Capítulo 7

Criptosistemas de curvas elípticas (CCE)

7.1. Introducción

En este capítulo presentaremos una revisión del esquema de cifrado de curvas elípticas (ECES), dos esquemas de firma digital mediante curvas elípticas (ECSS y ECDSA), y un protocolo de acuerdo de claves de curvas elípticas (ECKEP). [1]

7.2. Elección de una curva apropiada

Una curva elíptica E definida sobre un campo finito $GF(q)$ es *apropiada* para propósitos criptográficos [12] si se satisfacen las siguientes condiciones:

1. El orden $u = |E/GF(q)|$ de la curva debe ser divisible por un número primo grande r.

2. El divisor primo grande r no debe dividir $q^v - 1$ para $v = 1, 2, 3, \ldots, 10$. Esta condición asegura que el algoritmo de reducción MOV [65] no es factible.

Existen tres enfoques para seleccionar una curva elíptica sobre $GF(q)$ con tales características [12].

[1] De acuerdo al estándarelaborado por la IEEE [52].

7.2.1. Método 1 - Selección de una curva aleatoria

1. Seleccionar aleatoriamente los parámetros $a, b \in GF(q)$ que definen la ecuación de la curva elíptica. En el caso en el cual q es un primo, verificar que $4a^3 + 27b^2 \neq 0$. La ecuación de la curva es $E : y^2 = x^3 + ax + b$. En el caso en el cual $q = 2^m$, verificar que $b \neq 0$. La ecuación es $E : y^2 + xy = x^3 + ax^2 + b$.

2. Calcular el orden del grupo de puntos racionales de la curva $u = |E(GF(q))|$.

3. Factorizar u. Si este orden no es divisible por un primo grande r, entonces ir al paso 1.

4. Verificar que el divisor primo grande r de u no divide a $q^v - 1$, para $v = 1, 2, 3, \ldots, 10$. Si la prueba fracasa, entonces ir al paso 1.

5. Dar como salida la curva E.

El orden $|E(GF(q))|$ puede ser calculado mediante el algoritmo de Schoof [87]. Si bien el algoritmo es bastante ineficiente, recientemente se han descubierto significativas mejoras [57]. En el momento actual es factible calcular ordenes de curvas elípticas sobre $GF(p)$ donde p es cercano a 10^{499}, y ordenes de curvas elípticas sobre $GF(2^m)$ donde m es tan grande como 601 [57]. Pero lo más relevante es que el orden de curvas elípticas sobre el campo $GF(2^{155})$ pueden ser calculados en menos de 4 minutos en una estación de trabajo [57].

7.2.2. Método 2 - Seleccionando primero el orden de la curva

Para curvas elípticas sobre $GF(p)$ donde p es primo, se hace lo siguiente:

1. Seleccionar un orden u tal que

   a) $p + 1 - 2\sqrt{p} < u < p + 1 + 2\sqrt{p}$

   b) u es divisible por un primo grande r.

   c) r no divide $p^v - 1$ para $v = 1, 2, 3, \ldots, 10$.

2. Usar el algoritmo de Atkin-Morain [69] para encontrar parámetros $a, b \in GF(p)$ tal que la curva elíptica E definida por ellos tenga orden $|E(GF(p))| = u$.

3. Dar como salida la curva E.

Para curvas elípticas sobre $GF(2^m)$ se hace lo siguiente:

1. Seleccionar un entero pequeño c_{max}. Usualmente, $c_{max} = 2$ o 4.
2. Usar el algoritmo de Lay-Zimmer [55] para encontrar $a, b \in GF(2^m)$ tal que la curva elíptica E definida por ellos tenga orden $u = cr$, donde r es un primo y $c \leq c_{max}$.
3. Verificar que r no divide a $(2^m)^v - 1$ para $v = 1, 2, 3, \ldots, 10$. Si esto falla, entonces ir al paso 1.
4. Dar como salida la curva E.

El algoritmo de Lay-Zimmer [55] requiere de algunos cálculos previos para cada campo particular $GF(2^m)$. Una vez hecho esto, el algoritmo solo requiere algunos minutos en una estación de trabajo, aun cuando m sea tan grande como 300.

7.2.3. Método 3 - Usando el teorema de Weil

Esta técnica puede utilizarse para elegir curvas sobre $GF(2^m)$, cuando m es divisible por un número pequeño l.

1. Seleccionar una curva aleatoria $E : y^2 + xy = x^3 + ax^2 + b$, $b \neq 0$, donde $a, b \in GF(2^l)$. Observemos que dado que $GF(2^l)$ está contenido en $GF(2^m)$, es también cierto que $a, b \in GF(2^m)$, y por tanto E es también una curva sobre $GF(2^m)$.
2. Calcular $w = |E(GF(2^l))|$. Esto puede hacerse rápidamente dado que l es pequeño.
3. Sea $t = q^l + 1 - w$, y sea $c = m/l$. Entonces calcular el término c-ésimo de la secuencia $\{V_n\}$ definida por

$$\begin{aligned} V_0 &= 2, \\ V_1 &= t, \\ V_n &= tV_{n-1} - 2^l V_{n-2} \text{ para } n \geq 2. \end{aligned}$$

   Entonces hacer

$$u = |E(GF(2^m))| = 2^m + 1 - V_c.$$

4. Factorizar u; si el orden no es divisible por un primo grande r, entonces ir al paso 1.

5. Verificar que el divisor primo grande r de u no divide a $q^v - 1$, para $v = 1, 2, 3, \ldots, 10$. Si esta prueba falla, entonces ir al paso 1.

6. Dar como salida la curva E.

7.3. Generación de claves

En primer lugar se debe elegir un campo finito subyacente $GF(q)$, el cual es común a todos los usuarios. Existen dos opciones para generar el resto de los parámetros:

1. *Claves públicas cortas.* Los parámetros de la curva elíptica, es decir los elementos a y b (los cuales definen la curva elíptica), y un punto $P \in E$ de orden primo n, son elegidos por el administrador del sistema. Los parámetros $a, b, P = (x_P, y_P), n$ son públicos y comúnes a todos los usuarios.

2. *Claves públicas largas.* Cada usuario elige sus propios parámetros de curva elíptica, es decir los elementos a y b (los cuales definen a la curva elíptica) y el punto P de orden primo n. Para hacer la clave pública más corta, el parámetro a se restringe a $a = 0$ si el campo subyacente es $GF(2^m)$ y $a = 1$ si el campo subyacente es $GF(p)$. Los parámetros $b, P = (x_P, y_P)$, deben ser parte de la clave pública del usuario. Si se utiliza la firma digital ECDSA, entonces la clave pública debe incluir el orden n de P.

Para generar sus claves, cada principal en el sistema debe realizar el siguiente procedimiento:

1. Elegir un número entero aleatorio d tal que $1 \leq d < n$.

2. Calcular el punto $Q := dP$.

3. Sea $Q = (x_Q, y_Q)$.

4. *a*) Para la opción de claves públicas cortas, la clave pública del principal consiste del punto Q representado por los valores x_Q y y_Q.

   b) Para la opción de claves públicas largas, la clave pública del principal consiste del punto P, representado por x_P y y_P, y el punto Q representado por x_Q y y_Q. (Obsérvese que el parámetro b de la curva elíptica puede ser fácilmente calculado a partir de x_P y y_P como sigue: Si $q = p$ entonces $b = y_P^2 - x_P^3 - x_P$; si $q = 2^m$ entonces $b = y_P^2 + x_P y_P + x_P^3$.)

5. La clave privada del principal es el entero d.

Obsérvese que para que un tercero pueda determinar la clave privada a partir de la clave pública, debe ser capaz de resolver el problema del logaritmo discreto sobre curvas elípticas.

7.4. Representación del campo

En esta sección describiremos las posibles representaciones que pueden usarse para los elementos del campo subyacente $GF(q)$.

7.4.1. El campo finito $GF(p)$.

El campo finito $GF(p)$ está formado por el conjunto de enteros

$$\{0, 1, 2, \ldots, p-1\}.$$

Cada uno de tales enteros puede ser representado por una cadena binaria de longitud $t = \lceil \log_2 p \rceil$ la cual es la representación binaria del entero.

7.4.2. El campo finito $GF(2^m)$.

El campo finito $GF(2^m)$ se compone del conjunto de cadenas binarias de longitud exácta m. Un elemento $a \in GF(2^m)$ es representado por la cadena binaria $a = (a_0 a_1 a_2 \cdots a_{m-1})$. El elemento cero es representado por la cadena de los m bits en cero. La suma de dos elementos del campo se realiza sencillamente por el XOR de las representaciones binarias. La multiplicación en general depende de la representación, por ejemplo, en bases normales óptimas.

7.5. Representación de puntos sobre la curva.

Un punto P de la curva elíptica (distinto al punto al infinito $\mathcal{O}$) es representado por dos elementos del campo, la coordenada x de P y la coordenada y de P: $P = (x_P, y_P)$. El punto puede ser representado en forma más compacta almacenando únicamente la coordenada x_P y un cierto bit $\widetilde{y_P}$ derivado de las coordenadas x_P y y_P. A continuación describiremos esta técnica de compresión de puntos.

7.5.1. Compresión de puntos (Curvas elípticas sobre $GF(p)$)

Sea $P = (x_P, y_P)$ un punto sobre la curva elíptica $E : y^2 = x^3 + ax + b$ definida sobre un campo primo $GF(p)$. Entonces $\widetilde{y_P}$ se define como el bit menos significativo de y_P.

Supóngase que se nos proporcionan la coordenada x_P de P y el bit $\widetilde{y_P}$. Entonces podemos recuperar y_P de la siguiente forma:

1. Calcular el elemento del campo $\alpha = x_P^3 + ax_P + b \bmod p$.
2. Calcular una raíz cuadrada β de α módulo p.
3. Si el bit menos significativo de β es igual a $\widetilde{y_P}$ entonces $y_P \leftarrow \beta$ y en otro caso $y_P \leftarrow p - \beta$.

7.5.2. Compresión de puntos (Curvas elípticas sobre $GF(2^m)$)

La técnica descrita en esta sección solo aplica al caso en el cual los elementos del campo $GF(2^m)$ están respresentados con respecto a una base normal óptima.

Sea $P = (x_P, y_P)$ un punto sobre la curva elíptica $E : y^2 + xy = x^3 + ax^2 + b$ definida sobre un campo $GF(2^m)$. Entonces $\widetilde{y_P}$ se define como 0 si $x_P = 0$; si $x_P \neq 0$ entonces $\widetilde{y_P}$ se define como el bit menos significativo del elemento del campo $y_P \cdot x_P^{-1}$.

Supóngase que nos son dados la coordenada x_P de P y el bit $\widetilde{y_P}$. Entonces y_P puede ser recuperado como se indica a continuación.

1. Si $x_P = 0$ entonces y_P es obtenida por un corrimiento cíclico del vector representación de b una posición a la izquierda. Esto es, si $b = b_{m-1}b_{m-2}\cdots b_1b_0$ entonces $y_P \leftarrow b_{m-2}\cdots b_1b_0b_{m-1}$.
2. Si $x \neq 0$ entonces hacer lo siguiente:
   a) Calcular el elemento del campo $\alpha = x_P + a + bx_P^{-2}$ en $GF(2^m)$.
   b) Sea $\alpha = \alpha_{m-1}\alpha_{m-2}\cdots\alpha_1\alpha_0$ la representación vectorial de α.
   c) Construir un elemento $z = z_{m-1}z_{m-2}\ldots z_1z_0$ del campo haciendo

$$\begin{aligned} z_0 &= \widetilde{y_P}, \\ z_1 &= \alpha_0 \oplus z_0, \\ z_2 &= \alpha_1 \oplus z_1, \\ &\vdots \end{aligned}$$

$$
\begin{aligned}
z_{m-2} &= \alpha_{m-3} \oplus z_{m-3}, \\
z_{m-1} &= \alpha_{m-2} \oplus z_{m-2}
\end{aligned}
$$

d) Finalmente, calcular $y_P \leftarrow x_P \cdot z$.

7.6. Esquema de cifrado (ECES)

7.6.1. Cifrado

En esta sección describiremos el proceso de cifrado ECES (*Elliptic Curve Encryption Scheme*) [45][52].

El proceso de cifrado consiste de cuatro pasos: el formateo del bloque a cifrar, los cálculos sobre la curva elíptica, la inclusión de datos y la conversión de punto a cadena de bytes.

La entrada al proceso de cifrado es:

- Una cadena de bytes M, que son los datos a cifrar. La longitud de M no debe ser mayor a $l - 2$ bytes, donde l es la longitud del orden del campo $GF(q)$ en bytes, es decir,

$$l = \left\lceil \frac{t}{8} \right\rceil$$

  donde $t = \lceil \log_2 q \rceil$.

- Dos elementos del campo a y b los cuales describen la ecuación de la curva elíptica. Un elemento del campo es representado por una cadena binaria de longitud t.

- Dos elementos del campo x_P y y_P que juntos describen al punto $P = (x_P, y_P)$ de orden n.

- Dos elementos x_Q y y_Q que juntos describen al punto $Q = (x_Q, y_Q)$ que forma la clave pública.

La salida del proceso de cifrado deberá consistir en una cadena de bytes MC, de los datos cifrados. La longitud de MC es $2l + 1$ bytes si se utiliza compresión de puntos, y $3l + 1$ bytes si no se utiliza compresión de puntos.

Formateo del bloque

1. Rellenar a la izquierda los datos M con una cadena de relleno R que consiste en $l - 2 - |M|$ bytes cada uno con el valor ff, seguidos de un byte 00, para formar la cadena M' de longitud $l - 1$:

$$M' = R \parallel 00 \parallel M.$$

   Obsérvese que la longitud de R puede ser nula si la longitud de M es $l - 2$ bytes.

Cálculos sobre la curva elíptica

1. Seleccionar un entero aleatorio k en el intervalo $[1, n - 1]$.

2. Calcular el punto de la curva elíptica $(x_1, y_1) := kP$, donde P es el punto (x_P, y_P).

3. Calcular el punto de la curva elíptica $(x_2, y_2) := kQ$, donde Q es el punto (x_Q, y_Q).

4. Si $x_2 = 0$ ir al paso 1.

Inclusión de datos

1. Convertir M' a una cadena binaria, y entonces concatenar una cadena binaria de ceros de longitud $8 - 8l + t$ a la izquierda de esta cadena para formar un elemento m del campo.

2. Realizar una multiplicación en el campo para obtener $c := m \cdot x_2$.

Obsérvese que el bit más significativo del elemento m es 0. La razón para hacer esto es asegurar, en el caso en el cual q es igual a un primo p, que el entero representado por m es menor que el módulo p.

Conversión de punto a cadena de bytes

1. Añadir una cadena binaria de ceros de longitud $8l - t$ a la izquierda del elemento x_1 del campo, y convertir la cadena binaria resultante de $8l$ bits en una cadena de bytes X_1 de longitud l.

2. Calcular el valor $\widetilde{y_1}$.

3. Si se utiliza compresión de puntos entonces asignar a un byte, PC, el valor 01 si $\widetilde{y_1}$ es 0, o el valor 03 si $\widetilde{y_1}$ es 1.

   Si no es utilizada la compresión de punto entonces hacer lo siguiente:

   a) Asignar al byte PC el valor 00.

   b) Añadir una cadena binaria de ceros de longitud $8l-t$ a la izquierda del elemento y_1 del campo, y covertir la cadena binaria de $8l$ bits resultante en una cadena de bytes Y_1 de longitud l.

4. Añadir una cadena binaria de ceros de longitud $8l - t$ a la izquierda del elemento c del campo, y convertir la cadena binaria de $8l$ bits resultante a una cadena de bytes C de longitud l.

5. Finalmente, obtener el texto cifrado MC mediante la concatenación de las cuatro cadenas de bytes X_1, PC, Y_1, y C (si se utiliza compresión de puntos entonces Y_1 es la cadena nula):

$$MC = X_1 \parallel PC \parallel Y_1 \parallel C.$$

La longitud de MC es de $2l + 1$ bytes si se utiliza compresión de puntos, y de $3l + 1$ bytes si no es así.

7.6.2. Descifrado

En esta sección detallamos el proceso de descifrado de ECES.

El proceso de descifrado consiste de cuatro etapas: Conversión de cadena de bytes a puntos, cálculos de la curva elíptica, extracción de datos, y parsing del bloque descifrado.

La entrada al proceso de descifrado es:

- Una cadena de bytes MC, de longitud $2l+1$ o $3l+1$, que es el mensaje cifrado.
- Dos elementos del campo a y b los cuales describen a la ecuación de la curva elíptica.
- Un entero d, la clave privada.

La salida del proceso de descifrado debe ser una cadena de bytes M de longitud a lo más $l - 2$, que son los datos descifrados.

Conversión de cadena de bytes a punto

1. Hacer parse del mensaje cifrado MC para obtener las cadenas de bytes X_1, PC, Y_1 y C.

$$MC = X_1 \parallel PC \parallel Y_1 \parallel C.$$

   X_1 y C son cada uno de l bytes de longitud. PC es un solo byte. Si el bit menos significativo de PC es 1 entonces Y_1 es el byte nulo; en otro caso Y_1 es de l bytes de longitud.

2. Convertir X_1 a una cadena binaria, y entonces descartar los $8l - t$ bits más significativos para obtener un elemento x_1 del campo.

3. Si el bit menos significativo de PC es 1 entonces hacer $\widetilde{y_1}$ igual al segundo bit menos significativo de PC. Si el bit menos significativo de PC es 0 entonces convertir Y_1 a una cadena binaria, y entonces descartar los $8l-t$ bits más significativos para obtener la cadena binaria $\widetilde{y_1}$.

4. Convertir C a una cadena binaria, y entonces descartar los $8l - t$ bits más significativos para obtener un elemento c del campo.

Cálculos sobre la curva elíptica

1. Usar x_1 y $\widetilde{y_1}$ para obtener el punto (x_1, y_1) sobre la curva elíptica.

2. Calcular el punto $(x_2, y_2) := d(x_1, y_1)$ de la curva elíptica.

Extracción de datos.

1. Calcular $m := c \cdot x_2^{-1}$.

2. Descartar los $8 - 8l + t$ bits más significativos de m, y convertir la cadena binaria resultante en una cadena de bytes M' de longitud $l-1$.

Parsing del bloque descifrado

1. Hacer parsing de M' para obtener los datos M:

$$M' = R \parallel 00 \parallel M.$$

7.6.3. Ejemplo de cifrado con ECES

Este ejemplo ilustra el esquema de cifrado de curva elíptica ECES. En este ejemplo se ha tomado intencionalmente un campo subyacente pequeño para ilustrar mejor las operaciones realizadas. En la subsección siguiente tomaremos un caso real.

Por brevedad, en los ejemplos se han omitido las partes de inclusión y extracción de datos.

Configuración del Sistema

El campo subyacente será $GF(2^4)$ y la curva elíptica la descrita en el ejemplo 6.2. Se selecciona el punto $P = (x_P, y_P) = (\alpha^5, \alpha^{11})$. Dado que $5P = \mathcal{O}$, el punto P es de orden $n = 5$.

Generación de la clave

La entidad A realiza las siguientes operaciones:

1. A selecciona un entero aleatorio $d = 3$ en el intervalo $[1, 4]$.
2. A calcula el punto $Q = dP = 3P = (\alpha, 0) = ((1100), (0000))$.
3. A hace público al punto Q.
4. La clave privada de A es el entero $d = 3$.

Proceso de cifrado

Supongamos que la entidad B desea enviar los datos $M = 0111$ a la entidad A. La entidad B entonces realiza los siguientes pasos:

1. B obtiene la clave pública de A: $Q = (x_Q, y_Q) = (\alpha, 0)$.
2. B representa los datos M como elemento del campo $m = (0111) = \alpha^7$.
3. B selecciona la azar un entero aleatorio $k = 2$ en el intervalo $[1, 4]$.
4. B calcula el punto $2P = (x_1, y_1) = (\alpha, \alpha) = ((1100), (1100))$.
5. B calcula el punto $2Q = (x_2, y_2) = (\alpha^5, \alpha^{11}) = ((1010), (1110))$.
6. B calcula $c = m \cdot x_2 = \alpha^7 \cdot \alpha^5 = \alpha^{12} = (0001)$.
7. Finalmente, B transmite a A los datos cifrados:

$$(x_1, y_1, c) = (\alpha, \alpha, \alpha^{12}) = ((1100), (1100), (0001)).$$

Proceso de descifrado

La entidad A descifra el texto cifrado $C = (\alpha, \alpha, \alpha^{12})$ recibido de B. Para esto realiza los siguientes pasos:

1. A calcula el punto:

$$d(x_1, y_1) = 3(\alpha, \alpha) = (\alpha^5, \alpha^{11}) = ((1010), (1110)) = (x_2, y_2).$$

2. A recupera los datos: $m = c \cdot x_2^{-1} = \alpha^{12} \cdot \alpha^{-5} = \alpha^7 = (0111)$.

7.6.4. Cifrado ECES sobre la curva K-163

En el ejemplo anterior hemos utilizado una curva elíptica definida sobre $GF(2^4)$. El orden de esta curva es 16 (veáse el ejemplo 6.2), demasiado pequeño como para tener alguna utilidad práctica. El NIST [73] recomienda para fines criptográficos el uso de curvas definidas sobre campos de al menos grado de extensión 163, tal como la curva K-163 introducida previamente en el ejemplo 6.3.

A continuación ilustraremos el esquema de cifrado ECES utilizando la curva K-163. Todos los cálculos de esta sección fueron realizados mediante el código presentado en el apéndice B.

Configuración del sistema

El campo subyacente es $GF(2^{163})$ con la representación en base polinomial dada en el ejemplo 5.4 y la curva elíptica es la especificada en el ejemplo 6.3, conocida como K-163.

El punto generador es $G = (G_x, G_y)$ donde

$$\begin{aligned} G_x &= \text{0x8eee49c5e5d6e4ed397d70aaca11cbb7350c31ef2} \\ G_y &= \text{0x9d3aadcc835d635008e2f12385ff83d50bf070982.} \end{aligned}$$

El orden de G es

$$n = 5846006549323611672814741753598448348329118574063.$$

Generación de la clave

El principal A realiza las siguientes operaciones:

1. A selecciona un entero aleatorio

$$d = 372897787406025110559817060653469743567647493062 6$$

en el intervalo $[1, n]$.

2. A calcula el punto $Q = dG$. Si $Q = (Q_x, Q_y)$ entonces

$$\begin{aligned} Q_x &= 0\text{x}4\text{e}35\text{bba}00\text{f}9\text{faa}55\text{d}0193948\text{ba}7997\text{aaae}449\text{f}823 \\ Q_y &= 0\text{xa}0243\text{a}4\text{de}5501\text{a}0\text{bb}99497\text{c}878\text{b}9\text{c}3\text{db}0\text{f}5\text{b}59505. \end{aligned}$$

3. La clave pública de A es el punto Q.

4. La clave privada de A es el entero d.

Proceso de cifrado

Supongamos que el principal B desea enviar a A los datos

$$M = 0\text{x}123456789\text{abcdef}.$$

Por simplicidad, ya estamos tomando el mensaje en su representación como elemento del campo subyacente. El principal B realiza entonces los siguientes pasos:

1. B obtiene la clave pública de A, $Q = (Q_x, Q_y)$ donde

$$\begin{aligned} Q_x &= 0\text{x}4\text{e}35\text{bba}00\text{f}9\text{faa}55\text{d}0193948\text{ba}7997\text{aaae}449\text{f}823 \\ Q_y &= 0\text{xa}0243\text{a}4\text{de}5501\text{a}0\text{bb}99497\text{c}878\text{b}9\text{c}3\text{db}0\text{f}5\text{b}59505. \end{aligned}$$

2. B selecciona un entero aleatorio

$$k = 282359440703474456385945599617865132419521385384 7$$

en el intervalo $[1, n]$.

3. B calcula el punto $kG = (x_1, y_1)$ donde

$$\begin{aligned} x_1 &= 0\text{x}83237520\text{e}8480\text{a}7376\text{a}2\text{da}0\text{a}3\text{aaa}4\text{c}582\text{f}1\text{ccc}3\text{a}5 \\ y_1 &= 0\text{x}78720\text{dd}5179\text{ff}84\text{efc}1\text{f}5\text{a}060\text{e}20\text{f}711\text{c}3\text{a}7\text{f}2\text{f}36 \end{aligned}$$

4. B calcula enseguida el punto $kQ = (x_2, y_2)$ obteniendo las coordenadas

$$\begin{aligned} x_2 &= 0\text{xf}391\text{e}04\text{c}1\text{d}73\text{fba}43\text{f}86\text{c}847\text{da}13549\text{df}7\text{c}822044 \\ y_2 &= 0\text{x}914\text{f}4\text{b}01182976\text{d}7\text{afd}65010\text{d}9\text{e}4\text{d}11\text{c}43\text{a}163243 \end{aligned}$$

5. B calcula

$$c = m \cdot x_2 = 0\text{x}5\text{ec}39\text{ee}2\text{b}747\text{b}11\text{bb}8055\text{ad}881\text{dcfc}869\text{e}7\text{e}83\text{e}43.$$

6. Finalmente, B transmite a A los datos cifrados (x_1, y_1, c), esto es, el mensaje

$$\begin{aligned} &0\text{x}83237520\text{e}8480\text{a}7376\text{a}2\text{da}0\text{a}3\text{aaa}4\text{c}582\text{f}1\text{ccc}3\text{a}5 \\ &0\text{x}78720\text{dd}5179\text{ff}84\text{efc}1\text{f}5\text{a}060\text{e}20\text{f}711\text{c}3\text{a}7\text{f}2\text{f}36 \\ &0\text{x}5\text{ec}39\text{ee}2\text{b}747\text{b}11\text{bb}8055\text{ad}881\text{dcfc}869\text{e}7\text{e}83\text{e}43 \end{aligned}$$

Proceso de descifrado

La entidad A descifra el texto cifrado

$$\begin{aligned} x_1 &= 0\text{x}83237520\text{e}8480\text{a}7376\text{a}2\text{da}0\text{a}3\text{aaa}4\text{c}582\text{f}1\text{ccc}3\text{a}5 \\ y_1 &= 0\text{x}78720\text{dd}5179\text{ff}84\text{efc}1\text{f}5\text{a}060\text{e}20\text{f}711\text{c}3\text{a}7\text{f}2\text{f}36 \\ c &= 0\text{x}5\text{ec}39\text{ee}2\text{b}747\text{b}11\text{bb}8055\text{ad}881\text{dcfc}869\text{e}7\text{e}83\text{e}43 \end{aligned}$$

recibido de B. Para esto realiza los siguientes pasos:

1. A calcula el punto $d(x_1, y_1)$ obteniendo

$$\begin{aligned} x_2 &= 0\text{xf}391\text{e}04\text{c}1\text{d}73\text{fba}43\text{f}86\text{c}847\text{da}13549\text{df}7\text{c}822044 \\ y_2 &= 0\text{x}914\text{f}4\text{b}01182976\text{d}7\text{afd}65010\text{d}9\text{e}4\text{d}11\text{c}43\text{a}163243 \end{aligned}$$

2. A recupera los datos:

$$m = c \cdot x_2^{-1} = 0\text{x}123456789\text{abcdef}.$$

7.7. Firma digital (ECSS y ECDSA)

A continuación describimos dos esquemas de firma digital utilizando curvas elípticas. En ambos esquemas el mensaje a firmarse es primero reducido mediante una función de *hash* a un resúmen del mensaje de longitud fija, y entonces ese resúmen es firmado. La verificación de la firma en ambos esquemas requiere tanto la firma como el mensaje original. El esquema ECDSA es el análogo en curvas elípticas al algoritmo de firma digital del NIST [73].

7.7.1. Esquema de firma digital de curvas elípticas ECSS

Esta sección describe el proceso de generación de firma digital ECSS (*Elliptic Curve Signature Scheme*) [52].

El proceso de generación de firma consiste de cuatro pasos: resúmen del mensaje, cálculos sobre curvas elípticas, cálculos en el campo finito, y conversión de entero a cadena de bytes.

La entrada al proceso de firma es:

- Una cadena de bytes M, el mensaje, de longitud arbitraria.
- Dos elementos a y b en el campo, los cuales describen a la ecuación de la curva elíptica. Un elemento del campo está representado por una cadena binaria de l bits de longitud, o en forma equivalente por una cadena de bytes de longitud $l = \lceil \frac{t}{8} \rceil$.
- Dos elementos x_P y y_P del campo, los cuales describen al punto $P = (x_P, y_P)$.
- Un entero n, el orden de P. Un entero en el intervalo $[0, n-1]$ es representado por una cadena binaria de longitud $h = \lceil \log_2 n \rceil$ bits, o equivalentemente por una cadena de bytes de longitud $f = \lceil \frac{h}{8} \rceil$.
- Un entero d, la clave privada.
- Una función de hash criptográfica H.

La salida del proceso de generación de firma debe ser una cadena de bytes SI, la firma de M, de longitud $l + f$.

Resúmen del mensaje

1. Convertir M a una cadena binaria M'.
2. Calcular el valor de hash $e := H(M')$. Si H es, por ejemplo, el algoritmo SHA-1, entonces el valor de hash e es una cadena binaria de longitud 160.

Cálculos sobre la curva elíptica

1. Seleccionar un entero aleatorio k en el intervalo $[1, n-1]$.
2. Calcular el punto $(x_1, y_1) := kP$ sobre la curva elíptica, donde P es el punto (x_P, y_P).

Cálculos modulares

1. Convertir la cadena binaria e al entero $\bar{e}$.
2. Convertir la cadena binaria x_1 de t bits de longitud a un entero $\bar{x_1}$.
3. Calcular $r := \bar{x_1} + \bar{e} \bmod q$.
4. Calcular $s := k - dr \bmod n$.

Conversión de entero a cadena de bytes

1. Convertir r a una cadena binaria de longitud t. Añadir una cadena binaria de ceros de longitud $8l - t$ a la izquierda, y convertir la cadena binaria de $8l$ bits a una cadena de bytes R de longitud l.
2. Convertir s a una cadena binaria de longitud h. Añadir una cadena binaria de ceros de longitud $8f - h$ a su izquierda, y covertir la cadena resultante de $8f$ bits en una cadena de bytes S de longitud f.
3. La firma de M es la cadena de bytes

$$SI := R \parallel S$$

de longitud $l + f$.

7.7.2. Verificación de firma ECSS

Esta sección describe el proceso de verificación de firma ECSS.

El proceso de verificación de firma consiste de cuatro pasos: conversión de cadena de bytes a entero, resúmen del mensaje, cálculos sobre la curva elíptica, y verificación de firma.

La entrada del proceso de verificación de la firma es:

- Una cadena de bytes M, el mensaje.
- Una cadena de bytes SI, la firma de M.
- Dos elementos a y b en el campo, las cuales describen la ecuación de la curva elíptica.
- Dos elementos x_P y y_P del campo, los cuales describen al punto sobre la curva elíptica $P = (x_P, y_P)$.
- Dos elementos x_Q y y_Q del campo, los cuales describen al punto sobre la curva elíptica $Q = (x_Q, y_Q)$ que es la clave pública.

Conversión de cadena de bytes a entero

1. Hacer parsing de la cadena de bytes SI para obtener una cadena de bytes R de longitud l y una cadena de bytes S de longitud f

$$SI := R \parallel S.$$

2. Convertir R a una cadena binaria de longitud $8l$. Descartar los $8l - t$ bits más significativos y convertir el resultado a un entero r.

3. Convertir S a una cadena binaria de longitud $8f$. Descartar sus $8f - h$ bits más significativos y convertir el resultado a un entero s.

Resúmen del mensaje

1. Convertir M a una cadena binaria M'.

2. Calcular el valor de hash $e := H(M')$. Si H es, por ejemplo, el algoritmo SHA-1, entonces el valor de hash e es una cadena binaria de longitud 160.

Cálculos sobre la curva elíptica

1. Calcular el punto sobre la curva elíptica $(x_1, y_1) := sP + rQ$ donde $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$.

Verificación

1. Convertir la cadena binaria e a un entero $\bar{e}$.

2. Convertir la cadena binaria x_1 a un entero $\bar{x_1}$.

3. Calcular $r' := \bar{x_1} + \bar{e} \bmod q$.

4. Aceptar la firma como válida si y solamente si los enteros r y r' son iguales.

7.7.3. Ejemplo de firma digital ECSS

En este ejemplo supongamos que los parámetros del sistema y las claves de la entidad A son los mismos del ejemplo 7.6.3.

Generación de la firma ECSS

La entidad A desea firmar el mensaje $M = 10110100101110$. Supóngase que el valor de hash de M es $H(M) = 15$. Entonces A realiza los siguientes pasos:

1. Selecciona un entero aleatorio $k = 4$ en el intervalo $[1, 4]$.
2. Calcula $kP = 4P = (\alpha^5, \alpha^3) = (x_1, y_1)$.
3. Representa $x_1 = \alpha^5 = (1010)$, como el entero $8 + 2 = 10$.
4. Calcula

$$\begin{aligned} r &= (x_1 + e) \bmod q \\ &= (10 + 15) \bmod 16 \\ &= 9. \end{aligned}$$

5. Usa la clave privada $d = 3$ para calcular:

$$\begin{aligned} s &= (k - dr) \bmod n \\ &= (4 - 3 \cdot 9) \bmod 5 \\ &= 2 \end{aligned}$$

6. La firma del mensaje M es $(r, s) = (9, 2)$.

Verificación de la firma ECSS

La entidad B verifica la firma $(r, s) = (9, 2)$ sobre M como se indica a continuación.

1. B obtiene la clave pública de A: $Q = (\alpha, 0)$.
2. B calcula el punto

$$sP + rQ = (\alpha, \alpha) + (\alpha, \alpha) = (\alpha^5, \alpha^3) = (x_1, y_1).$$

3. B calcula $H(M) = e = 15$.
4. B representa $x_1 = \alpha^5 = (1010)$ como el entero $8 + 2 = 10$.

5. B calcula

$$\begin{aligned} r' &= (x_1 + e) \bmod q \\ &= (10 + 15) \bmod 16 \\ &= 9. \end{aligned}$$

6. B acepta la firma de A sobre M dado que $r' = r = 9$.

7.7.4. Algoritmo de firma digital ECDSA

A continuación detallaremos el proceso de generación de firma digital de curvas elípticas ECDSA [4].

El proceso de generación de la firma consiste de cuatro pasos: resúmen del mensaje, cálculos sobre la curva elíptica, calculos modulares, y conversión de enteros a cadenas de bytes.

La entrada del proceso de firma es:

- Una cadena de bytes M, el mensaje, de longitud arbitraria.
- Dos elementos a y b en el campo los cuales describen la ecuación de la curva elíptica. Un elemento del campo es representado por una cadena binaria de longitud t.
- Dos elementos x_P y y_P, los cuales describen al punto $P = (x_P, y_P)$.
- Un entero n, el orden de P. Un entero en el intervalo $[0, n-1]$ es representado mediante una cadena binaria de longitud $h = \lceil \log_2 n \rceil$ bits o, equivalentemente por una cadena de bytes de longitud $f = \lceil \frac{h}{8} \rceil$.
- Un entero d, la clave privada.
- Una función de hash criptográfica H.

La salida del proceso de firma debe ser una cadena de bytes SI, la firma de M, de longitud $2f$.

Resúmen del mensaje

1. Convertir M a una cadena binaria M'.
2. Calcular el valor de hash $e := H(M')$. Si H es, por ejemplo, el algoritmo SHA-1, entonces el valor de hash e es una cadena binaria de longitud 160.

Cálculos sobre la curva elíptica

1. Seleccionar un entero aleatorio k en el intervalo $[1, n-1]$.
2. Calcular el punto $(x_1, y_1) := kP$ sobre la curva elíptica, donde P es el punto (x_P, y_P).
3. Convertir la cadena de bits x_1 de longitud t a un entero $\bar{x_1}$.
4. Hacer $r := \bar{x_1} \bmod n$.

Si $r = 0$ entonces la verificación de la firma fallará. Sin embargo, si k es elegida aleatoriamente, la probabilidad de que $r = 0$ es despreciablemente pequeña.

Cálculos modulares

1. Convertir la cadena binaria e a un entero $\bar{e}$.
2. Calcular $s := k^{-1}(\bar{e} + rd) \bmod n$.

Conversión de entero a cadena de bytes

1. Convertir r a una cadena binaria de longitud h. Añadir una cadena binaria de ceros de longitud $8f - h$ a su izquierda, y convertir la cadena binaria resultante de $8f$ bits en una cadena de bytes R de longitud f.
2. Convertir s a una cadena binaria de longitud h. Añadir una cadena binaria de ceros de longitud $8f - h$ a su izquierda, y convertir la cadena binaria resultante de $8f$ bits en una cadena de bytes S de longitud f.
3. La firma de M es la cadena de bytes

$$SI := R \parallel S$$

de longitud $2f$.

7.7.5. Verificación de firma digital ECDSA

En esta sección describiremos el proceso de verificación de firma digital ECDSA.

El proceso de verificación de firma consiste de cuatro pasos: la conversión de cadena de bytes a entero, el resúmen de mensaje, cálculos sobre la curva elíptica, y la verificación.

La entrada al proceso de verficación de la firma es:

- Una cadena de bytes M, el mensaje.
- Una cadena de bytes SI, la firma de M.
- Dos elementos a y b del campo, los cuales describen a la ecuación de la curva elíptica.
- Dos elementos del campo x_P y y_P, los cuales definen al punto $P = (x_P, y_P)$.
- Un entero n, el orden de P.
- Dos elementos x_Q y y_Q del campo, que definen al punto $Q = (x_Q, y_Q)$ que es la clave pública.

Conversión de cadena de bytes a entero

1. Hacer parsing de la cadena de bytes SI para obtener las dos cadenas de bytes R y S cada una de longitud f:

$$SI := R \parallel S.$$

2. Convertir R a una cadena binaria de longitud $8f$. Descartar sus $8f - h$ bits más significativos y convertir el resultado a un entero r.
3. Convertir S a una cadena binaria de longitud $8f$. Descartar sus $8f - h$ bits más significativos y convertir el resultado a un entero s.

Resúmen del mensaje

1. Convertir M a una cadena binaria M'.
2. Calcular el valor de hash $e = H(M')$. Si H es, por ejemplo, el algoritmo SHA-1, entonces el valor de hash e es una cadena binaria de longitud 160.

Cálculos sobre la curva elíptica

1. Calcular $r \bmod n$; si éste es 0 entonces rechazar la firma.
2. Convertir la cadena binaria e a un entero $\bar{e}$.
3. Calcular $s^{-1} \bmod n$.
4. Calcular $u := s^{-1}\bar{e} \bmod n$ y $v := s^{-1}r \bmod n$.

5. Calcular el punto $(x_1, y_1) := uP + vQ$ de la curva elíptica, donde $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$.

Verificación

1. Convertir la cadena binaria x_1 de longitud t a un entero $\bar{x_1}$.
2. Hacer $r' := \bar{x_1} \bmod n$.
3. Aceptar la firma como válida sí y solamente si los enteros r y r' son iguales.

7.7.6. Ejemplo de firma digital ECDSA

A continuación mostraremos un ejemplo sencillo de firma digital ECDSA. Utilizando los mismos parámetros del ejemplo 7.6.3, A firma el mensaje $M = 11100011010111100$.

Generación de la firma ECDSA

Supongamos que el valor de hash de M es $H(M) = e = 7$. A realiza los siguientes pasos:

1. Selecciona un entero aleatorio $k = 2$ en el intervalo $[1, 4]$.
2. Calcula $(x_1, y_1) = 2P = (\alpha, \alpha)$.
3. Representa $x_1 = \alpha = (1100)$ como el entero $8 + 4 = 12$.
4. Hace $r = x_1 \bmod n = 12 \bmod 5 = 2$.
5. Calcula $s = k^{-1}(e + rd) \bmod n = 3 \cdot (7 + 2 \cdot 3) \bmod 5 = 4$.
6. La firma digital del mensaje M es $(r, s) = (2, 4)$.

Verificación de la firma ECDSA

La entidad B verifica la firma $(r, s) = (2, 4)$ sobre M como se indica a continuación.

1. B obtiene la clave pública de A: $Q = (\alpha, 0)$.
2. B calcula $H(M) = e = 7$.
3. B calcula $s^{-1}e \bmod n = 3$ y $v = s^{-1}r \bmod n = 3$.

4. B calcula el punto $(x_1, y_1) = uP + vQ = 3P + 3Q = (\alpha, 0) + (\alpha^5, \alpha^3) = (\alpha, \alpha)$.

5. B representa $x_1 = \alpha = (1100)$ como el entero $8 + 4 = 12$.

6. B calcula $r' = (x_1 \bmod n) = 2$.

7. B acepta la firma como válida dado que $r' = r = 2$.

7.7.7. Firma digital ECDSA sobre K-163

En los ejemplos anteriores se han tomado deliberadamente campos subyacentes pequeños, con el propósito de ilustrar mejor las operaciones efectuadas. A continuación ilustraremos el esquema de firma digital ECDSA utilizando la curva K-163. Todos los cálculos de esta sección fueron realizados mediante el código presentado en el apéndice C.

Generación de la firma

Supongamos que el principal A desea firmar el mensaje M cuya función de hash es

$$H(M) = h = 12345678901234567890.$$

Para calcular la firma digital A realiza lo siguiente:

1. Selecciona un número aleatorio $k \in [0, n]$,

$$k = 4963990331444374363255374326041000991501980822380$$

2. Calcula kG obteniendo el punto:

$$0x08e510043b59f0f92531622beceb51718a202f034$$
$$0x802276593b343e28f3ebc03afc34b0b856f3e97f4$$

3. Convierte a un valor entero r la coordenada x del punto kG,

$$r = 279428568760283366372638112708924061506968533137$$

4. Calcula

$$k^{-1} = 472880669986248654029182150888417131693184937909$$

5. Calcula

$$s = 89080963972761694150836476953288081760839441372 2$$

6. La firma digital es el par de números:

$$27942856876028336637263811270892406150696853313 7$$
$$89080963972761694150836476953288081760839441372 2$$

Verificación de la firma digital

Para verificar la firma de A en el mensaje M, el principal B hace lo siguiente:

1. Calcula $w = s^{-1}$ mód n,

$$w = 257153151217052769536553477763856234433133225805 1.$$

2. Obtiene $h = H(M)$ y $u_1 = h \cdot w$ mód n

$$u_1 = 414512041399756539636171642690210638310997297418 9.$$

3. Calcula $u_2 = r \cdot w$ mód n,

$$u_2 = 333843947216046125870409869479196031820714055165.$$

4. Hace $X = u_1G + u_2Q$ obteniendo el punto

$$0x08e510043b59f0f92531622beceb51718a202f034$$
$$0x802276593b343e28f3ebc03afc34b0b856f3e97f4$$

5. La coordenada x del punto anterior se convierte a su valor entero:

$$v = 27942856876028336637263811270892406150696853313 7$$

6. Como $r = v$, la firma es correcta.

7.8. Protocolo de acuerdo de clave (ECKEP)

En esta sección describiremos un protocolo mediante el cual dos entidades A y B establecen una clave secreta compartida K denominada la *clave de sesión*. La clave de sesión puede ser usada subsecuentemente para algún objetivo criptográfico, tal como privacia o autenticación. Este protocolo es conocido como el protocolo de acuerdo de clave basado en curvas elípticas o ECKEP (*Elliptic Curve Key Establishment Protocol*) [5].

Se supone que A y B utilizan los mismos parámetros de la curva elíptica: el campo subyacente $GF(q)$, la ecuación de la curva elíptica E, un punto P sobre la curva y n el orden de P. El principal A tiene la clave privada d_A y la clave pública $Q_A = d_A P = (x_A, y_A)$. El principal B tiene la clave privada d_B y la clave pública $Q_B = d_B P = (x_B, y_B)$.

El protocolo consiste en los siguientes pasos:

1. La entidad A hace lo siguiente:
   - *a*) Selecciona un entero aleatorio k_A, $1 \leq k_A \leq n-1$.
   - *b*) Calcula el punto $(x_1, y_1) = R_A := k_A P$.
   - *c*) A envia R_A a B.
2. La entidad B hace lo siguiente:
   - *a*) Selecciona un entero aleatorio, k_B, $1 \leq k_B \leq n-1$.
   - *b*) Calcula el punto $(x_2, y_2) = R_B := k_B P$.
   - *c*) B envia R_B.
3. A hace lo siguiente:
   - *a*) Calcula el entero $s_A := k_A + x_1 d_A x_A \bmod n$.
   - *b*) Calcula la clave de sesión $K := s_A(R_B + x_2 x_B Q_B)$.
4. B hace lo siguiente:
   - *a*) Calcula el entero $s_B := k_B + x_1 d_B x_B \bmod n$.
   - *b*) Calcula la clave de sesión $K := s_B(R_A + x_1 x_A Q_A)$.

Como puede observarse, el protocolo proporciona implicitamente autenticación de clave mutua. Tal como se ha descrito, no contiene la parte de confirmación de la clave compartida. Algunas otras características de este protocolo es que es *no interactivo* (los mensajes transmitidos entre ambos

participantes son independientes uno del otro), *simétrico en roles* (los mensajes enviados por ambos participantes tienen la misma estructura), no requiere cifrado, funciones de hash o marcas de tiempo, ofrece *secreto posterior perfecto*, y tiene bajos requerimientos de ancho de banda.