Criptografía sobre Curvas Elípticas.
Diploma en Matemática Mención Aplicaciones.
(IPES-UdelaR)
Prof. Horacio Castagna.

Aquí presentamos las ideas centrales desarrolladas en el trabajo final presentado para el Diploma, cuyo título es Criptografía sobre Curvas Elípticas, tutoreado por Dr. José Vieitez.

Los temas vinculados a la seguridad en el ciberespacio y las comunicaciones son de preocupación mundial. La idea central del trabajo es presentar de manera sencilla las ideas básicas que están detrás de los métodos involucrados en la protección de las comunicaciones y los datos. Para ello trataremos los aspectos teóricos referentes a las curvas elípticas, y cómo éstas pueden usarse para implementar algoritmos de codificado, cifrado y firma digital de documentos. Luego, para ilustrar y comprender el funcionamiento de ciertos algoritmos, se construye un ejemplo de "juguete" con la implementación de los mismos en un lenguaje de alto nivel.

1. Curva Elíptica y estructura de Grupo Abeliano.

Dado un cuerpo K definimos curva elíptica al conjunto de puntos $(x, y) \in K^2$ que verifican la ecuación:

$$y^2 + a_{11}xy + a_{01}y = x^3 + a_{20}x^2 + a_{10}x + a_{00}$$

Si la característica del cuerpo es distinta de 2 y de 3 podemos llevar la ecuación anterior a la forma:

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

De ahora en adelante trabajaremos con esta forma denominada Ecuación de Weierstrass.

Para poder definir una operación "suma de puntos" sobre la curva elíptica, necesitamos que la curva no sea singular. Esto se logra si su discriminante no es nulo, o sea:

$$\Delta = 4a^3 + 27b^2 \neq 0$$

Tomemos el punto O ("punto en el infinito" según la dirección del eje Oy) y el conjunto:

$$E(K) = \{(x, y) \in K^2 : y^2 = x^3 + ax + b\} \cup \{O\}$$

Definamos ahora $+ : E(K) \times E(K) \to E(K)$ de la siguiente manera:

i) Si $P, Q \in E(K)$ son distintos de O y con abscisas distintas, tomamos la recta PQ y la cortamos con $E(K)$, y al punto distinto de P y de Q así obtenido lo simetrizamos respecto de Ox y obtenemos $P + Q$.

ii) Si $P, Q \in E(K)$ son distintos de O y con abscisas iguales, o $P = Q$ con ordenadas 0, definimos $P + Q = O$.

iii) Si $P, Q \in E(K)$ con $P = Q$ y ordenadas distintas de 0, tomamos la recta tangente e $E(K)$ por P, la cortamos con $E(K)$, al punto distinto de P así obtenido lo simetrizamos respecto de Ox y obtenemos $P + Q$.

iv) $P + O = P$, $\forall P \in E(K)$.

A partir de lo definido obtenemos una estructura de grupo abeliano, considerando el conjunto $E(K)$ y la operación $+$, sea $(E(K), +)$.

2. Criptografía.

Desde tiempos remotos ha existido información sensible que se desea compartir entre dos personas, a la cual no tengan acceso los demás. Así nace la Criptología (del griego krypto: "oculto" y logos: "palabra"), que es la disciplina científica que se dedica al estudio de la escritura secreta.

La Criptografía, que podemos verla como parte de la Criptología, se ocupa del estudio de los algoritmos, protocolos y sistemas que se utilizan para proteger la información, y dar seguridad en las comunicaciones. Actualmente se realizan infinidad de comunicaciones y transacciones que se desean preservar de ojos y oídos ajenos (cliente-banco, compras por internet, correos electrónicos, chats, redes sociales, comercio, etc.).

Distinguimos dos grandes tipos de enfoque:

i) Criptografía de clave privada o simétrica. Sistema de cifrar que utiliza una clave solamente conocida por aquellos usuarios autorizados, que deben ponerse de acuerdo en la clave previamente a comunicarse.

ii) Criptografía de clave pública o asimétrica. Sistema de cifrar que utiliza una clave pública para un usuario, de modo que solamente él pueda descifrar el mensaje.

Para el trabajo, nos enfocaremos en los algoritmos de clave pública.

Los mismos intentan resolver los siguientes problemas:

i) Confidencialidad. Un documento debe poder leerlo sólo el receptor al que le fue enviado y nadie más.

ii) Autenticidad. El documento proviene, verdaderamente, de quien dice provenir.

iii) Integridad. El emisor y el receptor del documento deben estar seguros de que dicho documento no fue alterado.

iv) No repudio. El emisor envió el documento y no puede decir que no lo envió.

Según el tiempo de ejecución en función del tamaño de la entrada, podemos clasificar los algoritmos en dos tipos:

i) Complejidad Polinomial (problema fácil).

ii) Complejidad No Polinomial (problema difícil).

Para que el cifrado sea seguro (difícil de descifrar sin la clave), debemos utilizar algoritmos que se basen en complejidad no polinomial en el descifrado.

Existen varios problemas para los cuales no existen, hasta el momento, algoritmos en tiempo polinomial que los resuelvan.

Uno de ellos es el Problema del Logaritmo Discreto sobre Curvas Elípticas (PLDE).

El PLDE consiste en dados $(E(K), +)$; $P, Q \in E(K)$ con $Q = nP$ para algún $n \in \mathbb{N}$, no existen, de momento, algoritmos en tiempo polinomial para hallar n.

Presentamos a continuación distintos algoritmos que utilizan curvas elípticas, dependiendo del uso:

i) Intercambio de claves.

ii) Cifrado de mensajes completos.

iii) Firma Digital y verificación.

iv) Codificación.

2.1 Diffie-Hellman.

Este algoritmo es utilizado para el intercambio de claves.

A y B se ponen de acuerdo en utilizar cierta $E(K)$ y un punto $P \in E(K)$, en general con $K = \mathbb{Z}_p$.

A elige $n_A \in \mathbb{Z}$ y B elige $n_B \in \mathbb{Z}$. A calcula $R_A = n_A P$ y se lo envía a B. Por su parte B calcula $R_B = n_B P$ y se lo envía a A. Ahora A calcula $S = n_A R_B$ y B calcula $S = n_B R_A$, teniendo ambos el mismo S, secreto compartido.

¿Cómo funciona? Observemos que $S = n_A R_B = n_A n_B P = n_B n_A P = n_B R_A$.

2.2 El Gamal.

Este algoritmo se utiliza para el envío de mensajes cifrados.

Al igual que en el algoritmo anterior, A y B se ponen de acuerdo en utilizar cierta $E(K)$ y un punto $P \in E(K)$. Supongamos que B desea enviar a A cierto mensaje m.

A elige $n_A \in \mathbb{Z}$ y B elige $n_B \in \mathbb{Z}$. A calcula $R_A = n_A P$ y se lo envía a B. Ahora B calcula $R_B = n_B P$ y $S = m + n_B R_A$ y los envía a A.

A calcula $m = S - n_A R_B$ recuperando así m.

Veamos, $S - n_A R_B = m + n_B R_A - n_A R_B = m$, por lo que efectivamente A recupera el mensaje original m.

2.3 Elliptic Curve Digital Signature Algorithm (ECDSA).

Algoritmo para firmar digitalmente.

La forma en la que se puede asegurar la autenticación, el no repudio y la integridad en una comunicación, es mediante la introducción de la firma al documento o mensaje.

Lo más usual es utilizar primero una Función Hash (Huella), que lo que hace es, cuando aplicamos dicha función al mensaje, devolver un bloque de longitud fija. Luego este bloque se cifra y con ello obtenemos la firma y se adjunta al mensaje original.

Las propiedades que debe tener una Función Huella h, para que pueda cumplirse con la seguridad requerida son:

i) Dado un mensaje m, debe ser fácil y rápido (computacionalmente hablando) calcular $h(m)$.

ii) Dada una huella digital h_0, debe ser muy difícil (computacionalmente no existe algoritmo en tiempo polinomial) calcular x_0 de modo que $h(x_0) = h_0$. Esto se denomina resistencia a hallar una preimagen.

iii) Dados x_0 y $h(x_0) = h_0$, debe ser prácticamente imposible hallar x_1, de manera que $h(x_0) = h(x_1)$. Esto se denomina resistencia a hallar una segunda preimagen.

iv) Debe ser prácticamente imposible calcular x_0 y x_1, con $x_0 \neq x_1$ de modo que $h(x_0) = h(x_1)$. Esto se denomina resistencia a la colisión.

Entonces, una vez aplicada la función huella elegida al documento (texto, mensaje, archivo, etc.), se cifra dicha huella. Luego de cifrar la huella (firma electrónica), esta se anexa al documento, lo que produce un documento firmado, el cual se cifra, para luego enviarlo por la red a su destinatario.

Cuando el destinatario recibe el mensaje, se descifra, se obtiene el texto en claro y la firma, y pasamos a la etapa de verificación. Dicha etapa es sumamente importante, ya que es la que garantiza las propiedades de autenticación, integridad y no repudio.

El mecanismo de verificación es el siguiente, una vez descifrado el mensaje recibido, le aplicamos al texto en claro la función huella correspondiente, obtenemos la huella del texto que nos llegó. Luego con la clave pública, de firmar, del emisor, se descifra la firma y se obtiene la función huella del documento que se envió. Ahora sólo queda verificar que la huella del texto que se envió coincida con la huella del texto recibido. Si coincide, entonces el documento no ha sido alterado, preservando la integridad del mismo. Como solo la clave pública para firmar del emisor es la que descifra la firma, entonces se garantiza la autenticidad y el no repudio.

Ahora el algoritmo ECDSA.

A y B se ponen de acuerdo en utilizar cierta $E(K)$ de orden n (por seguridad $n > 2^{160}$) y un punto $P \in E(K)$.

A elige $n_A \in \mathbb{Z}$ y calcula $R_A = n_A P$, elige $k \in \mathbb{Z}$ y calcula $Q = kP$ de modo que la abscisa de Q, x_Q, sea distinta de 0.

Ahora calcula $h^{-1}{}_{(mod\, n)}$ y $h_m = h(m)$, y luego $s = k^{-1}(h_m + n_A x_Q)_{(mod\, n)}$.

Si $s \equiv 0_{(mod\, n)}$ elige k nuevamente y todo de nuevo, en caso contrario la firma es la pareja (x_Q, s). La firma se introduce en el mensaje m que se cifra para enviar a B (con El Gamal por ejemplo).

Veamos ahora como B puede verificar la firma. Descifra el documento y obtiene el mensaje m y la firma (x_Q, s). Ahora calcula $h(m) = h_m$, $V = h_m s^{-1} P + x_Q s^{-1} R_A$. Si $x_Q \equiv x_{V\ (mod\, n)}$ la firma se acepta, caso contrario se rechaza.

Veamos la justificación:

$V = h_m s^{-1} P + x_Q s^{-1} R_A = (h_m (h_m + n_A x_Q)^{-1} k)P + (x_Q (h_m + n_A x_Q)^{-1} k n_A)P = ((h_m + n_A x_Q)(h_m + n_A x_Q)^{-1} k)P = kP = Q$

2.4 Codificación.

Según lo que hemos visto hasta ahora, sabemos como cifrar y firmar un determinado documento utilizando un sistema criptográfico basado en el problema del logaritmo discreto elíptico (PLDE).

Ahora queda ver cómo se procede para realizar una equivalencia entre puntos de una curva elíptica y símbolos (caracteres y otros). De esta forma podemos asociar a un conjunto de símbolos, un conjunto de puntos sobre una curva elíptica.

Debemos realizar un procedimiento que garantice la existencia de un algoritmo, por el cual asociemos a un símbolo, uno y sólo un punto de la curva a utilizar.

Primero elegimos un alfabeto (la asignación de un entero a un símbolo, uno a uno, como ASCII o UNICODE) de N símbolos. Tomemos bloques de largo l para codificar.

Sea m_i el entero correspondiente al i-ésimo símbolo del bloque $b = (m_0 ... m_{l-1})$. Ahora elegimos k (con 50 suele ser suficiente) de modo que $kN^l < q$, siendo q el número de elementos del cuerpo K. Esto es para garantizar que dado un entero podré asociarle un punto de la elíptica con mucha probabilidad $(1 - \frac{1}{2^k})$.

Calculamos $x_b = \sum_{i=0}^{i=l-1} (m_i N^{l-1-i})$, asignamos $j = 0$. Chequeamos si existe y de modo que $(kx_b + j, y) \in E(K)$, si es así, dicho punto codifica el bloque, si no vamos incrementando j de a una unidad hasta obtener el punto de la elíptica asociado al bloque.

¿Cómo recuperar el bloque b teniendo el punto de la elíptica (x, y)?

Calculamos k^{-1} y asignamos $j = 0$. Ahora calculamos $k^{-1}(x - j)$ y chequeamos si pertenece al intervalo $[0, N^{l-1}]$, si esto ocurre hacemos $x_b = k^{-1}(x - j)$, si no incrementamos j en una unidad y repetimos.

Una vez obtenido el x_b descomponemos (por divisiones sucesivas por ejemplo) para obtener los m_i de modo que $x_b = \sum_{i=0}^{i=l-1}(m_i N^{l-1-i})$, recuperando el bloque $b = (m_0...m_{l-1})$.

3. Construcción de un ejemplo.

En el trabajo diseñamos un ejemplo donde trabajamos con el cuerpo $\mathbb{Z}_{103}$, la curva elíptica $E(\mathbb{Z}_{103}) = \{(x, y) \in \mathbb{Z}_{103}^2 : y^2 = x^3 - 2x + 1\} \cup \{O\}$ y el punto $P(65, 1)$ de la misma.

Por otro lado consideramos un alfabeto propio de 29 símbolos, con la asignación correspondiente a un punto de la elíptica, sea la siguiente tabla:

A	(0,1)	F	(20,37)	K	(30,26)	O	(36,21)	T	(41,6)	Z	(48,9)
B	(5,42)	G	(25,34)	L	(31,13)	P	(37,25)	U	(43,3)		(49,21)
C	(13,3)	H	(26,18)	M	(33,50)	Q	(38,1)	V	(44,15)	,	(51,35)
D	(15,16)	I	(27,36)	N	(34,32)	R	(39,29)	X	(46,42)	.	(52,42)
E	(18,21)	J	(28,24)	Ñ	(35,24)	S	(40,24)	Y	(47,3)		

Se utilizará El Gamal para cifrar y descifrar el mensaje deseado. También diseñamos la implementación de los algoritmos requeridos en un lenguaje de alto nivel.

Para más detalles sugerimos consultar el trabajo final.

Bibliografía.

Castagna, H. (2017). Criptografía sobre Curvas Elípticas.

Blake, I., Seroussi, G., Smart, N. (2005). Advances in Elliptic Curve Cryptography. Cambridge University Press.

Enge, A. (1999). Elliptic Curves and their Applications to Cryptography. Massachusetts, United States of America: Kluwer Academic Publishers.

Koblitz, N. (1994). A Course in Number Theory and Cryptography. New York: Springer-Verlag.

Lucena López, M. Criptografía y Seguridad en Computadores (4ª ed.). España: Universidad de Jaén.

Matheu García, S. (2015). Curvas Elípticas. Universidad de Murcia, Facultad de Matemáticas.

Medina Aparcana, R. (2012). Criptografía con curvas elípticas sobre cuerpos p-ádicos. Universidad Nacional de Ingeniería, Facultad de Ciencias, Lima.

Silverman, J. (2008). The Arithmetic of Elliptic Curves. Providence, Rhode Island: Springer.

Vieitez, J. (2015). Aritmética Modular y Criptografía. Salto, Uruguay.