pequeños para conseguir la misma fortaleza.

| Tamaño de clave ECC | Tamaño de clave RSA |
|---------------------|---------------------|
| 160 bits            | 1024 bits           |
| 224 bits            | 2048 bits           |
| 256 bits            | 3072 bits           |
| 384 bits            | 7680 bits           |
| 521 bits            | 15360 bits          |



# ¿Dónde se utilizan?

- **Voto electrónico:** Se utilizan los puntos de una curva elíptica para garantizar características como el secreto, la resistencia a los intentos de voto forzado o la posibilidad de verificar la transparencia del proceso electoral por parte de cualquier persona.
- **Blockchain:** Son utilizadas para la creación de firmas digitales a través del algoritmo, se pueden encontrar curvas elípticas en protocolos como Bitcoin o Ethereum



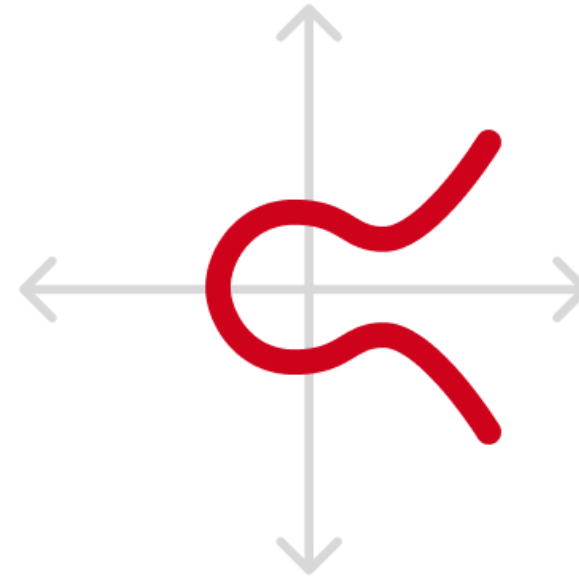
## ¿Cómo funciona?

Una curva elíptica se define mediante una ecuación de forma

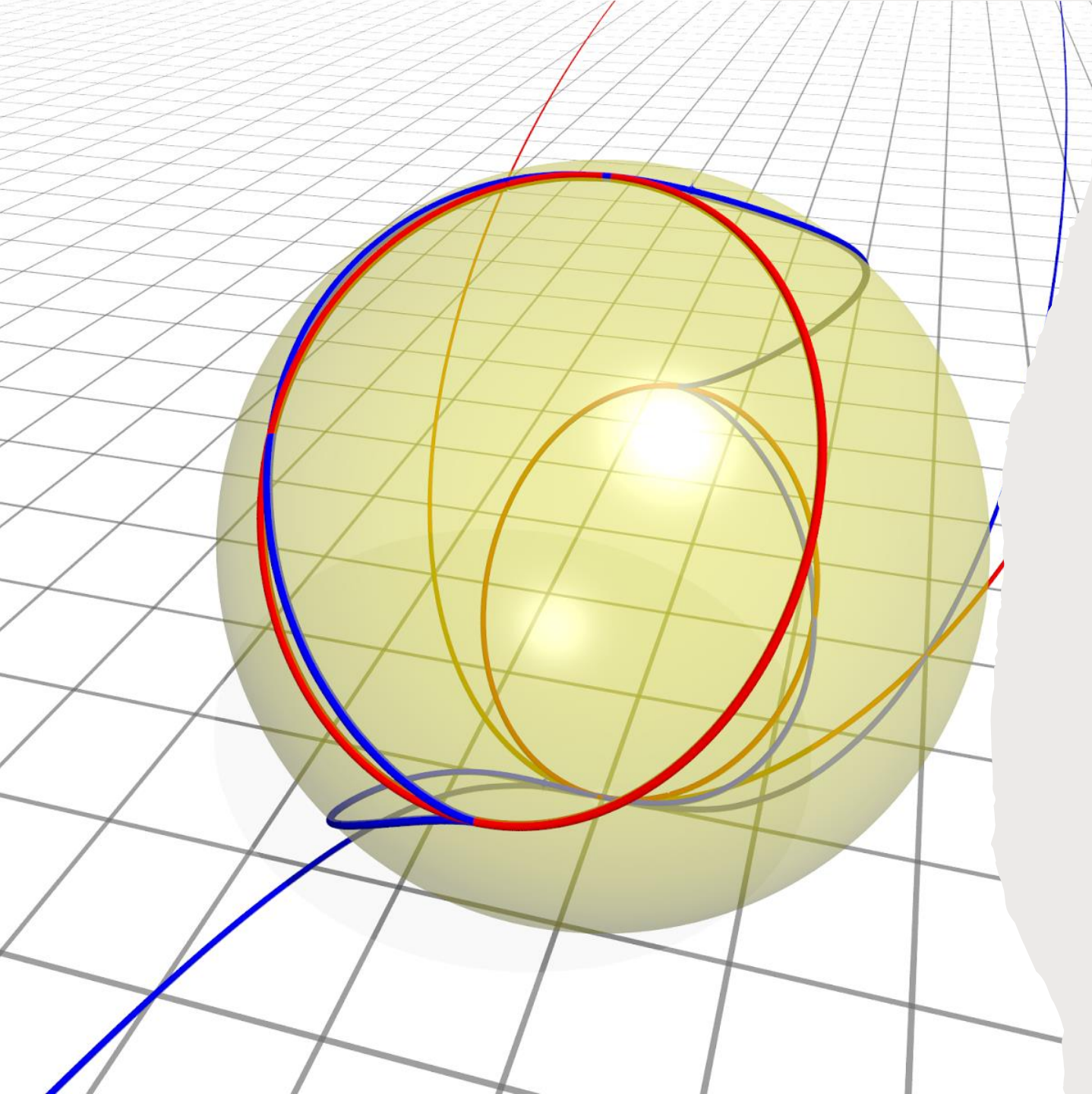
$$y^2 = x^3 + ax + b$$

Donde  $a$  y  $b$  son constantes y la curva se define sobre un campo finito.

La seguridad de la criptografía de curva elíptica se basa en la dificultad de resolver lo que se conoce como el problema de logaritmo discreto de la curva elíptica.



$$y^2 = x^3 + ax + b$$



Primer paso es elegir una misma curva para el emisor y receptor.

Existen curvas que ayudan a facilitar el proceso para la encriptación:

Las más comunes son: *secp256k1*, *secp256r1*, *secp384r1*, *secp521r1*

Estas curvas están definidas por un conjunto de parámetros establecidos en la séxtupla:

$$\mathbf{T} = (\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{G}, \mathbf{n}, \mathbf{h})$$

# Curva secp256k1

- Esta curva fue creada por Certicom, una compañía canadiense y se hizo popular por ser usada en Bitcoin.

- Sec: Standards for Efficient Cryptography.
- p: representa el campo primo.
- 256: Longitud en bits del campo primo.
- k1: Indica que es la primera curva de este tipo.

| Parámetro | Valor  |
|-----------|--|
| p         | 0xfffeffffc2f  |
| a         | 0x00<br>000000000000000000000000   |
| b         | 0x00<br>0000000000000000000000007  |
| G         | (0x79be667ef9dcbbac55a06295ce870b07029bfcd b2d<br>ce28d959f2815b16f81798,<br>0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68<br>554199c47d08ffb10d4b8) |
| n         | 0xffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8c<br>d0364141  |
| h         | 0x1  |

$$a = 0$$

$$b = 7$$

$p = 2^{256} - 2^{32} - 997$

La curva corresponde con la ecuación siguiente:

$$y^2 = x^3 + 7 \pmod{p}$$



Se elige un punto base  $G$  sobre la curva elíptica Este punto es el mismo para todos los usuarios que utilicen la curva

Se utiliza la siguiente ecuación:

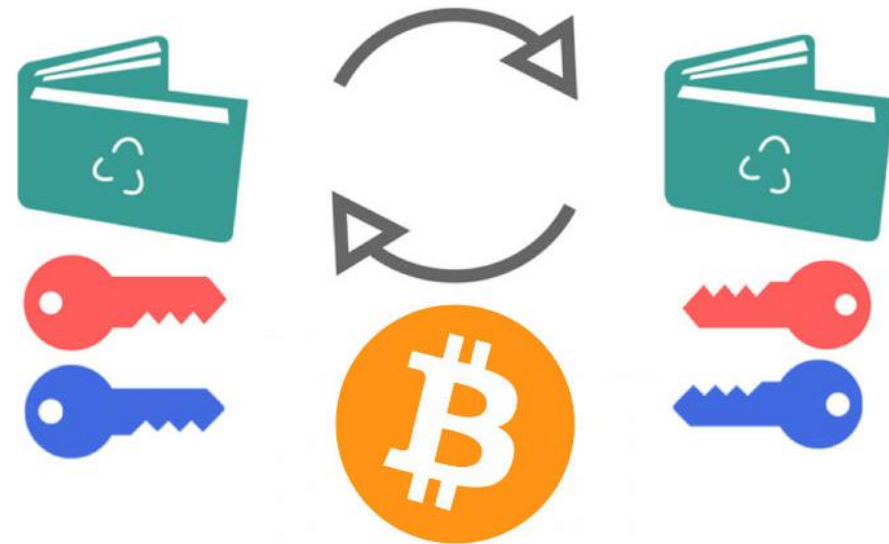
$$P = k * G$$

$K$ : será la clave privada del emisor.

Así obtenemos nuestra clave privada y pública:

**Clave privada:**  $k$

**Clave pública:**  $P = k * G$



Se tiene un numero  $k$  de tamaño 128 bits.

Eso crearía  $2^{128}$  combinaciones para poder encontrar el valor de  $k$ .

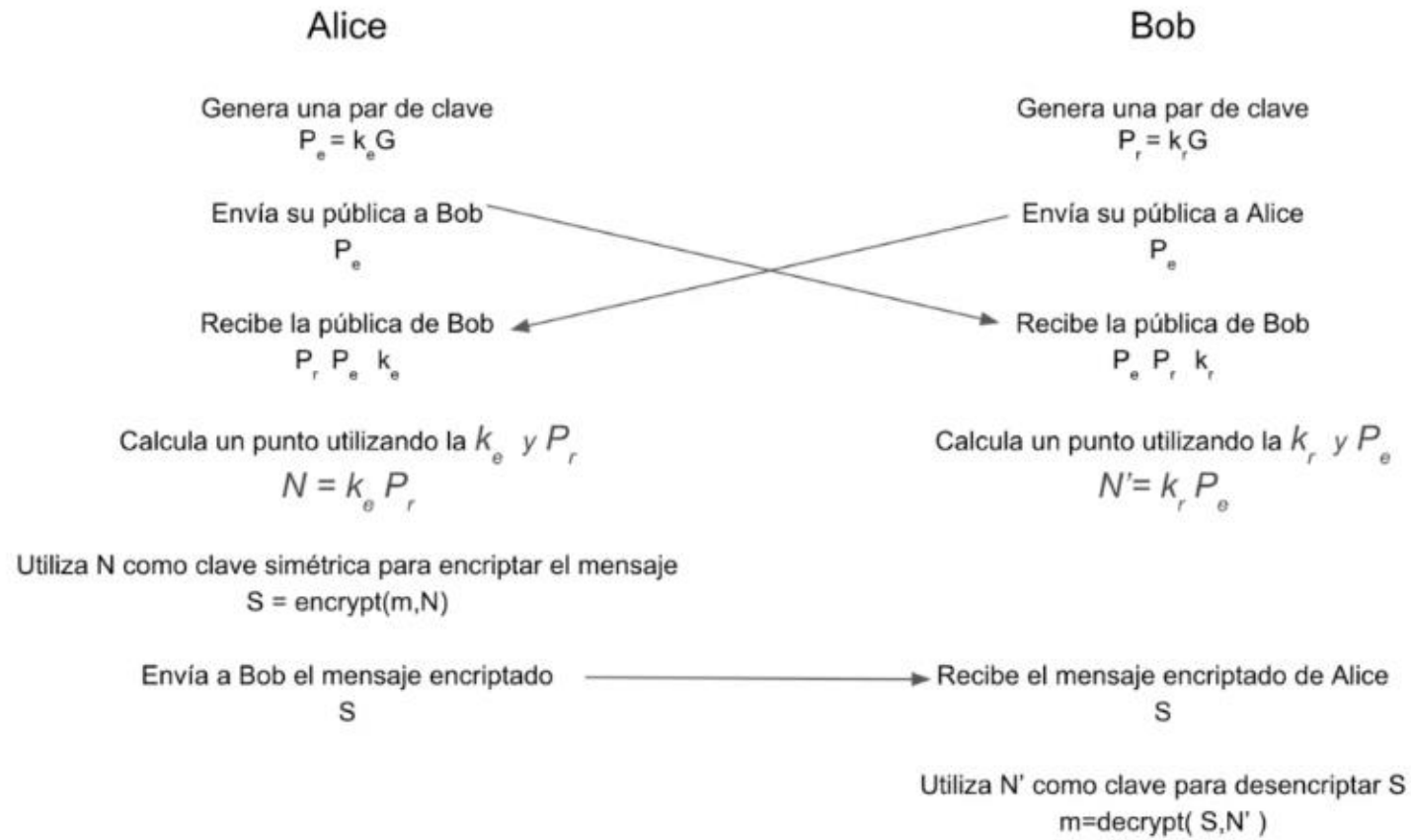
Suponiendo que existe una computadora que realiza 1 millón de sumas por segundo.

Se necesitarían 30.000.000.000.000.000 millones de años para realizar  $2^{128}$  sumas, teniendo en cuenta que la edad del universo es de 14.000 millones de años.

La probabilidad de encontrar  $k$  por fuerza bruta es mínima.



# Encriptación y desencriptación del mensaje





# Conclusión



# Referencias

- ¿Qué es la encriptación? | Cloudflare. (s. f.). <https://www.cloudflare.com/es-es/learning/ssl/what-is-encryption/>
- 
- Cifrado basado en hardware frente a basado en software. (s. f.). [Vídeo]. Kingston Technology Company. <https://www.kingston.com/latam/blog/data-security/what-is-encryption>
- ¿Qué es el cifrado de datos? Definición y explicación. (2023, 13 diciembre). latam.kaspersky.com. <https://latam.kaspersky.com/resource-center/definitions/encryption>
- 
- ¿Qué es el cifrado? Definición de cifrado de datos | IBM. (s. f.). <https://www.ibm.com/mx-es/topics/encryption>
- ¿Qué es un ataque de fuerza bruta? | Cloudflare. (s. f.). <https://www.cloudflare.com/es-es/learning/bots/brute-force-attack/>
- 
- Solis, C. R. (2023, 28 noviembre). Funcionamiento de Chacha20 - Carlos Rivas Solis - Medium. Medium. <https://medium.com/@carlosrivas.solis/funcionamiento-de-chacha20-50441be3493e>
- 
- ¿Qué es la encriptación y cómo funciona? | Google Cloud | Google Cloud. (s. f.). Google Cloud. <https://cloud.google.com/learn/what-is-encryption?hl=es-419>
-

# Referencias

- Ramírez, H. (2023, 24 marzo). Qué es la criptografía asimétrica y cómo funciona. Grupo Atico34. <https://protecciondatos-lopd.com/empresas/criptografia-asimetrica/>
- ArletHv. (2023, 14 agosto). Algoritmo de criptografía RSA. Huawei. <https://forum.huawei.com/enterprise/es/algoritmo-de-criptograf%C3%ADa-rsa/thread/691222638527135744-667212881550258176>
- Trevino, A., & Trevino, A. (2024, 26 marzo). ¿Qué es la criptografía de curva elíptica? Keeper Security Blog - Cybersecurity News & Product Updates. <https://www.keepersecurity.com/blog/es/2023/06/07/what-is-elliptic-curve-cryptography/>
- Kolokium. (2023, 13 junio). Criptografía de curva elíptica - Kolokium. <https://kolokium.com/blog/criptografia-de-curva-eliptica/>
- Crowford, D. (2024, 23 enero). What is ChaCha20? Proton. <https://protonvpn.com/blog/chacha20/>
- Secp256k1: un Algoritmo Clave en Criptomonedas. (2023, 25 agosto). Nervos Network. [https://www.nervos.org/es/knowledge-base/secp256k1\\_a\\_key%20algorithm\\_%28explainCKBot%29](https://www.nervos.org/es/knowledge-base/secp256k1_a_key%20algorithm_%28explainCKBot%29)