MOOC de Criptología Matemática. Diffie-Hellman y Firmas Digitales

Leandro Marín

Módulo III. Sesión 4.
Dificultad Alta

Curvas Elípticas en Criptografía

- Para trabajar con curvas elípticas nos deben proporcionar los siguientes parámetros:

Curvas Elípticas en Criptografía

- Para trabajar con curvas elípticas nos deben proporcionar los siguientes parámetros:
  - Un número primo p de entre 160 y 256 bits.

Curvas Elípticas en Criptografía

- Para trabajar con curvas elípticas nos deben proporcionar los siguientes parámetros:
  - Un número primo p de entre 160 y 256 bits.
  - Los parámetros a y b tales que $y^2 = x^3 + ax + b$ sea una curva elíptica no singular sobre el cuerpo $K = \mathbb{Z}_p$ y que tenga un número de puntos n que sea un número primo.

Curvas Elípticas en Criptografía

- Para trabajar con curvas elípticas nos deben proporcionar los siguientes parámetros:
  - Un número primo p de entre 160 y 256 bits.
  - Los parámetros a y b tales que $y^2 = x^3 + ax + b$ sea una curva elíptica no singular sobre el cuerpo $K = \mathbb{Z}_p$ y que tenga un número de puntos n que sea un número primo.
  - El valor de n nos lo proporcinarán. Aunque para algunos algoritmos no es necesario conocerlo, podemos suponer que es públicamente conocido.

Curvas Elípticas en Criptografía

- Para trabajar con curvas elípticas nos deben proporcionar los siguientes parámetros:
  - Un número primo p de entre 160 y 256 bits.
  - Los parámetros a y b tales que $y^2 = x^3 + ax + b$ sea una curva elíptica no singular sobre el cuerpo $K = \mathbb{Z}_p$ y que tenga un número de puntos n que sea un número primo.
  - El valor de n nos lo proporcinarán. Aunque para algunos algoritmos no es necesario conocerlo, podemos suponer que es públicamente conocido.
  - También nos tienen que proporcionar un punto G de la curva distinto de O.

Claves Públicas y Privadas

- Para generar las claves públicas y privadas se procederá del siguiente modo.

Claves Públicas y Privadas

- Para generar las claves públicas y privadas se procederá del siguiente modo.
- La clave privada s será un número aleatorio entre 1 y $n-1$.

Claves Públicas y Privadas

- Para generar las claves públicas y privadas se procederá del siguiente modo.
- La clave privada s será un número aleatorio entre 1 y $n-1$.
- La clave pública P será un punto de la curva, concretamente el punto sG calculado mediante la multiplicación escalar.

El Protocolo Diffie-Hellman

- Una de las principales aplicaciones de la criptografía de curvas elípticas es la posibilidad de poner de acuerdo a dos usuarios para poder comunicarse mediante una clave que sólo ellos conozcan y que posiblemente nunca se han comunicado.

El Protocolo Diffie-Hellman

- Una de las principales aplicaciones de la criptografía de curvas elípticas es la posibilidad de poner de acuerdo a dos usuarios para poder comunicarse mediante una clave que sólo ellos conozcan y que posiblemente nunca se han comunicado.
- Supongamos que son A(lice) y B(ob) los que desean comunicarse. Cada uno de ellos dispone de sus claves púbicas P_A y P_B y sus claves privadas s_A y s_B.

El Protocolo Diffie-Hellman

- Una de las principales aplicaciones de la criptografía de curvas elípticas es la posibilidad de poner de acuerdo a dos usuarios para poder comunicarse mediante una clave que sólo ellos conozcan y que posiblemente nunca se han comunicado.
- Supongamos que son A(lice) y B(ob) los que desean comunicarse. Cada uno de ellos dispone de sus claves púbicas P_A y P_B y sus claves privadas s_A y s_B.
- Alice calculará el punto $Q = s_A P_B$, lo cual puede hacer porque tiene su propia clave privada y la clave púbica de B es conocida.

El Protocolo Diffie-Hellman

- Una de las principales aplicaciones de la criptografía de curvas elípticas es la posibilidad de poner de acuerdo a dos usuarios para poder comunicarse mediante una clave que sólo ellos conozcan y que posiblemente nunca se han comunicado.
- Supongamos que son A(lice) y B(ob) los que desean comunicarse. Cada uno de ellos dispone de sus claves púbicas P_A y P_B y sus claves privadas s_A y s_B.
- Alice calculará el punto $Q = s_A P_B$, lo cual puede hacer porque tiene su propia clave privada y la clave púbica de B es conocida.
- Bob calculará el punto $Q' = s_B P_A$, lo cual puede hacer por las mismas razones que Alice.

El Protocolo Diffie-Hellman

- Una de las principales aplicaciones de la criptografía de curvas elípticas es la posibilidad de poner de acuerdo a dos usuarios para poder comunicarse mediante una clave que sólo ellos conozcan y que posiblemente nunca se han comunicado.
- Supongamos que son A(lice) y B(ob) los que desean comunicarse. Cada uno de ellos dispone de sus claves púbicas P_A y P_B y sus claves privadas s_A y s_B.
- Alice calculará el punto $Q = s_A P_B$, lo cual puede hacer porque tiene su propia clave privada y la clave púbica de B es conocida.
- Bob calculará el punto $Q' = s_B P_A$, lo cual puede hacer por las mismas razones que Alice.
- Estos dos puntos son iguales porque

$$Q = s_A P_B = s_A s_B G = s_B s_A G = s_B P_A = Q'.$$

El Protocolo Diffie-Hellman

- Una de las principales aplicaciones de la criptografía de curvas elípticas es la posibilidad de poner de acuerdo a dos usuarios para poder comunicarse mediante una clave que sólo ellos conozcan y que posiblemente nunca se han comunicado.
- Supongamos que son A(lice) y B(ob) los que desean comunicarse. Cada uno de ellos dispone de sus claves púbicas P_A y P_B y sus claves privadas s_A y s_B.
- Alice calculará el punto $Q = s_A P_B$, lo cual puede hacer porque tiene su propia clave privada y la clave púbica de B es conocida.
- Bob calculará el punto $Q' = s_B P_A$, lo cual puede hacer por las mismas razones que Alice.
- Estos dos puntos son iguales porque

$$Q = s_A P_B = s_A s_B G = s_B s_A G = s_B P_A = Q'.$$

- La coordenada x del punto $Q = Q'$ (o un número calculado a partir de él) será la clave AES para continuar la comunicación.

Análisis del Protocolo

- Como podemos ver, una única multiplicación escalar permite a Alice y Bob calcular el punto Q y por lo tanto un número con el que generar la clave AES.

Análisis del Protocolo

- Como podemos ver, una única multiplicación escalar permite a Alice y Bob calcular el punto Q y por lo tanto un número con el que generar la clave AES.
- Ningún otro usuario puede conocer este punto Q puesto que no conocen ni la clave privada de Alice ni la de Bob.

Análisis del Protocolo

- Como podemos ver, una única multiplicación escalar permite a Alice y Bob calcular el punto Q y por lo tanto un número con el que generar la clave AES.
- Ningún otro usuario puede conocer este punto Q puesto que no conocen ni la clave privada de Alice ni la de Bob.
- Incluso en dispositivos con fuertes limitaciones de recursos, este sistema permite comunicaciones seguras con cálculos asequibles y claves relativamente pequeñas (en comparación por ejemplo con las claves RSA).

Introducción

- Ya vimos anteriormente un método de firma digital denominado DSA *(Digital Signature Standard)*.

Introducción

- Ya vimos anteriormente un método de firma digital denominado DSA *(Digital Signature Standard)*.
- El protocolo ECDSA *(Elliptic Curve Digital Signature Standard)* es muy similar y está definido dentro del mismo documento FIPS PUB 186-4 que utilizamos para DSA.

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:
  - Un primo p del tamaño adecuado (entre 160 y 256 bits).

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:
  - Un primo p del tamaño adecuado (entre 160 y 256 bits).
  - Los parámetros a y b de la curva en forma de Weierstrass sobre el cuerpo $\mathbb{Z}_p$.

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:
  - Un primo p del tamaño adecuado (entre 160 y 256 bits).
  - Los parámetros a y b de la curva en forma de Weierstrass sobre el cuerpo $\mathbb{Z}_p$.
  - El número de puntos de la curva n (que debe ser un número primo puesto que supondremos cofactor 1).

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:
  - Un primo p del tamaño adecuado (entre 160 y 256 bits).
  - Los parámetros a y b de la curva en forma de Weierstrass sobre el cuerpo $\mathbb{Z}_p$.
  - El número de puntos de la curva n (que debe ser un número primo puesto que supondremos cofactor 1).
  - Un punto G de la curva distinto de O.

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:
  - Un primo p del tamaño adecuado (entre 160 y 256 bits).
  - Los parámetros a y b de la curva en forma de Weierstrass sobre el cuerpo $\mathbb{Z}_p$.
  - El número de puntos de la curva n (que debe ser un número primo puesto que supondremos cofactor 1).
  - Un punto G de la curva distinto de O.
  - Un número d en el intervalo $[1, n-1]$ que será la clave privada.

Parámetros

- Para firmar mensajes con ECDSA necesitamos los siguientes parámetros:
  - Un primo p del tamaño adecuado (entre 160 y 256 bits).
  - Los parámetros a y b de la curva en forma de Weierstrass sobre el cuerpo $\mathbb{Z}_p$.
  - El número de puntos de la curva n (que debe ser un número primo puesto que supondremos cofactor 1).
  - Un punto G de la curva distinto de O.
  - Un número d en el intervalo $[1, n-1]$ que será la clave privada.
  - El punto $P = d \cdot G$ será la clave pública.

Proceso de Firma

- Calcularemos un número k de forma aleatoria en el rango $[1, n-1]$.

Proceso de Firma

- Calcularemos un número k de forma aleatoria en el rango $[1, n-1]$.
- Tomaremos (x_1, y_1) el punto kG (que es un punto afín porque $k \neq 0 \ (mod \ n)$)

Proceso de Firma

- Calcularemos un número k de forma aleatoria en el rango $[1, n-1]$.
- Tomaremos (x_1, y_1) el punto kG (que es un punto afín porque $k \neq 0 \ (mod \ n)$)
- Calcularemos $r = x_1 \ (mod \ n)$. Si r resultase 0 volveríamos a calcular un nuevo valor de k.

Proceso de Firma

- Calcularemos un número k de forma aleatoria en el rango $[1, n-1]$.
- Tomaremos (x_1, y_1) el punto kG (que es un punto afín porque $k \neq 0 \ (mod \ n)$)
- Calcularemos $r = x_1 \ (mod \ n)$. Si r resultase 0 volveríamos a calcular un nuevo valor de k.
- Finalmente tomaremos $s = k^{-1}(sha1(M) + dr) \ (mod \ n)$. Si $s = 0$ recalcularemos la firma con un nuevo valor de k. El valor k^{-1} es el inverso módulo n del número k y M será el mensaje.
- La firma es el par de valores (r, s).

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.
- $w = s^{-1} \ (mod \ n)$

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.
- $w = s^{-1} \ (mod \ n)$
- $u_1 = sha1(M)w \ (mod \ n)$

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.
- $w = s^{-1} \ (mod \ n)$
- $u_1 = sha1(M)w \ (mod \ n)$
- $u_2 = rw \ (mod \ n)$

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.
- $w = s^{-1} \ (mod \ n)$
- $u_1 = sha1(M)w \ (mod \ n)$
- $u_2 = rw \ (mod \ n)$
- Calcularemos el punto $(x_0, y_0) = u_1 G + u_2 P$

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.
- $w = s^{-1} \ (mod \ n)$
- $u_1 = sha1(M)w \ (mod \ n)$
- $u_2 = rw \ (mod \ n)$
- Calcularemos el punto $(x_0, y_0) = u_1 G + u_2 P$
- Finalmente tomaremos $v = x_0 \ (mod \ n)$.

Verificación de Firma

- Si los valores r, s están fuera del rango $[1, n-1]$ daremos la firma como incorrecta.
- $w = s^{-1} \ (mod \ n)$
- $u_1 = sha1(M)w \ (mod \ n)$
- $u_2 = rw \ (mod \ n)$
- Calcularemos el punto $(x_0, y_0) = u_1 G + u_2 P$
- Finalmente tomaremos $v = x_0 \ (mod \ n)$.
- Si $v = r$ entonces la firma es correcta, si no la firma es incorrecta.

Algunos Comentarios Finales

- De todas estas operaciones, las más costosas computacionalmente son las operaciones de punto, en el caso de la generación de firma se necesita una única multiplicación escalar, mientras que en el caso de la verificación se necesitan dos por lo que en general será más costosa la verificación que la generación.

Algunos Comentarios Finales

- De todas estas operaciones, las más costosas computacionalmente son las operaciones de punto, en el caso de la generación de firma se necesita una única multiplicación escalar, mientras que en el caso de la verificación se necesitan dos por lo que en general será más costosa la verificación que la generación.
- El cálculo $u_1 G + u_2 P$ se puede hacer mediante el conocido truco de Shamir (que no hemos explicado aquí), pero podemos hacerlo también calculando primero $u_1 G$ luego $u_2 P$ y finalmente sumando los resultados.

Algunos Comentarios Finales

- De todas estas operaciones, las más costosas computacionalmente son las operaciones de punto, en el caso de la generación de firma se necesita una única multiplicación escalar, mientras que en el caso de la verificación se necesitan dos por lo que en general será más costosa la verificación que la generación.
- El cálculo $u_1 G + u_2 P$ se puede hacer mediante el conocido truco de Shamir (que no hemos explicado aquí), pero podemos hacerlo también calculando primero $u_1 G$ luego $u_2 P$ y finalmente sumando los resultados.
- Los cálculos de inversos módulo n se realizan mediante el algoritmo de Euclides extendido.