

Criptografía de Curvas Elípticas

Armando Faz Hernandez
armfazh@cloudflare.com

26 de septiembre de 2025

CatioCrypto 2025



1. Criptografía simétrica vs. asimétrica
2. Curvas elípticas
3. Aplicaciones
4. Caminos a seguir

Criptografía simétrica

Comunicación segura

- Alicia y Beto se comunican mediante un canal inseguro.
- Carlos tiene acceso al medio de comunicación.
 - Sin ser posible la lectura o modificación de los mensajes.



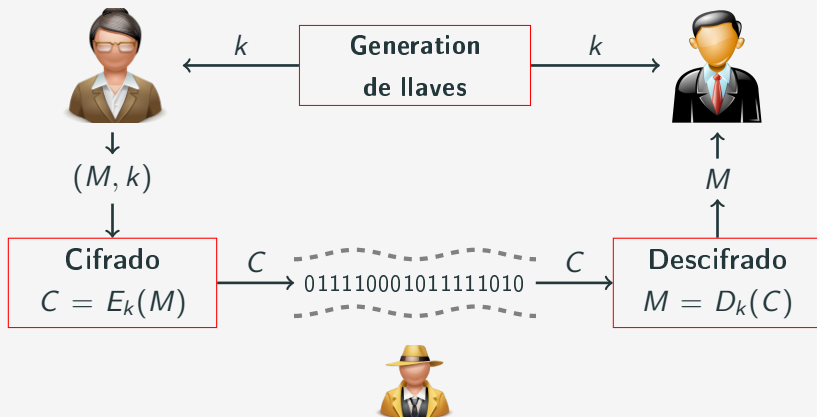
0111100001100010101011111010



Cifrado simétrico

Dada una llave secreta k , Alicia y Beto pueden intercambiar **mensajes cifrados**.

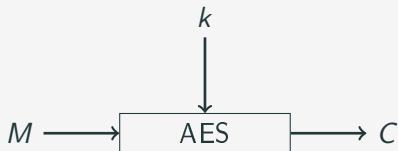
Carlos no puede leer los mensajes sin conocimiento de la llave k .



Cifrado simétrico

AES (*Advanced Encryption Standard*) es el estándar para cifrado simétrico de datos.

- Cifra mensajes de 128 bits (M) produciendo cifras de 128 bits (C) con el uso de una llave k .

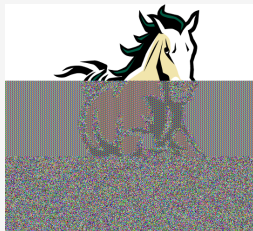


- Existen 3 variantes de AES de acuerdo al tamaño de la llave.
 - AES-128
 - AES-192
 - AES-256

Recomendación: Dejar de usar DES o RC4.

¡No es seguro!

Dividir el mensaje en bloques de 128 bits y cifrar cada bloque independientemente. (Modo ECB)



Modos de operación permiten cifrar mensajes de tamaño arbitrario usando un cifrador por bloques.

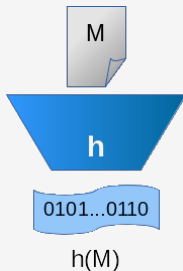
- CBC. Cifrado encadenado de bloques.
- CTR. Modo contador.
- GCM. Modo contador de Galois. (Cifrado autenticado)

Funciones de resumen

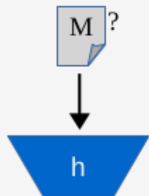
Las funciones de resumen mapean cadenas de bits de tamaño arbitrario en cadenas de bits de tamaño fijo.

$$h: \{0, 1\}^* \rightarrow \{0, 1\}^n$$

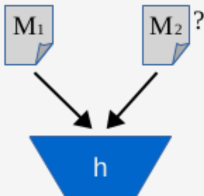
Se le conoce como **resumen** o **valor hash** a la salida de una función de resumen.



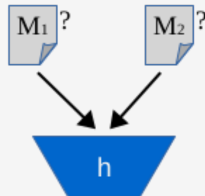
Funciones de resumen criptográfico



$$r = h(M)$$



$$h(M_1) = h(M_2)$$



$$h(M_1) = h(M_2)$$

1era Preimagen. Dado un resumen r encontrar un mensaje M tal que $r = h(M)$.

2nda Preimagen. Dado un mensaje M_1 encontrar un mensaje diferente M_2 tal que $h(M_1) = h(M_2)$.

Resistencia a colision. Encontrar dos mensajes diferentes M_1, M_2 tal que $h(M_1) = h(M_2)$.

Funciones de resumen criptográfico

SHA (Secure Hash Algorithm) es un conjunto de estándares para funciones de resumen criptográfico.

Año	Función	Salida (bits)
1993	SHA-0	160
1995	SHA-1	160
2001	SHA-2	224, 256, 384, 512
2015	SHA-3	224, 256, 384, 512

Funciones con salida extendible generan resúmenes de tamaño arbitrario.

- SHAKE128
- SHAKE256

Criptografía de llave pública

Deficiencias de la criptografía simétrica

- Se necesita de comunicación previa para establecer una llave compartida.
- Se necesita una llave para cada par de entidades.
 - En un grupo de n entidades, el número total de llaves es

$$\frac{n(n-1)}{2}$$

- Las entidades que comparten una llave poseen las mismas capacidades para cifrar y descifrar.
 - Alicia puede engañar a Beto enviando mensajes que supuestamente son de Beto.

Algunos de estos problemas los resuelve la **criptografía de llave pública**.

Criptografía de llave pública

En 1976, Diffie y Hellman observan que existen fenómenos que presentan una asimetría natural.

Ciertas acciones son fáciles de realizar pero difíciles de revertir

- Es fácil romper un vaso, pero es difícil reconstruirlo a partir de los pedazos.
- Es fácil multiplicar dos números primos, pero es difícil saber los primos que componen su producto.



La invención de la criptografía de llave pública revolucionó el campo de la criptografía.

Teoría de Grupos

Un grupo es un conjunto no vacío G junto con una operación binaria \star , y lo denotamos como (G, \star)

Cerradura Para todo $a, b \in G$, se cumple $a \star b \in G$.

Asociatividad Para todo $a, b, c \in G$, se cumple

$$(a \star b) \star c = a \star (b \star c).$$

Identidad Existe $e \in G$, llamada la identidad, tal que $e \star a = a$.

Inversos Para todo $a \in G$, existe $b \in G$, tal que $a \star b = e$.

El grupo es conmutativo si para todo $a, b \in G$, se cumple $a \star b = b \star a$.

Si en un grupo (G, \star) existe un subconjunto H de G que usando la misma operación binaria satisface los axiomas de grupo, entonces (H, \star) es un subgrupo de G .

Ejemplos de grupos

Conjunto	Operación	Identidad
Enteros (\mathbb{Z})	+	0
Reales (\mathbb{R})	\times	1
Matrices cuadradas	+	$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$
Horas de reloj	+	0h

¿Por qué (\mathbb{Z}, \times) no es un grupo?

Operación ($a, b \in G$)	Notación	
	Multiplicativa	Aditiva
$a \star b$	$ab, a \times b$	$a + b$
$a \star a$	a^2	$2a$
Inverso de a	a^{-1}	$-a$
$a \star \cdots \star a$ con k términos	a^k	ka
Identidad	$1 = a^0$	$0 = 0a$

Consecuentemente, los términos *exponenciación de grupo* y *multiplicación escalar* son usados indistintamente.

Si el grupo es conmutativo se acostumbra usar la notación aditiva.

Un grupo es **cíclico** si puede ser generado por un elemento del grupo.

Denotamos por $\langle g \rangle$ al grupo generado por un $g \in G$:

$$\langle g \rangle = \{g^k \mid k \in \mathbb{Z}\}$$

¡Atención!

- El **orden** de G es el numero de elementos en G .
- El **orden** de g es el numero de elementos en $\langle g \rangle$.

Logaritmo discreto

Sea G un grupo cíclico generado por g .

Para todo $h \in G$, existe un entero k tal que

$$h = g^k$$

Logaritmo discreto de h con respecto a g es la operación que dados g, h obtiene

$$k = \log_g(h) \in \mathbb{Z}$$

Puesto que g genera exactamente $n = |\langle g \rangle|$ elementos, entonces $k \in \{0, \dots, n-1\}$.

El problema del logaritmo discreto (DLP)

DLP: Dados $g, h \in G$ tal que $h = g^k$, encontrar k .

Existen algoritmos para resolver DLP:

- Paso de bebé, paso de gigante.
- Cálculo del índice.
- Algoritmo de Pohlig-Hellman.
- Algoritmo Rho de Pollard.
- Algoritmo de Shor.

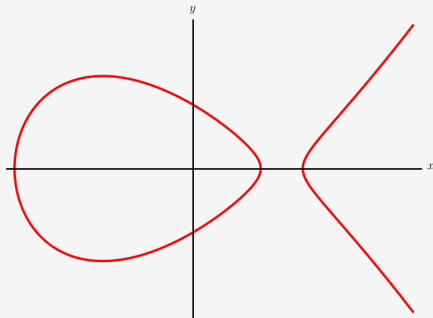
Curvas Elípticas

Curvas elípticas

Una curva elíptica está definida sobre un cuerpo k por la ecuación:

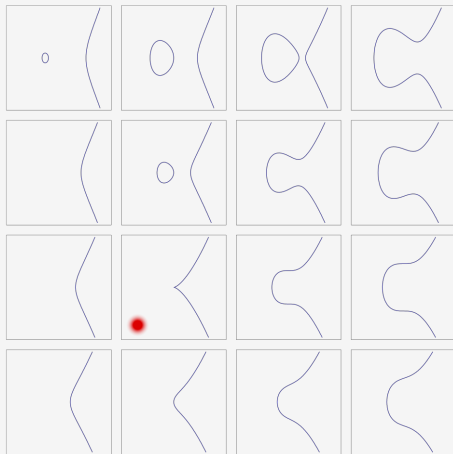
$$E/k: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

donde $a_1, a_2, a_3, a_4, a_6 \in k$ y su discriminante debe ser $\Delta \neq 0$.



Curvas elípticas

Todas son curvas elípticas, excepto una:



- no es una curva elíptica, su discriminante la delata.

Dependiendo de la característica de k , la ecuación se simplifica:

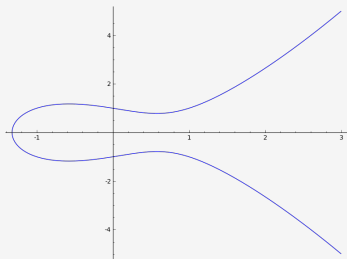
$$E/k: \begin{cases} y^2 + xy = x^3 + ax^2 + b, & \text{if } \text{char}(k) = 2, \\ y^2 = x^3 + ax^2 + b, & \text{if } \text{char}(k) = 3, \\ y^2 = x^3 + ax + b, & \text{if } \text{char}(k) \neq 2, 3. \end{cases}$$

Trabajaremos en cuerpos finitos, por lo tanto no consideramos cuando $\text{char}(k) = 0$.

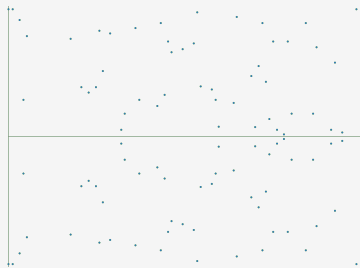
Curvas elípticas

Curva elíptica en diferentes cuerpos:

$$E: y^2 = x^3 - x + 1$$



E/\mathbb{R}



E/\mathbb{F}_{97}

Ver animación [!\[\]\(de95854c7ee024cfadc48187bbb781b2_img.jpg\)](#)

Curvas elípticas

Existen modelos que representan algunas curvas elípticas:

Weierstrass Modelo más general

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

Montgomery Curvas con un punto de orden 2

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

Twisted Edwards Curvas con un punto de orden 4

$$ax^2 + y^2 = 1 + dx^2y^2$$



Otros modelos: Twisted Hessian, intersecciones de Jacobi, Kummer, etc...

¿Cuántos puntos hay en una curva elíptica?

El teorema de Hasse provee un intervalo sobre la cantidad de puntos (N) de una curva elíptica E sobre \mathbb{F}_q :

$$|N - (q + 1)| \leq 2\sqrt{q}$$

Existen algoritmos para:

- Hallar N dados la curva E y el cuerpo \mathbb{F}_q . 
- Hallar la curva E dado N y el cuerpo \mathbb{F}_q . 

El conjunto de puntos racionales sobre k de la curva E se denota como

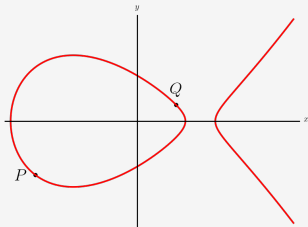
$$E(k) = \{(x, y) \in E\} \cup \{O\}$$

donde O es un punto al infinito.

Los puntos de la curva forman un grupo $(E(k), +)$ con O como la identidad del grupo.

Suma de puntos

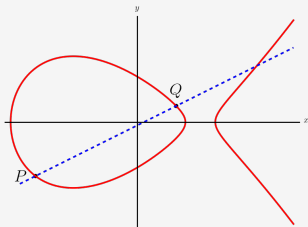
Sean P, Q puntos en la curva, se puede definir $P + Q$ mediante una construcción geométrica:



- Trace la línea que pasa por P y Q .
- Esta línea intersectará a la curva en un punto, sea R .
- Trace la línea vertical que pasa por R .
- Definimos como $P + Q$ al punto donde esta línea interseca a la curva.

Suma de puntos

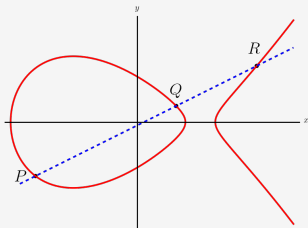
Sean P, Q puntos en la curva, se puede definir $P + Q$ mediante una construcción geométrica:



- Traze la línea que pasa por P y Q .
- Esta línea intersectará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $P + Q$ al punto donde esta línea interseca a la curva.

Suma de puntos

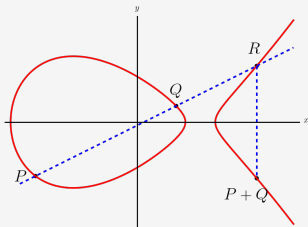
Sean P, Q puntos en la curva, se puede definir $P + Q$ mediante una construcción geométrica:



- Traze la línea que pasa por P y Q .
- Esta línea intersecará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $P + Q$ al punto donde esta línea interseca a la curva.

Suma de puntos

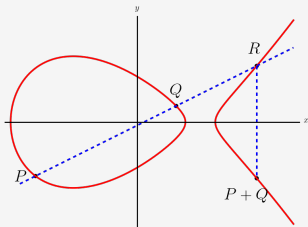
Sean P, Q puntos en la curva, se puede definir $P + Q$ mediante una construcción geométrica:



- Trace la línea que pasa por P y Q .
- Esta línea intersectará a la curva en un punto, sea R .
- Trace la línea vertical que pasa por R .
- Definimos como $P + Q$ al punto donde esta línea interseca a la curva.

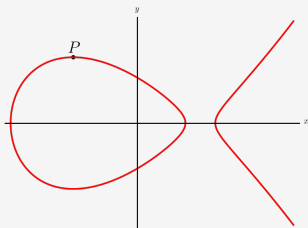
Suma de puntos

Sean P, Q puntos en la curva, se puede definir $P + Q$ mediante una construcción geométrica:



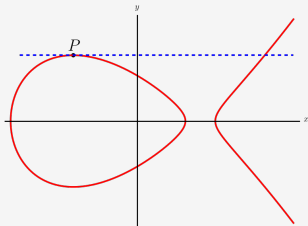
- Trace la línea que pasa por P y Q .
- Esta línea intersectará a la curva en un punto, sea R .
- Trace la línea vertical que pasa por R .
- Definimos como $P + Q$ al punto donde esta línea interseca a la curva.

Caso especial: Sumar $P + P$.



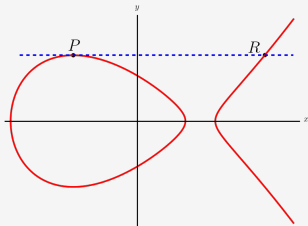
- Traze la línea tangente a la curva en P .
- Esta línea intersectará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $2P$ al punto donde esta línea interseca a la curva.

Caso especial: Sumar $P + P$.



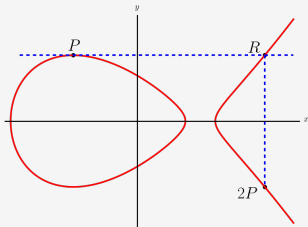
- Traze la línea tangente a la curva en P .
- Esta línea intersectará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $2P$ al punto donde esta línea interseca a la curva.

Caso especial: Sumar $P + P$.



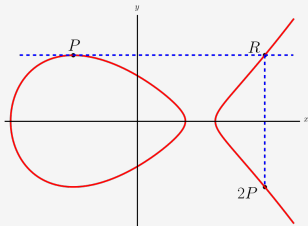
- Traze la línea tangente a la curva en P .
- Esta línea intersecará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $2P$ al punto donde esta línea interseca a la curva.

Caso especial: Sumar $P + P$.



- Traze la línea tangente a la curva en P .
- Esta línea intersecará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $2P$ al punto donde esta línea interseca a la curva.

Caso especial: Sumar $P + P$.



- Traze la línea tangente a la curva en P .
- Esta línea intersecará a la curva en un punto, sea R .
- Traze la línea vertical que pasa por R .
- Definimos como $2P$ al punto donde esta línea interseca a la curva.

Dado un punto $P \in E$ y un escalar $k \in \mathbb{Z}$, la multiplicación escalar es

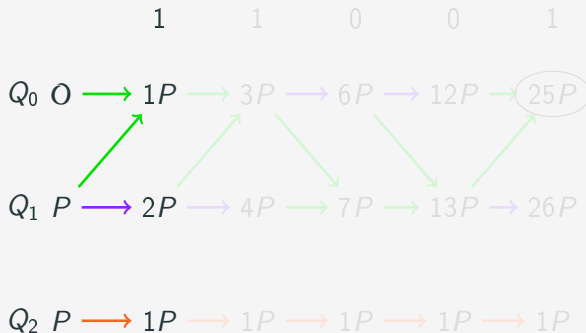
$$kP = \underbrace{P + P + \dots + P}_{k \text{ términos}}$$

Existen algoritmos con complejidad lineal $O(n)$ para calcular kP , donde $n = \log(k)$.

Escaleras para multiplicación escalar

El algoritmo de Montgomery recorre los bits del más al menos significativo.

Ejemplo: $k = (25)_{10} = (11001)_2$

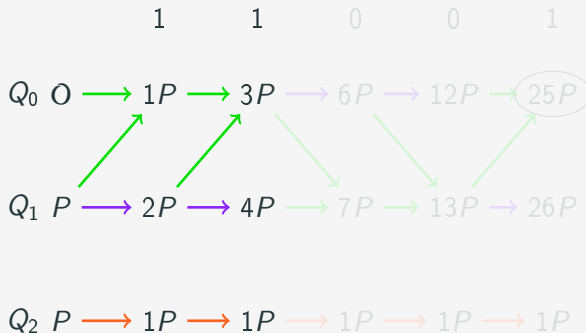


izq. \rightarrow derecha

Escaleras para multiplicación escalar

El algoritmo de Montgomery recorre los bits del más al menos significativo.

Ejemplo: $k = (25)_{10} = (11001)_2$

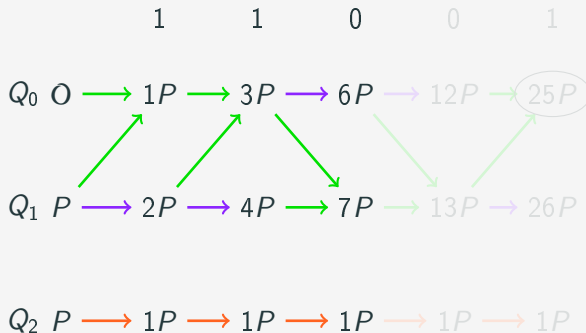


izq. \rightarrow derecha

Escaleras para multiplicación escalar

El algoritmo de Montgomery recorre los bits del más al menos significativo.

Ejemplo: $k = (25)_{10} = (11001)_2$

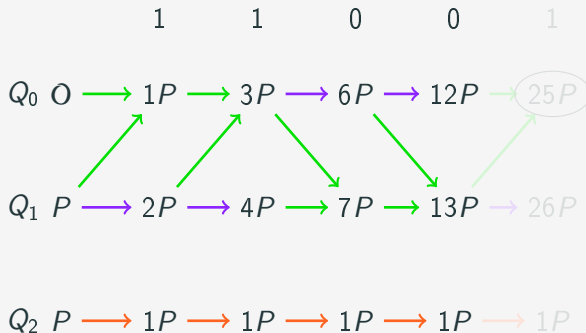


izq. \rightarrow derecha

Escaleras para multiplicación escalar

El algoritmo de Montgomery recorre los bits del más al menos significativo.

Ejemplo: $k = (25)_{10} = (11001)_2$

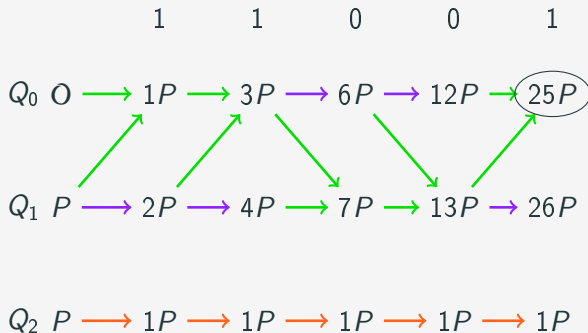


izq. \rightarrow derecha

Escaleras para multiplicación escalar

El algoritmo de Montgomery recorre los bits del más al menos significativo.

Ejemplo: $k = (25)_{10} = (11001)_2$



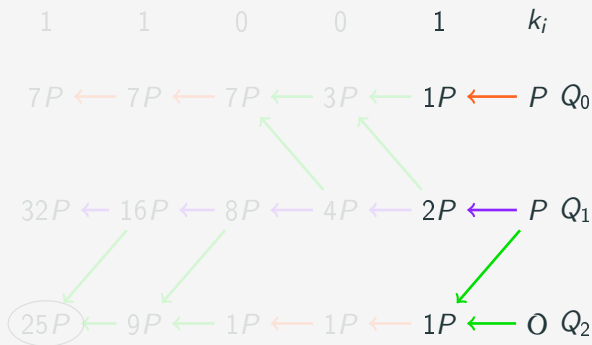
izq. \rightarrow derecha

Escaleras para multiplicación escalar

El algoritmo de Joye recorre los bits del menos al más significativo.



izq. \leftarrow derecha

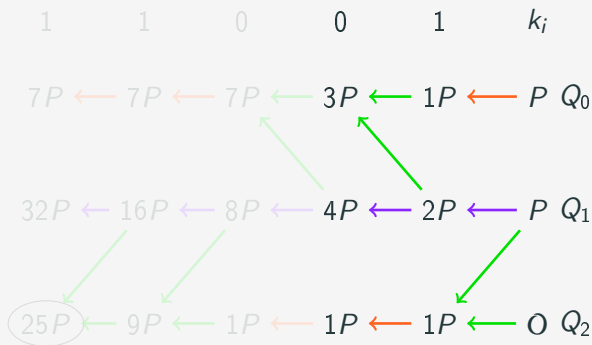


Escaleras para multiplicación escalar

El algoritmo de Joye recorre los bits del menos al más significativo.



izq. \leftarrow derecha

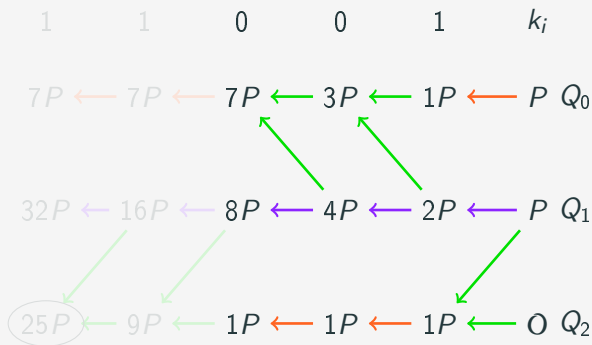


Escaleras para multiplicación escalar

El algoritmo de Joye recorre los bits del menos al más significativo.



izq. \leftarrow derecha

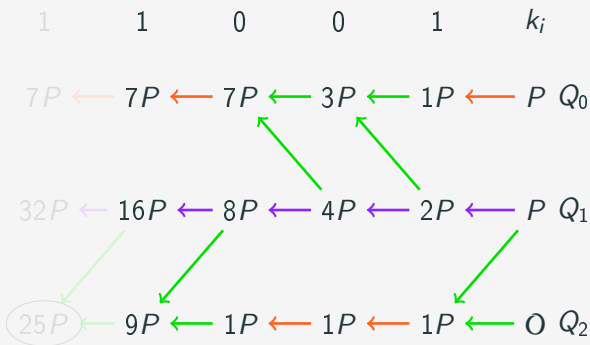


Escaleras para multiplicación escalar

El algoritmo de Joye recorre los bits del menos al más significativo.



izq. \leftarrow derecha

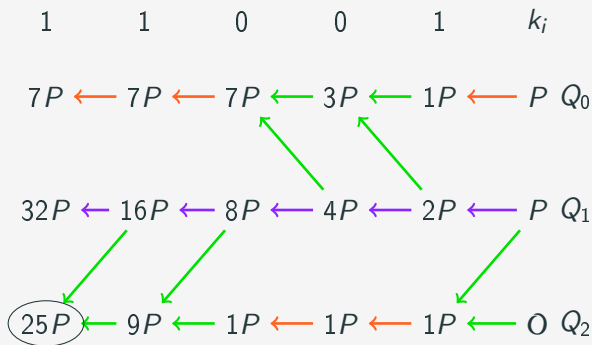


Escaleras para multiplicación escalar

El algoritmo de Joye recorre los bits del menos al más significativo.



izq. \leftarrow derecha



Escaleras para multiplicación escalar



- Existe una dualidad entre estos algoritmos. [↗](#)
- Ambos mantienen la misma invariante

$$Q_2 = Q_1 - Q_0$$

Problema del logaritmo discreto (ECDLP)

ECDLP: Dados $P, Q \in E$, tal que $Q = kP$, la tarea es encontrar k .

- El algoritmo de Pollard es el mejor algoritmo conocido para resolver ECDLP cuya complejidad es

$$O\left(\sqrt{\#E(\mathbb{F}_p)}\right)$$

donde $\#E(\mathbb{F}_p) \approx p$ es el número de puntos de la curva.

- Por ejemplo: para una curva elíptica sobre \mathbb{F}_p con $p \approx 2^{256}$, se requiere de 2^{128} operaciones para resolver ECDLP.

Criptografía de Curvas Elípticas

- En 1985, tanto Koblitz como Miller sugirieron el uso de curvas elípticas para criptografía.
- Ya que brinda un nivel de seguridad equivalente a RSA usando **llaves más pequeñas**.
Por ejemplo, para el nivel de seguridad de 128 bits.
 - RSA usa llaves de 3,072 bits
 - ECC usa llaves de 256 bits.
- Montgomery utilizó las curvas para criptoanálisis: factorización de enteros.

Protocolo Diffie-Hellman con Curvas de Montgomery

El RFC-7748 describe dos funciones para calcular un secreto compartido:

- **X25519** usa llaves de 32 bytes.
- **X448** usa llaves de 56 bytes.



$$a \xleftarrow{\$} \{0, 1\}^{256}$$



$$b \xleftarrow{\$} \{0, 1\}^{256}$$

$$K_A \leftarrow \text{X25519}(9, a)$$

$$K_B \leftarrow \text{X25519}(9, b)$$

$$K = \text{X25519}(K_B, a)$$

$$K = \text{X25519}(K_A, b)$$

K es la llave compartida.

Internamente la función X calcula una multiplicación escalar.

El RFC-8032 describe dos instancias de EdDSA.

- Ed25519
- Ed448.

EdDSA consiste de tres algoritmos.

- Generación de llaves.
- Firmar.
- Verificar.

Calcular la llave privada sk y la llave pública pk

- 1: $sk \in_R [0, \ell)$
- 2: $h = (h_{2b-1}, \dots, h_0)_2 \leftarrow H(sk)$
- 3: $a \leftarrow 2^n + \sum 2^i h_i, \quad \text{for } c \leq i < n$
- 4: $pk \leftarrow aB$
- 5: **return** (sk, pk)

Dado un mensaje M y el par de llaves (sk, pk)
calcula la firma (R, S) como

$$1: h = (\underbrace{h_{2b-1}, \dots, h_b}_{h_H}, \underbrace{h_{b-1}, \dots, h_0}_{h_L})_2 \leftarrow H(sk)$$

$$2: a \leftarrow 2^n + \sum 2^i h_i, \quad \text{for } c \leq i < n$$

$$3: r \leftarrow H(h_H \parallel M) \pmod{\ell}$$

$$4: R' \leftarrow rB$$

$$5: R \leftarrow \text{Codifica}(R')$$

$$6: S \leftarrow r + H(R \parallel pk \parallel M) \cdot a \pmod{\ell}$$

$$7: \text{return } (R, S)$$

Dado un mensaje M , una firma (R, S) y una llave pública pk :

$$P \leftarrow \text{Decodifica}(pk)$$

$$h \leftarrow H(R \parallel pk \parallel M) \pmod{\ell}$$

Acepte la firma si se cumple lo siguiente:

- $P \in \mathcal{E}_d(\mathbb{F}_p)$
- $S \in [0, \ell)$
- $SB = R + hP$

Maleabilidad en transacciones de Bitcoin

Bitcoin usa firmas ECDSA, en éstas tanto (r, s) como $(r, -s \bmod n)$ son firmas válidas del mensaje M .

Ataque:



Alicia firma una transacción (tx-id1) y la envía a través de la red.



Beto crea otra transacción (tx-id2) usando la otra firma válida.

Si tx-id2 se incluye primero en un bloque, Alicia pierde su transacción.

Solución: Declara una firma como válida sólo si $s < \frac{n}{2}$. [BIP 66]

Maleabilidad en transacciones de Bitcoin

Bitcoin usa firmas ECDSA, en éstas tanto (r, s) como $(r, -s \bmod n)$ son firmas válidas del mensaje M .

Ataque:



Alicia firma una transacción (tx-id1) y la envía a través de la red.



Beto crea otra transacción (tx-id2) usando la otra firma válida.

Si tx-id2 se incluye primero en un bloque, Alicia pierde su transacción.

Solución: Declara una firma como válida sólo si $s < \frac{n}{2}$. [BIP 66]

Maleabilidad en transacciones de Bitcoin

Bitcoin usa firmas ECDSA, en éstas tanto (r, s) como $(r, -s \bmod n)$ son firmas válidas del mensaje M .

Ataque:



Alicia firma una transacción (tx-id1) y la envía a través de la red.



Beto crea otra transacción (tx-id2) usando la otra firma válida.

Si tx-id2 se incluye primero en un bloque, Alicia pierde su transacción.

Solución: Declara una firma como válida sólo si $s < \frac{n}{2}$. [BIP 66]

Aplicaciones

Protocolo Diffie-Hellman usando certificados



$$\begin{aligned} k_A &\stackrel{\$}{\leftarrow} [1, \ell - 1] \\ P_A &\leftarrow k_A G \end{aligned}$$

Parámetros

$$p, \ell, \langle G \rangle = E(\mathbb{F}_p), CA_{pk}$$

Generación de llaves



$$\begin{aligned} k_B &\stackrel{\$}{\leftarrow} [1, \ell - 1] \\ P_B &\leftarrow k_B G \end{aligned}$$

Protocolo Diffie-Hellman usando certificados



Parámetros

$$p, \ell, \langle G \rangle = E(\mathbb{F}_p), CA_{pk}$$



$$k_A \xleftarrow{\$} [1, \ell - 1]$$

$$P_A \leftarrow k_A G$$

$$Cert_A = \text{Firmar}(P_A, CA_{sk})$$

CA emite un certificado

$$k_B \xleftarrow{\$} [1, \ell - 1]$$

$$P_B \leftarrow k_B G$$

$$Cert_B = \text{Firmar}(P_B, CA_{sk})$$

Protocolo Diffie-Hellman usando certificados



Parámetros
 $p, \ell, \langle G \rangle = E(\mathbb{F}_p), CA_{pk}$



$$k_A \xleftarrow{\$} [1, \ell - 1]$$

$$P_A \leftarrow k_A G$$

$$Cert_A = \text{Firmar}(P_A, CA_{sk})$$

$$k_B \xleftarrow{\$} [1, \ell - 1]$$

$$P_B \leftarrow k_B G$$

$$Cert_B = \text{Firmar}(P_B, CA_{sk})$$

$\{Cert_B, P_B\}$

$\{Cert_A, P_A\}$

Protocolo Diffie-Hellman usando certificados



Parámetros
 $p, \ell, \langle G \rangle = E(\mathbb{F}_p), CA_{pk}$



$$k_A \xleftarrow{\$} [1, \ell - 1]$$

$$P_A \leftarrow k_A G$$

$$Cert_A = \text{Firmar}(P_A, CA_{sk})$$

$$k_B \xleftarrow{\$} [1, \ell - 1]$$

$$P_B \leftarrow k_B G$$

$$Cert_B = \text{Firmar}(P_B, CA_{sk})$$

$\{Cert_B, P_B\}$

Verificar Certificado

$\{Cert_A, P_A\}$

$$\text{Verificar}(Cert_B, CA_{pk}) = \text{Si}$$

$$\text{Verificar}(Cert_A, CA_{pk}) = \text{Si}$$

Protocolo Diffie-Hellman usando certificados



Parámetros
 $p, \ell, \langle G \rangle = E(\mathbb{F}_p), CA_{pk}$



$$\begin{aligned} k_A &\xleftarrow{\$} [1, \ell - 1] \\ P_A &\leftarrow k_A G \\ \text{Cert}_A &= \text{Firmar}(P_A, CA_{sk}) \end{aligned}$$

$$\begin{aligned} k_B &\xleftarrow{\$} [1, \ell - 1] \\ P_B &\leftarrow k_B G \\ \text{Cert}_B &= \text{Firmar}(P_B, CA_{sk}) \end{aligned}$$

$$\begin{aligned} &\{\text{Cert}_B, P_B\} \\ \text{Verificar}(\text{Cert}_B, CA_{pk}) &= \text{Si} \\ P_{AB} &\leftarrow k_A P_B \end{aligned}$$

$$\begin{aligned} &\{\text{Cert}_A, P_A\} \\ \text{Verificar}(\text{Cert}_A, CA_{pk}) &= \text{Si} \\ P_{AB} &\leftarrow k_B P_A \end{aligned}$$

Protocolo Diffie-Hellman usando certificados



Parámetros
 $p, \ell, \langle G \rangle = E(\mathbb{F}_p), CA_{pk}$



$$k_A \xleftarrow{\$} [1, \ell - 1]$$

$$P_A \leftarrow k_A G$$

$$Cert_A = \text{Firmar}(P_A, CA_{sk})$$

$$k_B \xleftarrow{\$} [1, \ell - 1]$$

$$P_B \leftarrow k_B G$$

$$Cert_B = \text{Firmar}(P_B, CA_{sk})$$

$$\{Cert_B, P_B\}$$

$$\text{Verificar}(Cert_B, CA_{pk}) = \text{Si}$$

$$P_{AB} \leftarrow k_A P_B$$

$$\{Cert_A, P_A\}$$

$$\text{Verificar}(Cert_A, CA_{pk}) = \text{Si}$$

$$P_{AB} \leftarrow k_B P_A$$

$$k_A k_B G = P_{AB} = k_B k_A G \text{ es el secreto compartido.}$$

TLS (*Transport Layer Security*) es un protocolo de seguridad que ofrece privacidad e integridad de datos para las comunicaciones en Internet.

TLS 1.3 es la versión más reciente.

Versiones previas: TLS 1.1, TLS 1.2, SSLv3



Secure

https://example.com

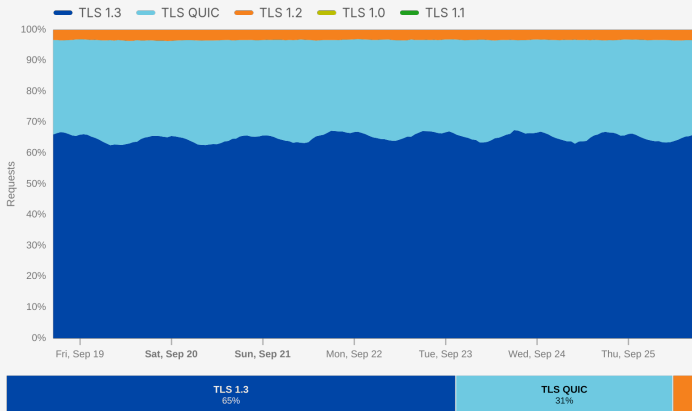


Versiones de TLS en uso

HTTP requests by TLS version time series

Distribution of HTTP requests by TLS version over time

Bot class: Likely human



Cloudflare Radar

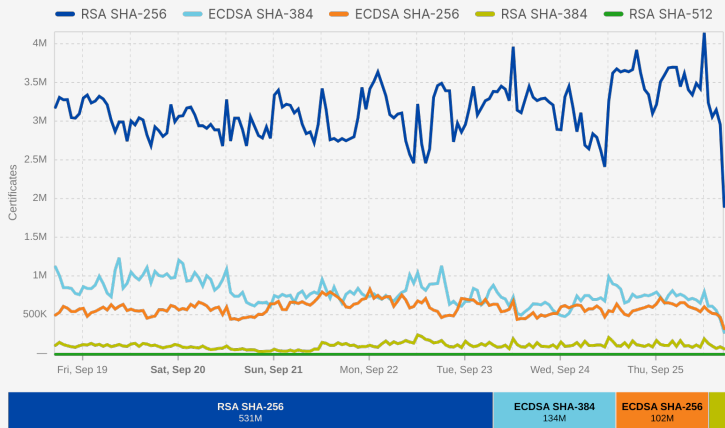
Last 7 days | Sep 26, 2025, 06:30 UTC

Fuente: Cloudflare Radar [↗](#)

Algoritmos de firma en los certificados

Certificates by signature algorithm time series

Distribution of certificates by signature algorithm over time



- Protocolos de Internet
 - DNS y DNSSEC
 - TLS y PKI
 - SSH
 - IPSec
- Cifrado de extremo a extremo.
- Blockchain
 - Contratos inteligentes.
 - Máquinas virtuales.
 - Computación verificable.
- Autenticación. Documentos de identidad.
- Autorización. Credenciales anónimas.
- Pruebas de conocimiento.
- Criptoanálisis: Factorización de enteros.

Caminos a seguir

¿Qué debería estudiar después?

Geometría Algoritmos de hashing a curvas, busca modelos alternativos de curvas (hiper-)elípticas.

Álgebra: Algoritmos de multiplicación escalar y suma de puntos.

Protocolos: Funciones pseudo-aleatorias, intersección de conjuntos, pruebas de conocimiento.

Computación: Desarrollo seguro de bibliotecas: BoringSSL, Go, rust, PyCA.

Electrónica: Ataques físicos y de canal lateral.

Ninguno de los anteriores: persiste en las cosas que has aprendido y de las cuales te convenciste, sabiendo de quiénes las has aprendido.

- [!\[\]\(a22ba4e13c745edbf29e51af246c4c12_img.jpg\) *Guide to Elliptic Curve Cryptography*](#) por Hankerson, Vanstone, Menezes.
- [!\[\]\(33b18af9a4b997eb52666cfeb3c44157_img.jpg\) *Handbook of Elliptic and Hyperelliptic Curve Cryptography*](#) por Cohen y Frey.
- [!\[\]\(262b158440b847a82f89a14cab8644ec_img.jpg\) *The Arithmetic of Elliptic Curves*](#) por Silverman.
- [!\[\]\(f51929fecf7b0dc947ac13f4c4835e8f_img.jpg\) *Introduction to Elliptic Curves and Modular Forms*](#) por Koblitz.
- [!\[\]\(dfbf0e54bcca114319aa65c906feb8d0_img.jpg\) *Elliptic Curve Cryptography for Developers*](#) por Rosing.

Lo bueno Las curvas elípticas son el **presente** de la criptografía.

- Hay mucho trabajo por realizar aún.
- Tanto en lo académico como en lo industrial.

Lo malo ECDLP se resuelve con el algoritmo cuántico de Shor

- Aún no hay computadoras cuánticas relevantes que rompan la criptografía actual.

Lo feo *Cosecha hoy, descifra mañana.* Debemos migrar la seguridad de nuestros sistemas a esquemas post-cuánticos híbridos.

Lo mejor Las curvas elípticas pueden ser el **futuro de la criptografía**.

- Hallar isogenías entre curvas elípticas es difícil incluso con cómputo cuántico. ¡Qué chimba!

¡Muchas gracias por su atención!



Fuente [🔗](#)

Disfruten de CatioCrypt, ASCrypto y Latincrypt 2025.

Criptografía de Curvas Elípticas

Armando Faz Hernandez

armfazh@cloudflare.com

26 de septiembre de 2025

CatioCrypto 2025

