Universidad Nacional Autónoma de México
Facultad de Ingeniería

Sistemas operativos

Profesor:

ING. GUNNAR EYAL WOLF ISZAEVICH

Alumno:
Hernández Gutiérrez Carlos Mario

Grupo: 06

Exposición:

Algoritmos de encriptación

Fecha de entrega: 14/mayo/2024

Semestre 2024-2

Índice

Introducción

Un algoritmo de cifrado o algoritmo criptográfico es un conjunto de reglas y procesos matemáticos, su función principal es encriptar y desencriptar un determinado mensaje este puede ser documentos, texto, información importante, etc. Esto funciona empleando una o más claves como parámetros del algoritmo, de modo que sean necesarias para recuperar el mensaje a partir de la versión cifrada. El mensaje antes de encriptar se llama texto plano y una vez encriptado se llama texto cifrado.

Estos algoritmos ayudan a prevenir fraudes de datos como los que llevan a cabo los hackers, con el objetivo de obtener información, ya sea personal o de una empresa ilegalmente.

Sin la existencia de estos no se podrían hacer muchas cosas que hoy en día hacemos de forma cotidiana, como sacar dinero del cajero de forma segura, comprar por internet o navegar con la confianza de que nadie estará viendo lo que buscamos y escribimos.

Las características principales de los algoritmos de cifrado son:

- **Confidencialidad:** Los algoritmos deben ser capaces de ocultar la información original para que solo pueda ser accedida por las personas autorizadas.
- **Integridad:** Deben garantizar que la información no se altere durante su transmisión o almacenamiento.
- **Robustez**: Los algoritmos deben ser capaces de resistir ciberataques y mantener la información a pesar de las amenazas.
- **Velocidad**: Deben ser capaces de codificar y decodificar información en un tiempo adecuado para el usuario.
- **Simplicidad**: Los algoritmos deben ser fáciles de implementar y utilizar.
- **No repudio:** Deben permitir a los destinatarios verificar la autenticidad del remitente y evitar que el remitente niegue haber enviado la información.

Para empezar a entender estos temas de los algoritmos de cifrado debemos saber el cómo se dividen y cuáles son las diferencias entre cada uno.

Vamos a hablar de 2 tipos de cifrado, la criptografía cuyos algoritmos solo usan una llave y se llaman algoritmos de cifrado simétricos y la criptografía de algoritmos de dos llaves llamados algoritmos de cifrado asimétricos.

Algoritmo simétrico

Este tipo de cifrado es de los primeros que existieron y se basa en utilizar una única clave secreta que se encarga de cifrar y descifrar la información y al ser una sola clave es necesario que el emisor como el receptor tengan esta para obtener la información correcta.
Cualquier otro usuario que quiera acceder al mensaje cifrado, deberá tener la contraseña de descifrado, de lo contrario el mensaje será erróneo o incompleto.
El método para cifrar los datos se basa en que el emisor va a cifrar el mensaje con su clave privada, lo enviará a través de un canal y el destinatario lo tendrá que descifrar con la misma clave privada que ha usado el emisor para evitar perdida de información.
Una de las desventajas de este tipo de algoritmos es tener una clave privada para todos los usuarios para cifrar y descifrar la información del mensaje.
Los ataques por fuerza bruta son unas de las tantas vulnerabilidades que existen para estos algoritmos, hay que tener en cuenta que al tener la llave pública puede ser fácil de obtener información, corromperla a través de intentos aleatorios hasta dar con el correcto y para evitar estas cosas se debe tener cuidado en donde se guarda esta clave, también el implementar un algoritmo complejo ayuda a que no sea vulnerado con facilidad y considerar la longitud de la clave empleada influye para evitar estos ataques.

Las ventajas de los algoritmos simétricos es que son rápidos y con el tiempo se ha ido mejorando e incorporando a hardware como en los procesadores de ordenadores, servidores, routers, etc.

Estos algoritmos se pueden encontrar en muchos campos donde se requiere proteger la confidencialidad de la información como, por ejemplo:

- **Comunicaciones seguras**: En aplicaciones de mensajería instantánea y correos electrónicos, se utilizan para asegurar que los mensajes y las conversaciones permanezcan privadas entre el remitente y el destinatario. Aplicaciones como WhatsApp y Telegram utilizan este tipo de cifrado simétrico.
- **Almacenamiento de datos sensibles**: En dispositivos de almacenamiento, como discos duros, unidades USB y servicios de almacenamiento en la nube, utilizan algoritmos de cifrado simétrico.
- **Protección de contraseñas**: En sistemas de autenticación y seguridad, como el almacenamiento de contraseñas en bases de datos, se utilizan algoritmos de cifrado simétrico para proteger las contraseñas de los usuarios.
- **Protocolos de seguridad de redes**: En protocolos de seguridad de redes, como SSL/TLS, SSH, se utilizan algoritmos de cifrado simétrico para proteger la comunicación entre dispositivos y servidores.

Dentro de estos algoritmos se encuentran dos tipos dependiendo de la cantidad de datos de entrada que se manejan a la vez: algoritmos de cifrado por bloques y algoritmos de cifrado de flujo.

Cifrado por bloques:
Los algoritmos de cifrado por bloques toman bloques de tamaño fijo del texto en plano y producen un bloque de tamaño fijo de texto cifrado, estos bloques normalmente son del mismo tamaño que la entrada. El tamaño del bloque entre más grande sea ayuda a tener mayor seguridad y evitar ataques de texto cifrado.

Para la asignación de bloques los algoritmos de cifrado simétrico realizan sustituciones y permutaciones en el texto plano hasta obtener el texto cifrado. Una característica para resaltar de estos algoritmos es que la encriptación y desencriptación toman los mismos pasos para llegar a su resultado, pero solo cambian en que las funciones empleadas para la desencriptación se toman en orden inverso en la encriptación. A continuación se muestran un ejemplo de este algoritmo.

AES (Advanced Encryption Standard)

El algoritmo AES, por sus siglas en inglés Advanced Encryption Standard (Estándar de Cifrado Avanzado), es una técnica de cifrado simétrico muy popular y utilizada para proteger datos confidenciales. Fue aprobado por el Instituto Nacional de Estándares y Tecnología (NIST) de Estados Unidos en 2001.

La fortaleza del AES radica en su capacidad para resistir ataques criptográficos conocidos, como el criptoanálisis diferencial y lineal, incluso con el aumento de la potencia computacional.
AES es un algoritmo que se basa en el cifrado por bloques, **el tamaño fijo del bloque es de 128 bits**. La longitud de la clave se puede elegir, y se tiene disponible **128, 192 y 256 bits**, siendo la longitud de 128 bits el estándar, pero también se usa el de 256 bits.
AES se encarga de generar una matriz de 4×4, cifra la información y lo hace bloque a bloque en lo que se denomina "rondas". Los datos se empiezan a separar por bloques, y luego estos se cifran con la clave que elijamos y dependiendo el tamaño se harán en ciertas rondas.

- Para una clave de 128 bits se aplican 10 rondas de cifrado y nos genera una combinación potencial de 3.4 x10^38 combinaciones.

- Para una clave de 192 bits se aplican 12 rondas y nos genera una combinación potencial de 6.2 x10^57 combinaciones.

- Para una clave de 256 bits las rondas aplicadas son 14 y nos genera una combinación de 1.1 x10^77 combinaciones.

La forma en que se cifra es la siguiente:

Cada ronda consta de cuatro operaciones básicas:

- **Sustitución de bytes:** Se sustituyen los bytes del bloque de datos por otros bytes, según una tabla de sustitución fija conocida como S-box.
- **Desplazamiento de filas**: Todas las filas del texto encriptado se desplazan en una cierta posición, la primera fila se mantiene igual, la segunda se desplaza un byte, la tercera 2 bytes y la tercera fila 3 bytes.
- **Mezcla de columnas**: Se realiza una operación de multiplicación de matrices entre cada columna del bloque de datos y una matriz fija.
- **Adición de la clave de ronda**: Se combina el bloque de datos con una subclave derivada de la clave original mediante una operación XOR.
- Y una vez completado estos 4 pasos se va a repetir “n” rondas que se hayan elegido.

Texto cifrado: Una vez completadas todas las rondas de cifrado, se obtiene el texto cifrado final.

Texto descifrado:

- Para descifrar el mensaje cifrado, se utiliza la misma clave de cifrado que se utilizó para cifrarlo.

- El proceso de descifrado con AES implica aplicar una serie de rondas de descifrado, que son similares a las rondas de cifrado, pero en orden inverso.

Este algoritmo se puede encontrar en diferentes aplicaciones como:

- **VPNs**: Este proceso conecta a los usuarios con diferentes servidores, se utiliza el cifrado AES para proteger los datos del usuario contra filtraciones y ciberataques.
- **Gestores de contraseñas**: Los gestores de contraseñas se utilizan para almacenar de forma segura las credenciales de inicio de sesión ocupando una única clave maestra. AES se utiliza en estos casos para proteger este tipo de software.
- **Wi-Fi:** La conexión inalámbrica a Internet suele utilizar muchos métodos de cifrado y AES se encuentra en este tipo de conexiones.
- **Aplicaciones móviles**: Cualquier aplicación que incluya mensajería o intercambio de fotos, documentos, videos, etc. Utiliza AES para ayudar a la seguridad de los datos.
- AES se utiliza en diferentes lenguajes de programación, como C, C++, Java o Python. También se puede encontrar en programas de compresión de archivos como lo son WinZip, 7 Zip, etc.

Cifrado de flujo de datos:

Estos algoritmos operan sobre 1 bit, sobre bytes o palabras, de los datos de entrada cada vez. El algoritmo genera una secuencia de bits que se emplea como clave y la encriptación se realiza combinando la secuencia cifrante con el texto plano.

El paradigma de este tipo de algoritmos se llama *One Time Pad*, que funciona aplicando una XOR a cada bit de la entrada de datos junto con otro generado aleatoriamente para obtener un bit de la salida de manera cifrada. La secuencia de bits aleatorios será la clave de cifrado, que es del mismo tamaño que la entrada y la salida.
Para poder obtener el texto en plano, el texto encriptado debe pasar por el mismo proceso empleado para encriptar usando la misma secuencia. Uno de estos algoritmos es el ChaCha20 que se muestra a continuación:

ChaCha20

El algoritmo ChaCha20 es un algoritmo de cifrado simétrico que puede soportar claves de 128 y 256 bits este es un cifrado de flujo. El número 20 en su nombre proviene de que este algoritmo realiza 20 rondas donde se aplican funciones no lineales para poder cifrar la información.

La complejidad que tiene ChaCha20 está diseñada para que pueda ser soportado por dispositivos que no posean una capacidad de procesamiento como lo son dispositivos IoT, teléfonos celulares, relojes inteligentes, entre otros.
Este algoritmo tiene un funcionamiento sencillo de implementar y entender, el mensaje que se enviara es encriptado aplicando una operación XOR de los bits de este mensaje con algo llamado “llaves secretas” y así modificando el mensaje de tal forma que sea imposible leerlo y que a su vez pueda ser reversible aplicando el mismo método.

Chacha20 hace uso de un “generador de llaves secretas” el cual solo necesita de 2 cosas, una llave o clave de 256 bits que será la que usáremos para encriptar y desencriptar el mensaje, una variable llamada “*nonce*” el cual es un número aleatorio que solo debería ser utilizado una única vez y no volver a usarse en encriptaciones futuras.

- **Nonce** (number used once): Es un valor arbitrario de tamaño 96 bits que se combina con la clave secreta para generar el flujo de cifrado.
- **Contador o block number:** Es un valor que se incrementa cada vez que se cifra un nuevo bloque y este comienza en 1.

Y estas 2 variables juntas aseguran que el flujo de cifrado generado sea diferente para cada bloque.

Como funciona el algoritmo

- Se inicia con una matriz de 4x4 bloques, cada uno de estos bloques de 32 bits, dando un total una matriz de 512 bits:
    - La primera fila de esta matriz se llena con una cadena que siempre es constante, la cual es "*expand 32-byte k*".
    - Las siguientes 2 filas se llenan con la clave que el usuario ingrese.
    - La última fila se llena con 2 variables, el contador y el nonce, estos se reparten entre las 4 últimas celdas.
- Ahora que se tiene la matriz preparada, se toma una "copia" de esta en un buffer que se usara más tarde.
- Se entra a un proceso llamado "Quarter-round", el cual se va a dedicar a mezclar toda la información de la matriz para obtener las llaves secretas con las que trabajará.
- La matriz se va a dividir de 2 formas distintas, por sus columnas o en diagonales.
- Una vez dividida la matriz toca hacer Quarter-round, para lo cual tomaremos una columna o diagonal.
- Al tener una de estas 2 opciones sigue una serie de operaciones binarias entre las celdas, las cuales son las siguientes.
    - Suma binaria.
    - Operación XOR.
    - Desplazamiento de bit n a la izquierda.

Repitiendo los pasos

- Esta serie de operaciones se van a repetir 20 veces, alternando entre las columnas o diagonales de la matriz.
- Se toma la copia de la matriz original que se tomó antes y se le suma a la matriz mezclada para ahora sí, imposibilitar el obtener la matriz original sin la clave principal.
- Esta clave final termina por hacer XOR con la parte del mensaje original y en caso de que se necesite más seguridad para mensajes más largos, se repite todo el proceso de la clave secreta, solo que el contador y el nonce generan un nuevo número aleatorio para mayor seguridad y así obtener nuestro mensaje cifrado.

Algoritmo asimétrico.

La criptografía asimétrica fue inventada en 1975 por Whitfield Diffie y Matin Hellman este algoritmo también es conocido como clave pública, donde se usan dos llaves diferentes cada usuario tendrá una clave pública y otra privada en cada uno de los extremos de la comunicación para cifrar y descifrar la información.

La clave pública podrá ser consultada por todos los usuarios del sistema que quieran comunicarse y esta clave no servirá para el proceso de descifrado. Para esto el usuario necesitará tener una clave secundaria, es decir, la clave privada, para descifrar la información. De este modo, la clave privada solo la tiene la persona responsable de descifrar, sin vulnerar la seguridad de este algoritmo.

El proceso generalmente implica lo siguiente:

- **Cifrado y descifrado:** Para cifrar un mensaje, se utiliza la clave pública del destinatario. El mensaje cifrado solo puede descifrarse con la clave privada correspondiente. Esto proporciona confidencialidad a los datos.
- **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 de este mismo. Esto garantiza la autenticidad e integridad del mensaje.

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 = Mensaje firmado
- Mensaje firmado + clave pública = Autenticación

Ventajas:

- **Facilidad en el intercambio de claves**: La clave pública de un usuario puede ser compartida abiertamente, lo que facilita el intercambio seguro de claves entre partes que desean comunicarse de manera segura.
- **Autenticación y no repudio:** Los algoritmos asimétricos permiten la autenticación de usuarios y la firma digital de documentos, lo que garantiza la integridad y autenticidad de la información. Además, el remitente no puede negar haber enviado un mensaje firmado digitalmente, lo que se conoce como no repudio.
- **Seguridad en la comunicación en línea**: Los algoritmos asimétricos proporcionan un nivel adicional de seguridad en la comunicación en línea al

cifrar datos, de forma que solo el destinatario puede descifrarlos, incluso si el mensaje se intercepta durante su envío.

- **Escalabilidad:** Los sistemas de clave pública pueden admitir una amplia gama de usuarios sin necesidad de compartir secretos entre ellos, lo que los hace escalables y adecuados para implementaciones en grandes redes.

Desventajas:

- **Rendimiento computacional**: Los algoritmos asimétricos suelen ser más lentos en términos de rendimiento computacional en comparación con los algoritmos simétricos. Esto se debe a la complejidad matemática de las operaciones necesarias para cifrar y descifrar los datos.
- **Tamaño de clave**: Las claves asimétricas tienden a ser más largas en comparación con las claves simétricas, lo que puede resultar en un mayor consumo de recursos de almacenamiento y ancho de banda.
- **Vulnerabilidad a ataques de canal lateral**: Los algoritmos asimétricos pueden ser más susceptibles a ataques de canal lateral, como ataques temporales, que pueden permitir deducir información sobre las claves privadas.

Estos algoritmos se pueden encontrar en:

- **Protocolos de comunicaciones seguras:**
  - Protocolo IPsec (Seguridad de la Capa de Internet): Utiliza intercambio de claves asimétrico para establecer asociaciones de seguridad y garantizar comunicaciones VPN seguras.

- **Firmas digitales:**
  - Los algoritmos asimétricos se utilizan para crear y verificar firmas digitales, lo que garantiza la autenticidad, la integridad y la no repudiación de documentos electrónicos. Las firmas digitales son comúnmente utilizadas en contratos electrónicos, transacciones financieras y documentos legales.

- **Autenticación de usuarios:**
  - Protocolo SSH (Secure Shell): Utiliza intercambio de claves asimétrico como Diffie-Hellman para establecer un canal seguro para las comunicaciones remotas y para autenticar a los usuarios estableciendo conexiones seguras entre clientes y servidores.

- **Servicios en línea y aplicaciones web:**
  - Plataformas como PayPal, bancos en línea y otros servicios utilizan cifrado asimétrico para proteger las comunicaciones y transacciones financieras.
- **Dispositivos móviles e IoT:**
  - Los algoritmos asimétricos, especialmente ECC, se utilizan en dispositivos con recursos limitados para establecer comunicaciones seguras y actualizaciones de firmware seguras.
- **Blockchain y criptomonedas:**
  - Las criptomonedas como Bitcoin utilizan criptografía de curva elíptica (ECC) para generar y administrar las direcciones y firmas digitales en las transacciones.

Ahora hablaremos de algunos algoritmos asimétricos que se usan:

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. Fue presentado por primera vez en 1977 y desde entonces ha sido ampliamente usado.
El propósito principal de RSA es permitir el cifrado y descifrado de datos utilizando dos claves diferentes: una clave pública para cifrar y una clave privada para descifrar. Esto resuelve el problema de distribución de claves que enfrentan los algoritmos simétricos.

RSA utiliza la factorización del producto de dos números primos y ofrece un cifrado de 1024 bits y se puede obtener una longitud de clave de hasta 2048 bits.

Como funciona:

Primero se tienen que generar las claves:

- Se eligen dos números primos grandes, p y q, de forma aleatoria.
- Se calcula **n = p * q** (n se llama el módulo).

- Con fi que es la función φ de Euler se calcula **φ(n) = (p - 1) * (q – 1).**

- Se escoge un entero positivo e menor que φ(n) o que sea co-primo con φ(n).

- Se calcula d, el módulo multiplicativo inverso de e, es decir, **d = e^-1 mod φ(n)**.

- La clave pública es el **par (e, n)** esto es, el módulo y el exponente de cifrado.

- La clave privada es el **par (d, n)** esto es, el módulo y el exponente de descifrado y debe mantener en secreto junto con los valores de los números “p” y “q”.

Cifrado de un mensaje:

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

Descifrado de un mensaje:

- Obtención del texto cifrado: Recibimos el texto cifrado c.
- Cálculo del mensaje original: El mensaje original “m” se calcula elevando “c” a la potencia del exponente de descifrado “d” y tomando el resultado módulo n: **m = c^d mod n**

El funcionamiento es bastante complejo de entender, ya que son muchas operaciones que se realizan, pero a la vez lo hace demasiado seguro una porque se han hecho cálculos para poder descifrar algún mensaje que haya sido encriptado con estos algoritmos y se necesitarían 1.500 años de potencia de cálculo para descifrar su versión de apenas 768 bits, si tenemos en cuenta que el algoritmo ofrece un cifrado de 1024 bits hasta 2048 bits.

Es un algoritmo de cifrado es más lento como requiere dos claves diferentes de una longitud extensa, pero el nivel de seguridad que proporciona para la información es demasiado incomparable.

EEC (Criptografía de Curva Elíptica)

Este método fue propuesto en 1985 por Neal Koblitz y Victor S. Miller, utiliza una operación matemática basada en curvas elípticas sobre un campo finito.
ECC se basa en funciones matemáticas que son simples de calcular en una dirección. Crear una clave ECC es bastante sencillo hoy en día, pero romperla es prácticamente imposible. En el caso de ECC, esta dificultad se presenta en el problema de calcular el logaritmo discreto de un elemento de la curva elíptica con respecto a un punto base conocido públicamente y a esta propiedad se le llama ”problema de logaritmo discreto de curva elíptica"

Como funciona el algoritmo

Generación de claves:

El método utiliza una curva elíptica de la siguiente forma:

$$y2 = x3 + ax + b$$

Donde a y b son parámetros que determinan la forma y la posición de la curva. La curva también puede contener un punto llamado "punto en el infinito", que actúa como el elemento neutro en la operación de suma.

El primer paso es elegir una de las curvas del dominio, estos parámetros se deben elegir con especial cuidado porque no todas las curvas poseen la misma fortaleza, y es por esto que empresas dedicadas a este tipo de seguridad han creado curvas con parámetros de dominios que ya estén validados y definidos para su uso y tener mayor simplicidad al momento de usar este algoritmo.
Las más comunes son: secp256k1, secp256r1, secp384r1, secp521r1

Se elige un punto base G sobre la curva elíptica, este punto es el mismo para todos los usuarios que utilicen la curva y lo podemos considerar como un dato público base sobre la que se construye la criptografía de curva elíptica. Para encontrar los siguientes parámetros se usa la siguiente ecuación:

$$\mathbf{P = k*G}$$

Donde:

- **P** y **G** son puntos de la curva que usaremos.
- **K:** será la clave privada del emisor y es un entero que define el número de iteraciones que se van a realizar sobre el punto G para obtener el punto P.
- **G:** es el punto base o generador, que es conocido por todo el mundo y único para la curva que estamos utilizando.

- **P:** es la clave pública y este se obtiene iterando k veces sobre G.

1. **Clave privada:** k
2. **Clave pública**: P = k * G

Sonará muy raro pensar que si se tiene a P y G como elementos públicos es fácil encontrar K y descifrar el mensaje, pero no es tan sencillo como parece. Dado las combinaciones que se hicieron y la complejidad del algoritmo es imposible tener el equipo adecuado para llegar a la respuesta adecuada.

Ejemplo:

- Supongamos que elegimos un número k de tamaño 128 bits, eso significa que el atacante debe realizar 2^128 operaciones de suma sobre el punto G, hasta encontrar el punto P, en ese momento habría encontrado el número k es la clave privada.
- Suponiendo que se tiene una capacidad de cómputo que permite realizar 1 millón de sumas por segundo.
- Se necesitarían 30.000.000.000.000.000 millones de años para realizar 2128 sumas, teniendo en cuenta que la edad del universo es de 14.000 millones de años, la probabilidad de encontrar k por fuerza bruta son mínimas.
- Esto es lo que hace que el algoritmo sea completamente seguro y no sea vulnerado de esta forma.

Cifrado:

- El remitente obtiene la clave pública Q del destinatario.
- Elige un número entero aleatorio d.
- Calcula el punto C1 = d * G y el punto C2 = M + d * Q, donde M es el mensaje representado como un punto sobre la curva.
- El texto cifrado es el par de puntos (C1, C2).

Descifrado:

- El destinatario recibe el texto cifrado (C1, C2).
- Utilizando su clave privada k, calcula el punto M' = C2 - k * C1.
- M' es el mensaje descifrado, representado como un punto sobre la curva.

ECC aporta mayor seguridad y algunos beneficios son los siguientes:

- ECC permite que las claves sean más pequeñas en extensión en comparación con otros sistemas criptográficos.

- ECC escala mejor, mientras que el RSA se vuelve más lento y pesado.

Una comparación con el algoritmo anteriormente hablado RSA sería el siguiente:

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

Los tamaños de clave más pequeños de ECC significan que se puede lograr un cifrado más fuerte con menos potencia informática

Donde se utilizan:

- **Protocolos de comunicación seguros**: ECC se utiliza en varios protocolos de comunicación seguros para proporcionar cifrado, firmas digitales e intercambio de claves. Algunos ejemplos son la seguridad de la capa de transporte (TLS) que se utiliza en las conexiones https.
- **Blockchain**: Muchas criptomonedas, como Bitcoin, Ethereum, utilizan la criptografía de curva elíptica para generar pares de claves públicas y privadas, así como para firmar transacciones. ECC ofrece la seguridad criptográfica necesaria para garantizar la integridad de las redes de cadena de bloques.
- **Tarjetas inteligentes y sistemas integrados**: ECC se utiliza comúnmente para proteger los sistemas de pago, los sistemas de control de acceso, los pasaportes electrónicos y otras aplicaciones que requieren soluciones criptográficas seguras y compactas.

Bibliografía

- ¿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
- 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

- ¿Qué es una clave criptográfica? | Claves y encriptación SSL | Cloudflare. (s. f.). https://www.cloudflare.com/es-es/learning/ssl/what-is-a-cryptographic-key/

- Sergio, C. C. L., Leticia, B. C., Pablo, S. A. J., Sergio, C. C. L., Leticia, B. C., & Pablo, S. A. J. (s. f.). Estudio comparativo de los algoritmos de criptografía simétrica AES, 3DES y ChaCha20. http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892022000100283

- colaboradores de Wikipedia. (2024c, mayo 1). Advanced Encryption Standard. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Advanced_Encryption_Standard

- colaboradores de Wikipedia. (2024b, febrero 11). RSA. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/RSA