Firmas digitales

Trabajo Fin de Grado

Autor:
Alejandro García López

Tutor:
Juan Antonio López Ramos

Grado en Matemáticas

Junio, 2021
Universidad de Almería

Índice general

Abstract in English

A digital signature is a simulation of the graphic signature with which personal documents are usually authenticated, with the particularity that it must be done digitally and that no one can sign for another. The digital signature comes to solve some of the main problems with the treatment of the information, as it happens with its analogical version: authentication of the origin of the information and non-repudiation of it. But in addition, in the digital case, the integrity of the information can also be ensured, thus managing to respond to three of the main objects of Cryptography.

In this work, the main digital signatures currently used appear. Digital signatures are based on some cryptosystems, which are studied before introducing the concept of digital signature.

Once the main types of digital signatures are understood, a somewhat more complex type of signature is explained: the blind digital signature. This type of signature is very important, as it allows to sign certain information and also guarantees anonymity. Its use is common in the field of electronic voting or even in payment with digital cash.

In recent years, cryptocurrencies are a very hot topic based on cryptography. They differ from conventional currencies or electronic money in that they are not issued by central banks, although they can offer the three traditional functions of money (unit of account, means of payment and store of value). Its creation and exchange are based on the blockchain technology. In this work, the operation of most of them is explained, and, in addition, an introduction and a deeper explanation is given for the most famous and usual cryptocurrency: Bitcoin, which bases, functions, structure and operation will be outlined in the final part of this document.

Resumen en español

Una firma digital es una simulación de la rúbrica gráfica con la que usualmente son autenticados los documentos personales, con la particularidad de que esta ha de ser realizada digitalmente y que nadie pueda firmar por otro. La firma digital viene a solucionar algunos de los principales problemas con el tratamiento de la información, tal cual ocurre con su versión analógica: autenticación de la procedencia de la información y no repudio de la misma. Pero además, en el caso digital, se puede asegurar también la integridad de la información, consiguiendo así dar respuesta a tres de los principales objetos de la Criptografía.

En este trabajo, aparecen las principales firmas digitales usadas actualmente. Las firmas digitales se basan en algunos criptosistemas, que son estudiados antes de introducir el concepto de firma digital.

Una vez entendidos los principales tipos de firmas digitales, se explica un tipo de firma algo más compleja: la firma digital ciega. Este tipo de firma es muy importante, pues permite firmar cierta información y además garantiza el anonimato. Su uso es común en el ámbito de las votaciones electrónicas o incluso en el pago con dinero electrónico.

Otro uso de la Criptografía de gran actualidad es el de las criptomonedas. La diferencia principal respecto de las monedas convencionales o incluso el dinero electrónico es que no son emitidas por bancos centrales, si bien puede fungir las tres funciones tradicionales del dinero (unidad de cuenta, medio de pago y depósito de valor). Su creación e intercambio se basan en la tecnología del blockchain o cadena de bloques. En el trabajo se explica su funcionamiento, cuya base, como podrá comprobarse, es también el de la firma digital.

1 Introducción

Las firmas digitales, objeto de este trabajo, son mecanismos de la Criptografía que se utilizan para corroborar la autoría y autenticidad de un documento o de una información.

Desde hace miles de años, la humanidad siempre ha querido poder autenticar la validez y la autoría de información. Hace siglos, estas tareas eran llevadas a cabo mediante firmas manuales, ya sean las manuscritas, las firmas de cera de la época medieval, etc.

Se define la sociedad de hoy día como la sociedad de la información, basada en el uso masivo de las tecnologías de la información y las comunicaciones. Esto ha supuesto una revolución en el esquema clásico de los elementos de la comunicación que estudiara el lingüista ruso Roman Jakobson en el siglo pasado.

La Criptografía es un asunto recurrente hoy día, de hecho, hay poca gente que no haya oído hablar de ella. Sin embargo, la mayoría de personas desconocen cómo funciona esta ciencia. En este trabajo trataremos de orientar al lector de tal forma que, sin tener conocimiento previo de esta materia, pueda adquirir ciertos rudimentos sobre ella y, concretamente sobre las firmas digitales. Presentaremos los conceptos básicos de la Criptografía, sus objetivos, los principales criptosistemas, todo ello, como base principal de este trabajo; las firmas digitales. Por tanto, lo primero que haremos será dar la definición de varios de los conceptos de la Criptografía.

En este capítulo veremos los conceptos básicos de la Criptografía, así como sus principales objetivos. Además, introduciremos los tipos de criptosistemas, y comenzaremos a nombrar algunos de ellos.

La Criptografía es la ciencia que se encarga de cifrar cierta información, de tal forma, que solo un conjunto de personas autorizadas pueda acceder a ella o a una parte de ella.

En esta definición encontramos otra palabra clave; cifrar, que consiste en, mediante una clave, transcribir una información de tal forma que, sin las herramientas adecuadas, no sea posible decodificarla.

Una clave es un conjunto de signos, reglas y normas que sirven para transcribir información de manera que sin tal clave u otra clave válida sea imposible la obtención de dicha información.

Aclarado el concepto de Criptografía, toca definir su objeto. La Criptografía tiene diversos objetivos que se pueden resumir en estos cuatro:

1. Confidencialidad: evitar que cierta información sea accesible por terceras personas no autorizadas a ello.

2. Integridad de los datos: capacidad de asegurar el contenido de cierta información no ha sido alterado en el proceso de envío desde la fuente hasta el destino, o desde su almacenamiento original, hasta que es consultada en cualquier momento posterior.

3. No repudio: capacidad de evitar la negación sobre la autoría de cierta información.

4. Autentificación: capacidad de asegurar el origen de cierta información.

Hemos hablado de los objetivos de la Criptografía, pero, ¿como funciona la Criptografía?

La Criptografía surge hace miles de años, sin embargo, no es hasta la era informática que esta llega a su máximo esplendor. Con la llegada de los ordenadores llega la

posibilidad de realizar cálculos de forma más rápida y sencilla. Por eso, que el mayor logro, y el mayor descubrimiento de la Criptografía surge en 1976, cuando Diffie y Hellman publican su artículo [5], en el que se introducen algunos métodos para la encriptación de información. Es en este punto, cuando se define quizás el concepto más importante de toda la Criptografía, el criptosistema.

Un criptosistema es un conjunto compuesto por varios conjuntos a su vez: un conjunto de posibles textos; un conjunto de posibles claves a usar; una forma de transformar, mediante una de las claves anteriores, los posibles textos en textos ininteligibles, que constituirán a su vez un conjunto de posibles textos cifrados, y una forma de transformar un texto cifrado en un texto original.

A continuación, procedemos a introducir los dos tipos principales de Criptografía: la Criptografía simétrica y la Criptografía de clave pública o asimétrica.

La diferencia principal entre ambas consiste en que la Criptografía simétrica la clave de cifrado y descifrado son la misma o, conociendo una, es computacionalmente sencillo obtener la otra. Sin embargo, en la Criptografía asimétrica se basa en que, la clave de cifrado es pública, es decir, cualquier persona puede acceder a ella, pero la de descifrado es únicamente conocida por el receptor.

Esta última surge del problema de que en la Criptografía simétrica, emisor y receptor deben compartir al menos una de las claves (ya sea cifrado o descifrado) por un canal seguro, cuya existencia no está garantizada. Por ello, y como veremos más adelante, la Criptografía pública, emplea criptosistemas de tal forma que sea computacionalmente inviable obtener la clave de cifrado poseyendo la de descifrado, o viceversa.

Es precisamente la Criptografía de clave pública el germen y base del objeto de estudio de este trabajo, las firmas digitales. Dado que en este tipo de Criptografía existen dos claves de carácter distinto, una pública, conocida por cualquiera y otra privada, mantenida en secreto por su legal poseedor, hará posible generar, mediante la clave privada, de forma única e imposible de falsificar, una pieza de información adicional a la información transmitida o almacenada, que permita verificar su procedencia, integridad y la imposibilidad de negar dicha procedencia por parte de cualquiera, mediante el uso de la clave pública.

2 Conceptos clave de la Criptografía

El siguiente capítulo está dedicado a introducir las definiciones básicas, así como los principales criptosistemas utilizados para la generación de firmas digitales.

Definición 2.1:[5] Un criptosistema es una 5-tupla (P, C, K, E, D) que cumple las siguietes condiciones:

1. P es un conjunto finito de posibilidades de texto sin formato.
2. C es un conjunto finito de posibilidades de texto cifrado.
3. K, también llamado K-espacio, es un conjunto finito de posibles claves.
4. Para cada K$\in K$, existe una encriptacion $e_K \in E$ correspondiente a una desencriptación $d_K \in D$ con $e_K : P \to C$ y $d_K : C \to P$, tal que $d_K(e_K(x)) = x$ para cada $x \in P$.

A continuación, veremos un ejemplo ilustrativo en el que se explica el funcionamiento de los criptosistemas en general:

Ejemplo 2.1: imaginemos que dos personas (Bob y Alice) quieren enviarse un mensaje, de tal forma que ninguna otra persona pueda verlo. Una tercera persona (Óscar), quiere descifrar dicho mensaje. ¿Cómo pueden Bob y Alice asegurarse que Óscar no pueda acceder a dicha información?

La respuesta es muy sencilla, Bob y Alice deberán recurrir a un criptosistema de la siguiente forma. Sea x el mensaje que se quiere enviar, Bob, mediante una función $e_K \in E$ encripta el mensaje, de tal forma que nadie puede obtener dicho mensaje. Además, Alice tendrá una clave K, que dará lugar a una función de desencriptado $d_K \in D$, es decir, la función inversa de e_K. Por tanto, Bob envía el mensaje cifrado a Alice, Oscar no puede entender dicho mensaje, ya que el no posee la función d_k, una vez llega a Alice, únicamente aplica la función d_k, obteniendo así x.

Con este sencillo ejemplo, vemos la utilidad que tienen los criptosistemas. Sin embargo, surge una duda, ¿cómo obtiene Alice la clave de descifrado?

Aquí es cuando debemos diferenciar los dos tipos principales de Criptografía ya antes mencionados, la Criptografía de clave pública y la Criptografía simétrica.

Definición 2.2:[5] Consideremos el sistema de encriptado (criptosistema) (P, C, K, E, D) tal que tenemos una serie de transformaciones $\{e_K$ para cada K$\in K\}$ y $\{d_K$ para cada K$\in K\}$ de encriptado y desencriptado respectivamente, siendo K, el K-espacio de claves. Se dice que el criptosistema es simétrico si, para cada función de encriptado e, es computacionalmente sencillo obtener la función de desencriptado d, y viceversa.

El concepto de que una operación o algoritmo sea computacionalmente sencillo indica que el número de operaciones involucradas en el mismo crece de modo polinómico a medida que el tamaño de los datos involucrados.

En la mayoría de casos de criptosistemas, $e = d$, sin embargo, no siempre es así.

Ejemplo 2.2: Imaginemos que queremos cifrar la palabra "PERRO", le aplicaremos una función de encriptado e a cada letra de tal forma que a cada letra le haga corresponder su siguiente en el alfabeto. Por tanto, el mensaje encriptado sería: "QFSSP", como vemos, es casi imposible conocer el significado de este mensaje, sin embargo, si conocemos la clave de encriptado e, nos resulta muy fácil, ya que si e únicamente sumaba una posición en el alfabeto, d deberá restarla.

Veamos ahora el otro tipo de Criptografía, la Criptografía de clave pública.

Definición 2.3:[5] Consideremos un sistema de encriptado (criptosistema) tal que tenemos una serie de transformaciones $\{e_K$ para cada K$\in K\}$ y $\{d_K$ para cada K$\in K\}$ de encriptado y desencriptado respectivamente, siendo K, el K-espacio de claves. Se dice

que el criptosistema es de clave pública si, conociendo la función de encriptado e y el mensaje cifrado c, es, inviable, obtener el mensaje descifrado m conociendo e, es decir, aunque se conozca e es inviable obtener la función d.

Hay que destacar que este descubrimiento en la Criptografía fue una de los mayores avances realizados en la historia. Como habíamos dicho antes, fue propuesto en 1976 por Diffie y Hellman en su artículo [5].

Veamos el potencial que tiene este tipo de criptosistemas.

Ejemplo 2.3: Imaginemos que Bob quiere enviar un mensaje a Alice. Bob recibe una caja fuerte abierta de Alice, una caja fuerte que no puede ser abierta salvo con su llave, la cual solo posee Alice. Bob mete el mensaje en la caja y la cierra, de esta forma, ya solo Alice puede abrir la caja y leer el mensaje, de hecho, aunque Bob quisiera, por ejemplo, porque se le olvida el mensaje, no podría, pues no tiene la llave. Alice recibirá la caja cerrada, la cual ha podido ser vista por otros, pero nadie la ha podido abrir.

Con este ejemplo, la función de encriptado e sería cerrar la caja, esta función la puede conocer cualquiera, por ello es pública, la función de desencriptado d sería el abrir la caja, que, como solo puede ser abierta con la llave, y esta solo la posee Alice, es privada.

Veamos ahora las principales ventajas e inconvenientes de cada uno de los tipos:

Ventajas de la Criptografía simétrica:

Podemos obtener la clave de descifrado d únicamente conociendo la clave de cifrado e, y viceversa. Esto permite enviar mensajes de una manera mucho más fluida.

Tanto emisor como receptor pueden intercambiar sus papeles, ya que ambos conocen la clave de descifrado como de cifrado.

Desventajas de la Criptografía simétrica:

No es 100 % segura, ya que para descifrar el mensaje del emisor, el receptor necesita conocer la clave d, o, en su defecto e, con el objetivo de obtener d. Por tanto, es necesario la existencia de un canal seguro, con el fin del intercambio de claves.

Ventajas de la Criptografía de clave pública:

Es muy segura, pues el receptor es el único poseedor de la clave de descifrado d y es, inviable obtener d por otro método.

Desventajas de la Criptografía de clave pública: Emisor y receptor no pueden intercambiar sus papeles, ya que el emisor no conoce la clave de descifrado d para poder recibir un mensaje el emisor del receptor, sería necesario crear otro criptosistema de clave pública.

Todos los criptosistemas que estudiaremos en este capítulo son de clave pública, pues son el tipo de criptosistemas en los que se basan las firmas digitales.

2.1 El criptosistema RSA

En este apartado definiremos y comentaremos el criptosistema más famoso en la actualidad; el criptosistema RSA [13]. Este criptosistema de clave pública fue desarrollado en 1979 y le debe su nombre a sus creadores Rivest, Shamir y Adleman. A continuación daremos su definición formal, para luego, indicar las ventajas e incovenientes de dicho criptosistema.

Criptosistema 2.1: criptosistema RSA [13]

Sea $n = pq$, con p y q primos.

Supongamos que $P = C = \mathbb{Z}_n$, y definimos $K = \{(n,p,q,a,b) : ab \equiv 1 \ mod(\phi(n))\}$, donde $\phi(n) = (p-1)(q-1)$.

Para K= (n,p,q,a,b) se definen:

$e_k(x) = x^b \ mod(n)$

$d_k(y) = y^a \ mod(n)$

con $x, y \in \mathbb{Z}_n$, y $ab \equiv 1 \ mod(\phi(n))$

El valor (n,b) es la clave pública y el valor (p,q,a) conforma la clave privada.

Veamos ahora que el criptosistema funciona, es decir, veamos que e_k es la inversa de d_k, o, equivalentemente, que e_k y d_k son funciones de encriptado y desencriptado válidas respectivamente.

Supongamos que $P = C = \mathbb{Z}_n$. Sea $x \in P$, sean $a, b \in \mathbb{Z}$ tales que $ab \equiv 1 \ mod(\phi(n))$. Entonces:

Como $ab \equiv 1 \ mod(\phi(n))$, entonces podemos escribir $ab = t\phi(n) + 1$ con $t \geq 1$ y $t \in \mathbb{Z}$. Tenemos entonces:

$$e_k(x) = x^b$$

y por tanto:

$$d_k(e_k(x)) = (x^b)^a \ mod(n) = x^{ab} \ mod(n) = x^{t\phi(n)+1} mod(n) = x^{t\phi(n)} x \ mod(n) \overset{(1)}{=} 1^t x \ mod(n) = x \ mod(n) \overset{(2)}{=} x$$

Donde hemos empleado los siguientes resultados:

(1) teorema de Euler[8]:

Teorema 2.2: teorema de Euler

Si a y n son enteros primos relativos, entonces $a^{\phi(n)} \equiv 1 (mod n)$

(2) como $x \in P = \mathbb{Z}_n$, entonces $x(mod(n)) = x$

Implemetación del criptosistema RSA: [8]

Veamos ahora los pasos que debemos realizar para establecer un criptosistema RSA:

1. Generar dos primos grandes, p y q, con $p \neq q$.
2. Con p y q obtenemos n, y $\phi(n) = (p-1)(q-1)$.
3. Elegimos un número aleatorio b (con $1 < b < \phi(n)$) de tal forma que $mcd(b, \phi(n)) = 1$.
4. Calculamos $a = b^{-1} \ mod(\phi(n)$.
5. La clave pública es (n,b) y la clave privada es (p,q,a)

Ejemplo 2.1: supongamos que Bob quiere recibir un mensaje de Alice. Bob elige dos números primos p y q, por ejemplo, $p = 5$ y $q = 7$ con ellos calcula $n = p \cdot q = 5 \cdot 7 = 35$, a continuación Bob calcula $\phi(n) = (p-1)(q-1) = 4 \cdot 6 = 24$.

Más adelante, Bob elige un b que cumpla que $mcd(\phi(n), b) = 1$ (siendo mcd el máximo común divisor).

Escoge $b = 7$ y calcula $b^{-1} \ mod(\phi(n) = 7^{-1} \ mod(24) = 7$. Por tanto, $a = 7$.

Bob publica los valores de n y b, es decir, 35 y 18. Imaginemos que Alice quiere enviar el mensaje "2", lo único que debería hacer es calcular $e_k(2) = 2^7 \ mod(35) = 23$. Bob recibe entonces el mensaje "29", y ahora calcula $d_k(23) = 23^7 \ mod(35) = 2$.

La seguridad actualmente del criptosistema RSA se basa en escoger primos de gran longitud, al menos de 1024 bits. Veamos ahora que, conociendo n y b, encontrar $\phi(n)$ o a es computacionalmente equivalente a factorizar n. Veamoslo con la siguiente proposición:

Proposición 2.1:[8] sea $n = pq$, siendo p y q dos números primos distintos. Si conocemos n y $\phi(n)$ es sencillo obtener p y q.

Demostración:

Claramente:

$$n - \phi(n) + 1) = pq - (p-1)(q-1) + 1 = pq - pq + p + q - 1 + 1 = p + q$$

Además, sabemos que pq y $p+q$ son las raíces de la ecuación:

$$x^2 - (n - \phi(n) + 1)x + n = x^2 - (p+q)x + pq = (x-p)(x-q).$$

Además, podemos calcularlos de la siguiente forma:

$$p, q = \frac{(n-\phi(n)+1 \pm \sqrt{(n-\phi(n)+1)^2 - 4n}}{2}$$

Obteniendo así p y q.

■

Para la demostración de la siguiente proposición usaremos el siguiente algoritmo teórico:

Algoritmo 2.1: factorización del exponente universal:[8]

Supongamos que tenemos un exponente r tal que $a^r \equiv 1 \ mod(n)$ para todo a tal que $mcd(a, n) = 1$. Escribimos $r = 2^k m$ con m un número impar. Se escoge un número aleatorio a tal que $1 < a < n - 1$. Si $mcd(n, a) \neq 1$, tenemos un divisor de n, por tanto, supongamos que $mcd(n, a) = 1$. Sea $b_0 = a^m \ mod(n)$, y definimos $b_{u+1} \equiv b_u^2 \ mod(n)$ para todo $0 \leq u \leq k-1$. Si $b_0 \equiv 1 \ mod(n)$, se para y se prueba otro a. Si, para algún u se tiene que $b_u \equiv -1 \ mod(n)$, se para y se prueba otro a. Si, para cierto u, $b_{u+1} \equiv 1 \ mod(n)$ pero $b_u \not\equiv \pm 1 \ mod(n)$ entonces $mcd(b_u - 1, n) \neq 1$, y por tanto, nos da un divisor de n.

Probando algunos valores de a, se tienen unas altas probabilidades de factorizar n.

Es importante recalcar que este algoritmo no se usa en la práctica, y solo tiene valor a nivel teórico, esto es debido a que es casi imposible encontrar el exponente r cuando se trata de números grandes.

Proposición 2.2: [8] Si conocemos a y b, es probable que se pueda obtener n.

Demostración:

En al algoritmo 2.1, hemos visto que si tenemos un exponente $r > 0$ tal que $x^r \equiv 1 \ mod(n)$ para todo a con $mcd(a, n) = 1$, entonces probablemente podamos factorizar n.

Tenemos que $ab \equiv 1 \ mod(\phi(n))$, por tanto, $ab - 1$ es múltiplo de $\phi(n)$, entonces podemos escribir $ab - 1 = k\phi(n)$ con $k \in \mathbb{Z}$. Por tanto:

$$x^{ab-1} \equiv x^{k\phi(n)} \equiv (x^{\phi(n)})^k \ mod(n)$$

cuando $mcd(a, n) = 1$. Entonces podemos aplicar el método del algoritmo universal, y, por tanto, probablemente se pueda factorizar n.

■

2.2 El criptosistema Rabin

El caso de $b = 2$, no está permitido en el criptosistema RSA, esto es debido a que $\phi(n)$ es un número par, por tanto b no lo puede ser. Ahora, ¿cómo obtenemos b?, ¿tenemos que ir probando números y calculado su máximo común divisor con $\phi(n)$ hasta que nos salga 1? Por suerte, la respuesta es no, cuando no sea posible el cálculo de b, se emplea el criptosistema Rabin[12].

Criptosistema 2.2: Criptosistema Rabin Sea $n = p \cdot q$, donde p y q son primos y $p, q \equiv 3\ (mod(4))$. Tomamos $P = C = \mathbb{Z}_n^*$, se define $K = \{(n, p, q)\}$.

Para un cierto k= (n, p, q), se define:

$e_K(x) = x^2\ (mod(n))$

$d_K(y) = \sqrt{y}\ (mod(n))$

La clave pública es n y la clave privada es (p, q).

Claramente e_K es inversa de d_K, por tanto el criptosistema funciona.

Antes de dar un ejemplo del criptosistema Rabin, haremos un repaso general acerca de la aritmética modular relacionada con las raíces cuadradas en $\mathbb{Z}_n$. Podrán encontrar esta más información acerca de las raíces cuadradas en $\mathbb{Z}_n$ en [17].

Proposición 2.3: sea p un número primo tal que $p \equiv 3\ mod(4)$ y sea y un entero. Sea $x = y^{\frac{p+1}{4}}\ mod(p)$. Entonces se tiene:

1. Si y tiene una raíz cuadrada $mod(p)$, entonces las raíces cuadradas de $y\ mod(p)$ son $\pm x$.

2. Si y no tiene raíces cuadradas $mod(p)$, entonces -y tiene raices cuadradas $mod(p)$, y estas son $\pm x$.

Demostración:

Si $y \equiv 0\ mod(p)$, ambos resultados son triviales. Por tanto, supongamos $y \not\equiv 0\ mod(p)$. El teorema de Fermat nos asegura que $y^{p-1} \equiv 1\ mod(p)$. Por tanto:

$$x^4 \equiv y^{p+1} \equiv y^2 y^{p-1} \equiv y^2\ mod(p)$$

Por tanto, $(x^2 + y)(x^2 - y) \equiv 0\ mod(p)$, por tanto $x^2 \equiv \pm y\ mod(p)$. Por tanto, y o -y tienen raíces $mod(p)$. Supongamos ahora que ambas tienen raíces $mod(p)$, se tiene entonces que $y \equiv a^2\ mod(p)$ y $-y \equiv b^2\ mod(p)$. Por tanto (dado que $y \not\equiv 0\ mod(p)$), $(\frac{a}{b})^2 \equiv -1\ mod(p)$. Por tanto, -1 es una raíz cuadrada $mod(p)$. Pero esto es absurdo si $p \equiv 3\ mod(4)$. Nos queda enconces que solo y o -y tienen raíces $mod(p)$. Además, si y tiene una raíz cuadrada $mod(p)$, entonces $y \equiv x^2$, y las dos raíces cuadradas son $\pm x$. Si -y tiene raíces cuadradas $mod(p)$, entonces $x^2 \equiv -y$.

■

Veamos un ejemplo inseguro del criptosistema Rabin, donde veremos más claro el cálculo de raíces $mod(p)$.

Ejemplo 2.2: Supongamos $n = 77 = 11 \cdot 7$. Se tiene enconces que la función de encriptado es $e_K(x) = x^2\ mod(77)$ y la función de desencriptado es $d_K(y) = \sqrt{y}\ mod(77)$.

Supongamos que Bob va a enviar el mensaje $x = 32$ a Alice.

Primero Bob encripta el mensaje, mediante e_K, por tanto:

$$e_K(32) = 32^2\ mod(77) = 23\ mod(77).$$

Bob comparte públicamente el valor de $n = 77$. Ahora Alice debe desencriptar el mensaje, por tanto aplica la función d_K:

$$d_K(23) = \sqrt{23}\ mod(77).$$

Calculamos ahora:
$t = 23^{\frac{7+1}{4}}\ mod(7)$
$t = 23^{\frac{11+1}{4}}\ mod(11)$
Resolvamos ahora el sistema:

$$\begin{cases} x^2 \equiv 4\ mod(7) \\ x^2 \equiv 1\ mod(11) \end{cases}$$

Para ello emplearemos el algoritmo chino del resto [17], el cual nos dice lo siguiente:
Sea el sistema de congruencias:

$$\begin{cases} x \equiv c_1 mod(m_1) \\ x \equiv c_2 mod(m_2) \\ \quad . \\ \quad . \\ \quad . \\ x \equiv c_n mod(m_n) \end{cases}$$

1. Se calcula $M = m_1 \cdot m_2 \cdot ... \cdot m_n$
2. Para cada $= 1, 2, ..., n$, se calcula:
$M_k = \frac{M}{m_k}$
$u_k = M_k^{-1}\ mod(m_k)$
3. El número $z = c_1 \cdot M_1 \cdot u_1 + c_2 \cdot M_2 \cdot u_2 + ... + c_n \cdot M_n \cdot u_n$ es solución del sistema de congruencias.
4. El número $r = z\ mod(M)$, es solución del sistema de congruencias y $0 \leq r < M$.
Continuamos con el ejemplo:
1. Calculamos $M = 7 \cdot 11 = 77$.
2. Calculamos m_k y u_k para cada $k = 1, 2$:
$M_1 = \frac{77}{7} = 11 \Longrightarrow u_1 = 11^{-1}\ mod(7) = 2$
$M_2 = \frac{77}{11} = 7 \Longrightarrow u_1 = 7^{-1}\ mod(11) = 8$
3. El número $z = 4 \cdot 11 \cdot 2 + 7 \cdot 8 = 144$.
El número $z' = z = 4 \cdot 11 \cdot 2 - 7 \cdot 8 = 32$
4. El número $r_1 = 144\ mod(77) = 67$ es solución del sistema.
El número $r_2 = 32\ mod(77) = 32$ es solución del sistema.
Como estamos trabajando con x^2, los opuestos también serán raíces, por tanto:
El número $r_3 = -67\ mod(77) = 10$ es solución del sistema.
El número $r_4 = -32\ mod(77) = 45$ es solución del sistema.
Como vemos uno de los resultados coincide con el texto sin cifrar, usualmente Bob le dará indicaciones a Alice sobre cuál de los cuatro números escoger.

2.3 Criptosistemas basados en el problema del logaritmo discreto. El criptosistema de ElGamal

Anteriormente hemos presentado dos criptosistemas cuya seguridad de basa en el problema que surge para factorizar un número en dos números primos, pero hay muchos otros tipos de criptosistemas de clave pública, en este capítulo comentaremos el denominado problema del logaritmo discreto, y presentaremos el principal criptosistema basado en este problema; el criptosistema de ElGamal[6].

El problema del logaritmo discreto

En este apartado, veremos la definición formal del problema del logaritmo discreto [16], con el fin de ilustrar la base del criptosistema de ElGamal. El problema se presenta así:

Sea $(G,\cdot)$ un grupo multiplicativo, sea $\alpha \in G$ un elemento de orden n, y $\beta \in <\alpha>$.

Encuentre el único entero a, con $0 \leq a \leq n-1$, tal que $\alpha^a = \beta$, o, equivalentemente $a = \log_\alpha \beta$ A esta expresión se le conoce como el logaritmo discreto de β.

Este problema resulta muy útil en la Criptografía, pues resolver dicho problema es computacionalmente costoso, y a veces inviable, sin embargo, la operación contraria es computacionalmente eficiente, es decir, no requiere un gran gasto computacional.

Criptosistema 2.3: criptosistema de ElGamal en $\mathbb{Z}_p^*$[6]

Sea p un número primo tal que la resolución del problema del logaritmo discreto en $(\mathbb{Z}_p^*,\cdot)$ sea inviable. Sea $\alpha \in \mathbb{Z}_p^*$ un elemento primitivo. Sea $P = \mathbb{Z}_n^*$, y $C = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$., y definimos $K = \{(p,\alpha,a,\beta : \beta \equiv \alpha^a\ (mod(p))$.

Para cierto K=(p,α,a,β), y para un número secreto $k \in \mathbb{Z}_{p-1}$, definimos: $e_K = (y_1,y_2)$, siendo (y_1,y_2):

$$y_1 = \alpha^k\ mod(p)$$
$$y_2 = x\beta^k\ mod(p)$$

Para ciertos $y_1,y_2 \in \mathbb{Z}_p^*$, se define:

$$d_K(y_1,y_2) = y_2(y_1^a)^{-1}\ mod(p)$$

Los valores p,α y β son la clave pública, a es la clave privada.

Veamos ahora que el criptosistema funciona, es decir, que la función de encriptado e_K es la inversa de la función de desencriptado d_K.

Sea p un número primo tal que la resolución del problema del logaritmo discreto en $(\mathbb{Z}_p^*,\cdot)$, sea inviable. Sea $\alpha \in \mathbb{Z}_p^*$ un elemento primitivo. Sea $P = \mathbb{Z}_n^*$, y $C = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$., y definimos $K = \{(p,\alpha,a,\beta : \beta \equiv \alpha^a\ (mod(p))$. Sea $k \in \mathbb{Z}_{p-1}$. Sea $x \in P$ un mensaje sin cifrar, se tiene entonces:

$$e_K(x,k) = (y_1,y_2) = (\alpha^k\ mod(p), x\beta^k\ mod(p))$$

Por tanto, y teniendo en cuenta que $\beta \equiv \alpha^a\ (mod(p))$, obtenemos:

$$d_K(e_K(x)) = d_K(\alpha^k \ mod(p), x\beta^k \ mod(p)) = x\beta^k \cdot [(\alpha^k)^a]^{-1} mod(p) =$$
$$= \mathrm{x}(\alpha^k)^a \cdot [(\alpha^k)^a]^{-1} mod(p) = x \ mod(p)$$

Por tanto d_K es la inversa de e_K. En conclusión el criptosistema funciona.

Implementación del criptosistema de ElGamal: Supongamos que Alice quier enviar un mensaje x a Bob. Se seguirán los siguientes pasos:

1. Alice obtiene la clave pública (p, α, β).
2. Alice escoge un número al azar $k \in \mathbb{Z}_{p-1}$.
3. Alice calcula $(y_1, y_2) = (\alpha^k \ mod(p), x\beta^k \ mod(p))$.
4. Alice envía el mensaje (y_1, y_2) a Bob.
5. Bob aplica la función de desencriptado d_K, usando la clave privada a.

A continuación, veremos un ejemplo (inseguro) de este criptosistema.

Ejemplo 2.3:

Imaginemos que Bob quiere recibir un mensaje $x = 8$ de Alice. Bob escoge un $p = 9$, un $a = 5$ y $\alpha = 2$ elemento primitivo de $\mathbb{Z}_9$, ahora Bob calcula $\beta = \alpha^a \ mod(9) = 2^5 \ mod(9) = 5$ comparte la clave pública $(p, \alpha, \beta) = (9, 2, 5)$.

Alice escoge el número $k = 3$, aunque podría haber escogido cualquier otro de $\mathbb{Z}_8$. A continuación, ella calcula el par $(y_1, y_2) = (\alpha^k \ mod(p), x\beta^k \ mod(p)) = (2^3 \ mod(9), 8 \cdot 5^3 \ mod(9)) = (8, 1)$. Alice envía dichos valores a Bob.

Por último, Bob únicamente tiene que aplicar la función de desencriptado.

$d_K = y_2(y_1^a)^{-1} \ mod(p) = (8^5)^{-1} \ mod(p) = 8^{-1} \ mod(p) = 8 = x$

3

Firmas digitales

A lo largo de la historia de la humanidad, las personas hemos ido usando distintos tipos de firmas, desde las firmas de cera de la nobleza medieval, hasta las firmas que todos conocemos hoy en día.

El principal objetivo de las firmas es corroborar la autoría de un documento, cuadro, estudio, etc. Para ello, es importante que la firma solo la pueda realizar el autor de ella, pues si no, no habría forma de comprobar la autoría.

Sin embargo, con el apogeo de las nuevas tecnologías, y la evolución de la Criptografía y la seguridad informática, los métodos convencionales de firmas se quedaron atrás. El por qué es muy sencillo de entender, imaginémonos que queremos firmar un contrato de piso, imprimimos el contrato, lo firmamos manualmente, lo escaneamos y lo volvemos a enviar. El proceso, pese a ser muy sencillo, es inservible si hablamos en términos de seguridad, cualquier persona que tenga acceso a tu firma manual, podría firmar haciéndose pasar por nosotros.

Por tanto, una propiedad que deben tener las firmas digitales es que la firma no solo tiene que estar asociada al firmante, si no también al mensaje que se manda, es decir, una firma digital en un contrato identifica al firmante y al contrato.

El proceso para crear una firma digital consta de dos pasos: el proceso de firmado y el proceso de verificación.

Definición 3.1:[16] un esquema de una firma digital es una 5-tupla *(P,A,K,S,V)*, que cumple las siguientes condiciones:

1. P es un conjunto finito de posibles mensajes.
2. A es un conjunto finito de posibles firmas.
3. K, también llamado K-espacio, es un conjunto finito de posibles claves.
4. Para cada $K \in K$, hay una función de firma $s_K \in S$, y una correspondiente función de verificación $v_K \in V$. Además, $s_K : P \longrightarrow A$ y $v_K : P \times A \longrightarrow \{true, false\}$ son funciones que cumplen que para cada mensaje $x \in P$ y para cada firma $y \in A$:

$$v_K(x,y) = \begin{cases} true & si \quad y = s_K(x) \\ \\ false & si \quad y \neq s_K(x) \end{cases}$$

El par (x,y) con $x \in P$ e $y \in A$ es lo que se conoce como mensaje firmado.

Para cada $K \in K$, las funciones s_K y v_K deben ser funciones de tiempo polinomial.

La función de verificación v_K es pública, y la función de firma s_K es privada. Además, teniendo un mensaje x debe ser, computacionalmente inviable obtener un y', con $y' = s_K(x)$ para cierto K, tal que $v_K(y') = true$. Si alguien es capaz de obtener un y' tal que $v_K(x,y') = true$, entonces a y' se le conoce como falsificación. Una falsificación y' es una firma válida, es decir, $v_K(x,y') = true$, que no ha sido creada por el firmante.

3.1 Firmas digitales basadas en el criptosistema RSA

Alice firma un mensaje x usando la función de desencriptado del criptosistema RSA d_k. Alice es la única que puede hacerlo, recordemos que $d_K = s_K$ es privada. El algoritmo de verificado se basa en la función e_K, todo el mundo puede verificar el mensaje firmado de Alice, pues e_K es pública.

Sistema de firmado 3.1: sistema de firmado RSA [13]

Sea $n = pq$ con p y q primos.

Supongamos que $P = C = \mathbb{Z}_n$, y definimos $K = \{(n,p,q,a,b) : ab \equiv 1 \ mod(\phi(n))\}$, donde $\phi(n) = (p-1)(q-1)$

Para K= (n,p,q,a,b) se definen:

$s_k(x) = x^a \ mod(n) = y$

$v_K(x,y) = true \Leftrightarrow x \equiv y^b \ mod(n)$

con $x, y \in \mathbb{Z}_n$, y $ab \equiv 1 \ (mod(\phi(n))$

El valor (n,b) es la clave pública y el valor (p,q,a) conforma la clave privada.

Para la verificación de la firma únicamente hay que tener en cuenta que $x^{ab} \equiv x \ mod(n)$.

El principal problema de la firma RSA es que tal cual se encuentra definida no puede proporcionar confidencialidad, dado que cualquiera que acceda a la clave pública b y a la firma $s_k(x)$, obtendría el texto original. En segundo lugar, existe la posibilidad de falsificar una firma sin necesidad de conocer la clave que permite su construcción. Para ello, basta observar que si x_1 y x_2 son dos mensajes de los que se conoce su firma, $s(x_1)$ y $s(x_2)$ respectivamente, entonces, el valor $s(x_1)s(x_2)$ sería la firma correspondiente al valor x_1x_2, con lo que una tercera parte podría crear mensajes que podrían ser autenticados como válidos sin que se conociese, tal y como se indica más arriba, el valor de la clave privada, a.

Por otro lado, también puede observarse que la firma es equivalente en tamaño al propio mensaje que se firma, lo que puede ser también un inconveniente.

Por todo lo anterior, se hace absolutamente necesario el contar con una herramienta adicional que permita alcanzar el objetivo para el que se define la firma digital, además de hacerlo de un modo más eficiente.

Así pues introducimos el concepto de función hash o función resumen en la siguiente sección.

3.2 Funciones Hash

Definición 3.2:[8] Una función Hash es una función de una vía $h : X \longrightarrow Y$, tal que dado $x \in X$, sea del tamaño que sea , proporciona $y = f(x)$ siempre del mismo tamaño.

La función h es libre de colisiones débiles si, dado $x \in X$, es computacionalmente imposible hallar un $x' \in X$, con $x \neq x'$ tal que $h(x) = h(x')$.

Nótese que esta propiedad tiene gran importancia, ya que, supongamos que tenemos un mensaje $x \in X$, y una función Hash $h : X \longrightarrow Y$. Si una tercera persona obtiene un $x' \neq x$ tal que $h(x') = h(x)$, podría cambiar un mensaje por otro, y el valor de la función Hash no cambiaría.

La función h es libre de colisiones fuertes si es computacionalmente imposible encontrar $x, x' \in X$ distintos tales que $h(x) = h(x')$.

En el caso de firmas digitales tenemos lo siguiente:

Sea $f : X \longrightarrow Y$ una función de una vía con puerta trasera y sea $h : Y \longrightarrow Y$ una función Hash. Entonces $f^{-1} \circ h$ reproduce el funcionamiento de una firma digital. Para verificar una firma digital, se calcula el valor hash del mensaje m, y se comprueba que $(f \circ f^{-1} \circ h)(m) = h(m)$.

Hemos de observar que el uso de una función hash, nos permite ahora usar el esquema de firma de RSA sin ningún problema. Si por cualquier causa, el valor $h(m)^a \ mod n$, firma del mensaje m mediante el esquema de firma RSA, cae en poder de un adversario, si éste usa la correspondiente clave pública, b, entonces $(h(m)^a)^b = h(m)$ y, dado que h es una función de una vía, será computacionalmente inviable encontrar el valor del mensaje m, con lo que se ha solucionado el problema de confidencialidad de dicha firma.

El siguiente lema es un ejemplo de una función Hash basada en el problema del logaritmo discreto, al igual que el criptosistema de ElGamal:

Lema 3.1: lema de Chaum-van Heijst-Pftizmann.[3]

Sean p, q dos primos tales que $p = 2q + 1$, sean α y β dos generadores de $\mathbb{Z}_p$. Identifiquemos $\mathbb{Z}_q$ con el conjunto $\{0, 1, ..., q-1\} \subseteq \mathbb{N}$. Se tiene entonces que la función $h : \mathbb{Z}_q \times \mathbb{Z}_q \to \mathbb{Z}_p^*$ dada por $h(x_1, x_2) \to \alpha^{x_1}\beta^{x_2}$ es una función Hash, en la que encontrar una colisión es equivalente a resolver el logaritmo discreto $log_\alpha(\beta)$.

Demostración:

Sea $s = log_\alpha(\beta)$. Se tiene entonces que $\alpha^{x_1}\beta^{x_2} = \alpha^{x_1}\alpha^{sx_2}$ y, por otro lado, $\alpha^{x_1+s}\beta^{x_2-1} = \alpha^{x_1}\alpha^{s}\alpha^{sx_2}\alpha^{-s} = \alpha^{x_1}\alpha^{sx_2}$. Por tanto,$(x_1, x_2)$ tiene la misma imagen que $(x_1 + s, x_2 - 1)$.

Recíprocamente, sean $(x_1, x_2) \neq (x_3, x_4)$, tales que $h(x_1, x_2) = h(x_3, x_4)$. De este modo, obtenemos que $\alpha^{x_1-x_3} = \beta^{x_4-x_2}$. Si $x_2 = x_4$, como hemos identificado $\mathbb{Z}_q$ con el conjunto $\{0, 1, ..., q-1\}$, $x_1 = x_3$. Por tanto, debe ser $x_1 \neq x_3$ y $x_2 \neq x_4$. Sin pérdida de generalidad, , suponemos $x_4 > x_2$. LLamamos $d = mcd(x_4 - x_2, p-1)$. Como $q > x_4 - x_2 \geq 1$ y $p - 1 = 2q$, obtenemos dos posibilidades, $d = 1$ o $d = 2$.

1.Si $d = 1$, para $y \equiv (x_4 - x_2)^{-1} \ mod(p-1)$, se tiene que $(\beta^{x_4-x_2})^y = \beta$. Además, $(\beta^{x_4-x_2})^y = (\alpha^{x_3-x_1})^y$, por tanto, $(x_3 - x_1)y = s$.

2. Si $d = 2$, entonces $mcd(x_4-x_2, q) = 1$, dado que $p-1 = 2q$. Sea $y \equiv (x_4-x_2)^{-1} \ mod(q)$. Se tiene entonces que $y(x_4 - x_2) = 1 + kq$, con $k \in \mathbb{Z}$. Por tanto, $\beta^{qk}\beta = \beta^{qk+1} = \beta^{y(x_4-x_2}$.

Por otro lado, $\beta^{qk} = \pm 1$, ya que $(\beta^q)^2 = \beta^{p-1} = 1$. En conclusión, $s \in \{(x_4 - x_2)y, (x_4 - x_2)y + q\}$.

■

En la práctica se ha venido sucediendo el uso de múltiples funciones hash a lo largo de la historia reciente de la Criptografía, dado que aunque las funciones hash son funciones de una vía, no dejan de ser aplicaciones $f : X \to Y$ donde el conjunto X tiene un cardinal mucho mayor que el del conjunto Y, con lo que la existencia de colisiones y, por tanto, la posibilidad de falsificar una firma en base a dichas propiedades es cuestión de tiempo tras el uso o recomendación de una de ellas. Actualmente las funciones hash recomendadas son las de la familia SHA3[15].

3.3 Firmas digitales basadas en el criptosistema de ElGamal

En este apartado, definiremos y daremos las principales características de la firma digital basada en el criptosistema de ElGamal, además daremos algunas variantes de dicho sistema de firmado.

A diferencia del apartado anterior, donde veíamos que se podía combinar la encriptación de clave pública y el sistema de firmado RSA, en este caso, el sistema de firmado de ElGamal está diseñado únicamente con el objetivo de realizar firmas.

El sistema de firmado de ElGamal es no determinista, es decir, hay muchas firmas válidas para un mismo mensaje, en este caso, la función de verificado debe reconocer la validez de todas ellas.

Sistema de firmado 3.2: sistema de firmado de ElGamal [6]

Sea p un primo tal que el problema del logaritmo discreto en $\mathbb{Z}_p$ (sección 2.3) sea irresoluble. Sea $\alpha \in \mathbb{Z}_p^*$ un elemento primitivo. Se define:

$$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \ mod(p)\}.$$

Para $K = (p, \alpha, a, \beta)$, y para un cierto número aleatorio secreto $k \in \mathbb{Z}_{p-1}^*$, se define:

$$s_K(x, k) = (\gamma, \delta), \text{ donde}$$
$$\gamma = \alpha^k \ mod(p), \text{ y}$$
$$\delta = (x - a\gamma)k^{-1} \ mod(p-1).$$

Para $x, \gamma \in \mathbb{Z}_p^*$ y $\delta \in \mathbb{Z}_{p-1}$, se define:

$$v_K(x, (\gamma, \delta)) = true \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \ mod(p)$$

Los valores p,α y β son la clave pública, a es la clave privada.

Veamos ahora que el sistema de firmado es válido, se tiene:

$$\beta^\gamma \cdot \gamma^\delta \equiv \beta^{\alpha^k} \alpha^{k(x-a\alpha^k)k^{-1}} \ mod(p) \equiv \alpha^{\alpha^k + x - a\alpha^k} \ mod(p) \equiv \alpha^x$$

Por tanto el sistema de firmado es válido.

Ejemplo 3.1: Sean $p = 467$, $\alpha = 2$, $a = 127$, se tiene:

$\beta = \alpha^a \ mod(p) = 2^{127} \ mod(467) = 132$

Imaginemos que Alice quiere enviar el mensaje x=100 a Bob, ella escoge el valor aleatorio $k = 213$, se tiene que $mcd(213, 466) = 1$ y $213^{-1} \ mod(466) = 431$. Entonces:

$$\gamma = 2^{213} \ mod(487) = 29$$
$$\delta = (100 - 127 \cdot 29) \cdot 431 \ mod(466) = 51$$

Se puede verificar la validez de la firma comprobando lo siguiente:

$$132^{29} \cdot 29^{51} \equiv 189 \ mod(467) \equiv 2^{100} \ mod(467)$$

Por tanto la firma es válida.

Notemos que hay tres formas de atacar este sistema de firmado:

La primera es, escogiendo el valor γ, intentar obtener el valor de δ, sin embargo, esto se encontraría ante el problema del logaritmo discreto (sección 2.3).

La segunda consistiría en escoger el valor de δ, y tratar de obtener el valor de γ mediante la ecuación:

$$\beta^\gamma \cdot \gamma^\delta \equiv \alpha^x \ mod(p)$$

Este problema no parece tener relación con ninguno de los problemas de la Criptografía estudiados anteriormente. Por ello, por ahora, la obtención de su solución es inviable.

La tercera forma consiste en intentar calcular los valores de γ y δ simultáneamente. Nunca se ha propuesto una posible solución de este problema, de hecho, no se sabe si su resolución es posible, aunque tampoco se ha probado que no lo sea.

3.4 Sistema de firmado Schnorr

Este sistema de firmado es una variante del sistema de firmado de ElGamal. En este caso, supongamos p y q son primos tales que $p-1 \equiv 0\ mod(q)$, usualmente, se suelen escoger valores similares a $p \approx 2^{1024}$ y $q \approx 2^{160}$. El sistema de firmado Schnorr, trata de modificar el sistema de firmado de ElGamal, haciendo que un mensaje de $\log_2 q$ bits, sea firmado usando una firma de $2\log_2 q$ bits, pero realizando los cálculos en $\mathbb{Z}_p$.

Este algoritmo, trabaja en un subgrupo de $\mathbb{Z}_p^*$ de orden q. La seguridad del algoritmo se basa en que en dicho subgrupo de $\mathbb{Z}_p^*$ sea inviable la resolución del problema del logaritmo discreto.

Sistema de firmado 3.3: sistema de firmado Schnorr [14]

Sea p un primo tal que la resolución del problema del logaritmo discreto en $\mathbb{Z}_p^*$ sea inviable. Sea q el un primo que divide a $p-1$. Sea $\alpha \in \mathbb{Z}_p^*$ una raíz q-ésima de 1 módulo p. Tomamos $P = \{0,1\}^*$, $A = \mathbb{Z}_q \times \mathbb{Z}_q$, y definimos:

$$K = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a\ mod(p)\},$$

con $0 \leq a \leq q-1$. Los valores p, q, α y β conforman la clave pública y el valor a es la clave privada. Definimos una función Hash $h : \{0,1\}^* \to \mathbb{Z}_q$.

Para un cierto $K = (p, q, \alpha, a, \beta)$ y para un número aleatorio secreto k, con $1 \leq k \leq q-1$, se define:

$$s_K(x, k) = (\gamma, \delta)$$

donde

$$\gamma = h(x, \alpha^k\ mod(p))$$
$$\delta = k + a\gamma\ mod(q).$$

Para $x \in \{0,1\}^*$ y $\gamma, \delta \in \mathbb{Z}_q$, se tiene:

$$v_K(x, (\gamma, \delta)) = true \iff h(x,\ \alpha^\delta \beta^{-\gamma}\ mod(p)) = \gamma$$

Pasemos ahora a verificar la validez del sistema de firmado Schnorr, se tiene:

$$h(x, \alpha^\delta \beta^{-\gamma}\ mod(p)) = h(x, \alpha^{k+\gamma a}\alpha^{-\gamma a}\ mod(p)) = \gamma$$

Por tanto el sistema de firmado es válido.

3.5 La firma DSS

En la sección 3.3, hemos visto el funcionamiento del sistema de firmado de ElGamal. El principal problema de esta sistema de firmado, es que para encriptar el mensaje, es necesario aumentar la longitud de bits, es decir, de un elemento en $\mathbb{Z}_p$, pasábamos a tener un elemento en $\mathbb{Z}_p \times \mathbb{Z}_p$. Esto provoca complicaciones computacionales, ya que, conforme aumentemos los bits del mensaje, aumentarán aún más los bits del mensaje encriptado. Por ello, es tan importante el siguiente sistema de firmado:

Sistema de firmado 3.4: sistema de firmado DSS [18]

Sea p un primo y q un primo divisor de p-1. Sea α un elemento de orden q en $\mathbb{Z}_p*$, es decir, $\alpha^q \equiv 1\ mod(p)$. Para un entero aleatorio $a \in \mathbb{Z}_p*$, se define:

$$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \ mod(p)\}$$

Sea $x \in \mathbb{Z}_p$ el mensaje que se desea firmar.Para un cierto K=(p, α, a, β),y para un k entero aleatorio, se definen las siguiente función de firmado:

$$s_K(x,k) = (\gamma, \delta) \ mod(p), \text{ donde}$$
$$\gamma = \alpha^k \ mod(p), \text{ y}$$
$$\delta = (x + a\gamma)k^{-1} \ mod(q)$$

La clave pública es (p, q, α, β) y la clave privada es a.

Sea (γ, δ) una firma, para el verificar la validez de la firma se deben realizar los siguientes pasos:

1. Verificar que $0 < \gamma < p$ y $0 < \delta < q$, si no se cumplem esta condición la firma no es válida.
2. Calcular $w = \delta^{-1} \ mod(q)$.
3. Calcular $u = x \cdot w \ mod(q)$.
4. Calcular $v = \gamma \cdot w \ mod(q)$.
5. Comprobar que $\gamma = (\alpha^u \beta^v \ mod(p))mod(q)$.

Pasemos a verificar la validez del sistema de firmado DSS, se tiene:

$$(\alpha^u \beta^v \ mod(p))mod(q) \equiv (\alpha^{xw} \beta^{\gamma w} \ mod(p))mod(q) \equiv (\alpha^{xw + a\gamma w} \ mod(p))mod(q) \equiv \alpha^{w(x+a\gamma)}$$
$$mod(p))mod(q) \equiv \alpha^{x+a\gamma)^{-1}k(x+a\gamma} \ mod(p))mod(q) \equiv \alpha^k$$

Por tanto el sistema de firmado es válido.

Ejemplo 3.2: Sean p=7, q=3 y $\alpha = 2$. Supongamos que $x = 6$.

Claramente se tiene que $q|p - 1$ y $\alpha^q \equiv 1 \ mod(p)$.

Se escoge número aleatorio $a \in \mathbb{Z}_p*$, por ejemplo tomemos $a = 5$. Se tiene entonces que $\beta \equiv \alpha^a \ mod(p) = 2^5 \ mod(7) \equiv 4 \ mod(7)$.

Se publica la 4-tupla $(p, q, \alpha\beta) = (7, 3, 2, 4)$, y se mantiene el valor de a secreto.

Se selecciona un número entero aleatorio k, por ejemplo $k = 2$.

Se calcula $\gamma = (\alpha^k mod(p)) \ mod(q) = (4 \ mod(7)) \ mod(3) = 1 \ mod(3) = 1$.

Se calcula $\delta = k^{-1}(x + a\gamma) \ mod(q) = 2(6 + 5 \cdot 1) \ mod(3) = 1$.

La firma sería el par $(1, 1)$

Veamos que la firma es válida. Recordemos que los valores (p, q, α, β) son públicos.

Se calcula $w = \delta^{-1} \ mod(q) = 1^{-1} \ mod(3) = 1$.

Se calcula $u = xw \ mod(q) = 6 \cdot 1 \ mod(3) = 0$.

Se calcula $v = \gamma w \ mod(q) = 1 \cdot 1 \ mod(3) = 1$.

Comrpobamos lo siguiente:

$$\gamma = (\alpha^u \beta^v \ mod(p))mod(q) \Longleftrightarrow 1 = (2^0 \cdot 4^1 mod(7)) \ mod(3) = (4mod(7)) \ mod(3) = 4$$
$$mod(3) = 1$$

Por tanto la firma es válida.

El principal problema de la firma DSS y de la firma Schnorr es la longitud de la clave requerida, ya que, con objeto de que sea inviable la resolución del problema del logaritmo discreto, se exige que la longitud del primo sobre el que se definen los cálculos es demasidado extensa. Es por ello, que en la actualidad se emplean otras firmas mucho más eficientes, por ejemplo la firma basada en curvas elípticas.

3.6 Firmas basadas en curvas elípticas

En el año 2000, se aprueba una mejora del sistema de firmado DSS, llamado sistema de firmado basado en curvas elípticas. Tenemos dos puntos A y B en una curva elíptica definida en $\mathbb{Z}_p$ para cierto p primo. El logaritmo discreto $m = log_A B$ es la clave privada. El orden de A es un número primo largo q.

Antes de hablar sobre este sistema de firmado, definiremos y daremos las principales características de las curvas elípticas.

Definición 3.3: curva elíptica.[16]

Sean $a, b \in \mathbb{R}$ dos constantes tales que $4a^3 + 27b^2 \neq 0$. Se define la curva elíptica sin singularidades como el conjunto de soluciones $(x, y) \in \mathbb{R} \times \mathbb{R}$ de la ecuación $y^2 = x^3 + ax + b$, junto con un punto $\mathbb{O}$ denominado punto en el infinito.

Definición 3.4: curva elíptica en un módulo primo.[16]

Sea $p > 3$ un primo. Se define la curva elíptica $E := y^2 = x^3 + ax + b$ sobre $\mathbb{Z}_p$ como el conjunto:

$$\{(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p : y^2 \equiv x^3 + ax + b \ mod(p)\}$$

donde $a, b \in \mathbb{Z}_p$ son constantes tales que $4a^3 + 27b^2 \not\equiv 0 \ mod(p)$, junto con un punto $\mathbb{O}$ denominado punto en el infinito.

Sean $P = (x_1, y_1)$ y $Q = (x_2, y_2)$ dos puntos de E, se define la operación suma como:

Si $x_1 = x_2$ e $y_2 = -y_1$, entonces $P + Q = \mathbb{O}$, en caso contrario, $P + Q = (x_3, y_3)$, donde:

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

donde se define:

$$\lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1} & si \quad P \neq Q \\ (3x_1^2 - a)(2y_1)^{-1} & si \quad P = Q \end{cases}$$

Por otro lado se tiene que $P + \mathbb{O} = \mathbb{O} + P = P$.

Si definimos la operación interna + como hemos visto anteriormente, se tiene que $(E, +)$ es un grupo abeliano.

Ahora sí, pasamos a explicar el funcionamiento del sistema de firmado basado en curvas elípticas.

En el sistenma de firmado DSS, el valor $\alpha^k \ mod(p)$ era reducido a modulo q, con el fin de obtener el valor de γ, que era la primera componente de la firma (γ, δ). En el sistema de firmado de curvas elípticas, a partir de ahora llamado ECDSS, el valor análogo es r, que es la x-coordenada de la curva elíptica en el punto kA, reducido a módulo q, para un cierto k aleatorio. Este valor r es la primera componente de la firma (r, s).

Por último, el valor s, se obtiene haciendo el mismo cálculo que hacíamos en el sistema de firmado DSS con γ, a, k y el mensaje x, a los valores r, m, k y el mensaje x.

Resumamos lo que hemos obtenido:

Sistema de firmado 3.5: sistema de firmado basado en curvas elípticas.[19]

Sea p un primo grande y sea E una curva elíptica definida sobre $\mathbb{Z}_p$. Sea A un punto de E con orden q. La clave secreta es un número aleatorio d, con $0 < d < q$, y la clave pública es el valor $Q = dA$ en E.

Sea M, con $0 < M < q$, el mensaje que se desea firmar. Se escoge un número aleatorio k, con $0 < k < q$, tal que $mcd(k, q) = 1$. A continuación se calcula $R = (x, y) = kA$ y $r = x \ mod(q)$. Si q divide a r, se calcula un nuevo r con un k distinto.

Una vez obtenido r, se calcula $s = (M - rd)k^{-1}$.

La firma obtenida de M, es el par (r, s).

Para verificar la firma, debemos realizar los siguientes pasos:

1. Se calcula $w = s^{-1} \ mod(q)$.
2. Se calcula $u = Mw \ mod(q)$.
3. Se calcula $v = rw \ mod(q)$.
4. Se calcula $R = uA - vQ$.
5. Se verifica que $MA = rQ + sR$

Pasemos a verificar la validez del sistema de firmado basado en curvas elípticas:

$$rQ + sR = rQ + suA - svQ = rQ + sMwA - srwQ \ mod(q) =$$
$$= rQ + sMs^{-1}A - srs^{-1}Q \ mod(q) = rQ + MA - rQ = MA$$

Por tanto el sistema de firmado es válido.

Ejemplo 3.3:

Se escoge un primo, por ejemplo, p=83.

Escogemos una curva elíptica que cumpla las condiciones de la definición 3.4, por ejemplo:

$$\{(x, y) \in Z_{83} x Z_{83} : y^2 = x^3 + 11x + 43\}$$

Escogemos un número de puntos, para facilitar los cálculos, se escoge un número de puntos primo, ya que así, cualquier punto será generador. En este caso, hemos escogido 73 puntos, tomamos como generador cualquier punto, por ejemplo A=(12,34), el cual tiene orden 73, por tanto, q=73.

Escogemos un número aleatorio d, con $0 < d < q$, por ejemplo, escogemos $d = 27$. A continuación se calcula:

$$Q = dA = 27(12, 34) = (51, 17) \ mod(73)$$

Supongamos ahora que deseamos firmar el mensaje M=51. Se escoge otro número aleatorio k, con $0 < k < q$, tal que $mcd(k, q) = 1$, como q es primo, nos vale cualquiera, por ejemplo $k = 7$. Se calcula:

$$R = kA = 7(12, 34) = (82, 23)$$

Luego $r = 82$. Como 7 es coprimo con 73, es válido el r seleccionado.

Se calcula $s = (M - rd)k^{-1} = (51 - 82 * 27)7^{-1} = 56 \ mod(73)$

Por tanto la firma obtenida es el par $(82, 56)$.

Veamos que es válida, se calcula:

$$w = s^{-1}\ mod(73)=56^{-1}\ mod(73) = 30$$
$$u = Mw = 51 * 30\ mod(73) = 70$$
$$v = 82 * 30\ mod(73) = 51$$
$$R = uA - vQ = 70 * (12,34) - 51 * (51,17) = (28,68) - (9,46) = (28,68) + (9,37) = (82,23)$$

Verificamos que $MA = rQ + sR$:

$$MA = 51 * (12,34) = (41,39)$$

$$rQ + sR = 82 * (51,17) + 56 * (82,23) = (29,10) + (51,17) = (41,39)$$

Por tanto la firma es válida.

Todos los cálculos se han realizado con el programa Cryptool, cuyo enlace de descarga podrán encontrar en [20].

4 Firmas digitales ciegas

En el capítulo anterior, veíamos una herramienta para poder verificar la autoría de cierta información, sin embargo, en todos los ejemplos de firmas digitales, veíamos como era necesario publicar la identidad del firmante. Pero, hay ocasiones en las que no queremos que se conozca nuestra identidad. Los dos ejemplos más claros de esto son el voto electrónico y el dinero digital.

En el caso del voto electrónico, se tiene que una persona desea enviar su voto anónimamente, pero la autoridad directora de la votación debe ser capaz de verificar que ese voto es válido, y que, por ejemplo, la misma persona no ha votado dos veces.

Así, surge la idea de crear un tipo de firmas digitales que permitan autenticar la autoría de cierta información, pero no permitan a nadie, aparte del firmante, conocer la identidad del éste. Estas firmas se conocen como firmas digitales ciegas.

En este capítulo, trataremos y explicaremos el funconamiento de estas firmas digitales, para ello, y pondremos dos ejemplos, uno del voto digital y otro del dinero digital, centrándonos más en este último, aunque el funcionamiento de ambos es análogo.

4.1 El dinero digital: propiedades

Cerca del comienzo del siglo XXI, fotocopiar dinero era posible, aunque un cuidadoso receptor pudiera discernir las diferencias entre la copia y el original. Las copias de información electrónica, sin embargo, son indistinguibles del original. Por lo tanto, alguien que tiene una moneda electrónica válida, podría hacer varias copias. Se necesita algún método para evitar este doble gasto. Una idea sería que un banco central tuviera copias de cada moneda y quien tiene cada una. Pero si las monedas son grabadas a medida que son gastadas, el anonimato está comprometido. Ocasionalmente las comunicaciones con un banco central podrían fallar temporalmente, así que es también deseable para la persona que recibe la moneda ser capaz de verificarla como legítima sin necesidad de contactar con el banco en cada transacción.

T. Okamoto y K. Ohta (Okamoto-Ohta) enumeran que las seis propiedades de dinero efectivo digital deberían tener: [10]

1. El dinero debe poder ser enviado de forma segura a través de redes de ordenadores.

2. El dinero no puede ser copiado ni reutilizado.

3. El emisor del dinero debe poder ser anónimo. Si el dinero ha sido utilizado de forma legítima, nadie, ni el receptor del dinero, ni el banco, puede identificar al emisor.

4. La transacción ha de poderse hacer *offline*, es decir, no es necesario ninguna comunicación con el banco central durante la transacción.

5. El dinero puede ser transferido a otros.

6. Una cantidad de dinero debe poder ser dividida en pequeñas cantidades.

Okamoto y Ohta idearon un sistema que satisfacía dichas propiedades, sin embargo, muchos de los sistemas satisfacen solo algunas de ellas.

Es evidente que un sistema de dinero digital es mucho más complicado que un sistema usual. Esto es debido a que los objetos electrónicos pueden ser falsificados sin ningún coste, al contrario que el dinero a papel.

Es aquí donde aparecen las firmas digitales, si asociamos una firma digital a nuestro dinero, será mucho más complicado falsificarlo, ya que, para ello, deberían atacar dicha firma, lo cual hemos visto en el anterior capítulo que no es nada sencillo.

El problema entonces que surge es mantener el anonimato. Como solución a estos problemas se propuso la idea de las firmas digitales ciegas.

4.2 Firmas digitales ciegas

Para explicar el funcionamiento de estas firmas digitales, explicaremos el funcionamiento de cada una de las partes: una entidad verificadora, un firmante y una entidad de confianza.[16]

Inicialización

Este proceso se realiza una única vez, y debe ser realizado por una autoridad central. Se escoge un p primo tal que $q = \frac{p-1}{2}$, también sea primo. Sea g el cuadrado de una raíz primitiva en módulo p. Esto implica que $g^{k_1} \equiv g^{k_2}\ mod(p) \Longleftrightarrow k_1 \equiv k_2\ mod(q)$ (1). Se escogen 2 exponentes secretos aleatorios, y se definen g_1 y g_2 como g elevado a cada uno de dichos exponentes en módulo p.

A continuación estos exponentes son eliminados, ya que no tienen ninguna utilidad, y si un hacker lograra obtenerlos, la seguridad del sistema se vería comprometida.

Los números g, g_1 y g_2 son públicos. Además, son escogidas dos funciones Hash, también públicas. La primera, denotada por H, tiene como entrada una 5-tupla de números enteros y devuelve un número entero en módulo q. La segunda, denotada por H_0, tiene como entrada una 4-tupla de enteros y devuelve un número entero en módulo q.

La entidad de confianza

La entidad de confianza escoge su número de identidad secreto x y calcula:

$$h \equiv g^x, h_1 \equiv g_1^x\ y\ h_2 \equiv g_2^x\ mod(p) \tag{4.1}$$

Los números h_1, h_2 y h son públicos e identifican a la entidad de confianza.

El firmante

El firmante escoge un número de identidad secreto u y calcula su número de cuenta:

$$I \equiv g_1^u\ mod(p) \tag{4.2}$$

El número I es enviado a la entidad de confianza, que almacena dicho número con toda la información personal del firmante (nombre, dirección, etc). Sin embargo, el firmante no envía el valor u a la entidad de confianza. A continuación, la entidad de confianza envía el valor z' al firmante, donde

$$z' \equiv (Ig_2)^x mod(p) \tag{4.3}$$

La entidad verificadora

La entidad verificadora escoge un número de identificación M y lo registra en la entidad de confianza.

Pasos previos al firmado

El firmante contacta con la entidad de confianza, pidiendo una información. La entidad de confianza requerirá una prueba de identificación, tal y como pasa cuando queremos retirar dinero de un cajero. Una firma es representada por una 6-tupla de números (A, B, z, a, b, r).

Esto puede parecer muy complicado, sin embargo, es necesario para poder asegurar las dos metas que exigíamos al principio del capítulo.

Los números se construyen de la siguiente forma:

1. La entidad de confianza escoge un número aleatorio w (es un número distinto para cada firma), y calcula:

$$g_w \equiv g^w \ y \ \beta = (Ig_2)^w \ mod(p) \tag{4.4}$$

y envía los valores g_w y β al firmante.

2. El firmante escoge una 5-tupla secreta y aleatoria de enteros $(s, x_1, x_2, \alpha_1, \alpha_2)$

3. El firmante calcula:

$$A \equiv (Ig_2)^s, \ B \equiv g_1^{x_1} g_2^{x_2}, \ z \equiv (z')^s, \ a \equiv g_w^{\alpha_1} g^{\alpha_2}, \ b \equiv \beta^{s\alpha_1} A^{\alpha_2} \ mod(p) \tag{4.5}$$

Las firmas con $A = 1$ no están permitidas. Esto ocurre solo de dos formas, o bien $s \equiv 0 \ mod(p)$, por ello se exige que $s \not\equiv 0 \ mod(p)$, o bien $Ig_2 \equiv 1 \ mod(p)$, lo que implicaría que el firmante ha resuelto el logaritmo discreto escogiendo un número al azar u.

4. El firmante calcula:

$$c \equiv \alpha_1^{-1} H(A, B, z, a, b) \ mod(q) \tag{4.6}$$

y envía el valor de c a la entidad de confianza.

5. La entidad de confianza calcula y envía el valor c_1 al firmante, donde:

$$c_1 \equiv cx + w \ mod(q) \tag{4.7}$$

6. El firmante calcula:

$$r \equiv \alpha_1 c_1 + \alpha_2 \ mod(q) \tag{4.8}$$

La firma es la 6-tupla (A, B, z, a, b, r).

Este procedimiento ha de ser repetido cada vez que el firmante quiera una firma. La entidad de confianza deberá escoger un nuevo número secreto aleatorio w y el firmante deberá escoger una nueva 5-tupla $(s, x_1, x_2, \alpha_1, \alpha_2)$.

Firmar

El firmante entrega la firma (A, B, z, a, b, r) a la entidad verificadora, y este sigue el siguiente procedimiento:

1. La entidad verificadora comprueba lo siguiente:

Teorema 4.1: sea (A, B, z, a, b, r) una firma digital. Sea p el número primo definido al principio de esta sección. Sea H la función Hash asociada a esta firma. Sea g el número público identificador de la entidad de confianza. Se tiene entonces que:

$$g^r \equiv ah^{H(A,B,z,a,b)} \text{ y } A^r \equiv z^{H(A,B,z,a,b)} b \ mod(p)$$

Demostración:

i)

$$\begin{gathered} g^r \equiv ah^{H(A,B,z,a,b)} \ mod(p) \\ \iff g^r \equiv g_w^{\alpha_1} g^{\alpha_2} h^{H(A,B,z,a,b)} \ mod(p) \iff g^r \equiv g^{w\alpha_1} G^{\alpha_2} h^{H(A,B,z,a,b)} \ mod(p) \\ \iff g^r \equiv g^{w\alpha_1+\alpha_2+xH(A,B,z,a,b)} \ mod(p) \iff r \equiv w\alpha_1 + \alpha_2 + xH(A,B,z,a,b) \ mod(q) \\ \iff \alpha_1 c_1 + \alpha_2 \equiv w\alpha_1 + \alpha_2 + xH(A,B,z,a,b) \ mod(q) \\ \iff \alpha_1 cx + \alpha_1 w + \alpha_2 \equiv w\alpha_1 + \alpha_2 + xH(A,B,z,a,b) \ mod(q) \\ \iff \alpha_1 \alpha_1^{-1} H(A,B,z,a,b) \equiv xH(A,B,z,a,b) \ mod(q) \iff 0 \equiv 0 \end{gathered}$$

ii)

$$\begin{gathered} A^r \equiv z^{H(A,B,,z,a,b)} b \ mod(p) \iff (Ig_2)^{sr} \equiv bz'^{sH(A,B,,z,a,b)} \ mod(p) \\ \iff (g_1^u g_2)^{sr} \equiv \beta^{sx_1} A^{\alpha_2} (Ig_2)^{xsH(A,B,,z,a,b)} \ mod(p) \\ \iff (g_1^u g_2)^{sr} \equiv (Ig_2)^{ws\alpha_1} (Ig_2)^{s\alpha_2} (Ig_2)^{xsH(A,B,,z,a,b)} \ mod(p) \\ \iff (g_1 g_2)^{sr} \equiv (g_1^u g_2)^{ws\alpha_1} (g_1^u g_2)^{s\alpha_2} (g_1^u g_2)^{xsH(A,B,,z,a,b)} \\ \iff (g^{uk_1+k_2})^{sr} \equiv (g^{uk_1+k_2})^{ws\alpha_1} (g^{uk_1+k_2})^{s\alpha_2} (g^{uk_1+k_2})^{xsH(A,B,z,a,b)} \ mod(p) \\ \iff (uk_1+k_2)sr \equiv (uk_1+k_2)ws\alpha_1 + (uk_1+k_2)s\alpha_2 + (uk_1+k_2)xsH(A,B,z,a,b) mod(q) \\ \iff sr \equiv ws\alpha_1 + s\alpha_2 + xsH(A,B,z,a,b) \ mod(q) \\ \iff s\alpha_1 c_1 + s\alpha_2 \equiv ws\alpha_1 + s\alpha_2 + xsH(A,B,z,a,b) \ mod(q) \\ \iff s\alpha_1 w + s\alpha_1 cx \equiv ws\alpha_1 + xsh(A,B,z,a,b) \ mod(q) \\ \iff ws\alpha_1 + xsh(A,B,z,a,b) \equiv ws\alpha_1 + xsh(A,B,z,a,b) \ mod(q) \iff 0 \equiv 0 \end{gathered}$$

■

Si esto se cumple, la firma es válida. Aún así, son necesarios más pasos para prevenir el el posible doble uso de la firma.

2. La entidad verificadora calcula:

$$d = H_0(A, B, M, t), \tag{4.9}$$

donde t representa la fecha y hora de la transacción.

A continuación, la entidad verificadora envía el valor d al firmante.

3. El firmante calcula:

$$r_1 \equiv dus + x_1 \text{ y } r_2 \equiv ds + x_2 \ mod(q)$$

donde u es el número secreto del firmante, y s, x_1 y x_2 son parte de la 5-tupla secreta y aleatoria escogida antes. El firmante envía los valores r_1 y r_2 a la entidad verificadora.

4. La entidad verificadora comprueba lo siguiente:

Teorema 4.2: sea (A, B, z, a, b, r) una firma digital. Sea p el número primo definido al principio de esta sección. Sean g_1 y g_2 los números públicos que identifican a la entidad de confianza. Sean r_1 y r_2 los valores definidos en el punto anterior y $d = H_0(A, B, M, t)$ el valor de la función Hash definida anteriormente. Se tiene:

$$g_1^{r_1} g_2^{r_2} \equiv A^d B \ mod(p)$$

Demostración:

$$\begin{gathered} g_1^{r_1} g_2^{r_2} \equiv A^d B \ mod(p) \Longleftrightarrow g_1^{r_1} g_2^{r_2} \equiv (g_1^u g_2)^{sd} g_1^{x_1} g_2^{x_2} \ mod(p) \\ \Longleftrightarrow g^{k_1 r_1 + k_x r_2} \equiv g^{(k_1 u + k_2) sd + k_1 x + k_2 x_2} \ mod(p) \\ \Longleftrightarrow k_1 r_1 + k_2 r_2 \equiv (k_1 u + k_2) sd + k_1 x_1 + k_2 x_2 \ mod(q) \\ \Longleftrightarrow k_1 dus + k_1 x_1 + k_2 ds + k_2 x_2 \equiv k_1 dus + k_2 ds + k_1 x_1 + k_2 x_2 \ mod(q) \\ \Longleftrightarrow 0 \equiv 0 \end{gathered}$$

■

Si esto se cumple, la entidad verificadora acepta la firma, si no, la rechaza.

Verificación de la firma

Ahora, la entidad verificadora quiere guardar la firma en la entidad de confianza. La entidad verificadora envía la firma (A, B, z, a, b, r) y la 3-tupla (r_1, r_2, d). La entidad de confianza comprueba lo siguiente:

1. La firma (A, B, z, a, b, r) nunca ha sido depositada en la entidad de confianza. Si nunca ha sido depositada, se pasa al siguiente paso. Sin embargo, si si ha sido depositada previamente, la entidad de confianza salta al procedimiento de control de fraude, explicado más adelante.

2. La entidad de confianza comprueba lo siguiente:

$$g^r \equiv a h^{H(A,B,z,a,b)},\ A^r \equiv z^{H(A,B,z,a,b)} b \text{ y } g_1^{r_1} g_2^{r_2} \equiv A^d B \ mod(p)$$

Como podemos observar son las mismas equivalencias de los teoremas 4.1 y 4.2, por lo que su demostración no es necesaria.

Si las equivalencias son ciertas, la entidad de confianza acepta la firma, y ésta es depositada.

Control de fraude

Hay muchas formas posibles de intentar realizar un fraude. A continuación veremos como podemos combatir el fraude:

1. El firmante intenta usar la firma dos veces, una vez con la entidad verificadora y otra vez con alguien al que llamaremos vendedor. La entidad verificadora envía la firma junto a la 3-tupla (r_1, r_2, d). El vendedor envía la firma junto a la 3-tupla $(r_1'.r_2', d')$. Con un simple cálculo se tiene lo siguiente:

$$r_1 - r_1' \equiv us(d - d') \text{ y } r_2 - r_2' \equiv s(d - d') \ mod(q)$$

Por lo tanto, $u \equiv (r_1 - r_1')(r_2 - r_2')^{-1}\ mod(q)$. La entidad de confianza calcula $I \equiv g_1^u\ mod(p)$ e identifica al firmante. Hasta que la entidad de confianza no puede calcular u de otra forma, tiene una prueba, o al menos una duda razonable, de que se ha usado dos veces la misma firma. El firmante deberá ser denunciado si se determina que la resolución del logaritmo discreto asociado al sistema era inviable.

2. La entidad verificadora intenta depositar la firma dos veces. Una con la firma válida (r_1, r_2, d) y otra con una falsificación (r_1', r_2', d').

Esto es imposible para la entidad verificadora, ya que es realmente complicado para la entidad verificadora generar números que cumpla esta igualdad:

$$g_1^{r_1'} g_2^{r_2'} \equiv A^{d'} B\ mod(p)$$

3. Alguien intenta crear una firma falsa. Esto requiere encontrar números que cumplan que $g^r \equiv ah^{H(A,B,z,a,b)}$ y $A^r \equiv z^{H(A,B,z,a,b)} b\ mod(p)$, lo cual es realmente complejo de hacer.

Por ejemplo, si la persona tuviera los valores de A, B, z, a, b, intentar encontrar el valor de r requiere resolver el problema del logaritmo discreto para que se cumpla la primera ecuación.

4. Imaginémonos ahora que una persona, a la que llamaremos Kevin, recibe una firma. Kevin intenta ingresar la firma en la entidad de confianza pero también intenta gastarla con una entidad verificadora.

Kevin entrega la firma a la entidad verificadora , la cual recibe un valor d', que, si bien es parecido a d, no es exactamente igual.

Kevin no conoce los valores u, x_1, x_2, s, pero debe escoger los valores r_1' y r_2' tales que $g_1^{r_1'} g_2^{r_2'} \equiv A^{d'} B\ mod(p)$. Lo cual nos lleva otra vez a que Kevin debería resolver un problema de logaritmo discreto.

Kevin no puede utilizar los valores r_1 y r_2 que conoce, puesto que, como $d \neq d'$, la entidad verificadora podría comprobar que $g_1^{r_1} g_2^{r_2} \not\equiv A^{d'} B$.

5. Un trabajador de la entidad de confianza intenta crear una firma. Esta persona tiene la misma información que tenía Kevin en el ejemplo anterior, además del número de identificación I. Es posible crear una firma que satisfaga la ecuación $g^r \equiv ah^{H(A,B,z,a,b)}$. Sin embargo, como el el número u es secreto, el trabajador de la entidad de confianza no puede calcular un valor de r_1 válido.

Si $s = 0$ fuera un valor permitido, este fraude si sería posible, ésta es una de las razones por las que $A = 1$ es un valor no está permitido.

6. Alguien roba la firma al firmante e intenta gastarla. La primera ecuación si se cumple, sin embargo, el ladrón no conoce el valor u, por lo tanto no puede calcular dos valores r_1 y r_2 tales que $g_1^{r_1} g_2^{r_2} \equiv A^d B$.

7. Imaginémonos que Kevin roba la firma y los valores (r_1, r_2, d) a la entidad verificadora antes de que sean enviados al banco. A no ser que la entidad de confianza exija a los entidades verificadoras guardar la hora y fecha de cada transacción, el robo habrá sido satisfactorio. Este ejemplo, es un problema que también tiene el dinero en efectivo.

El anonimato

Hemos visto que en ningún momento ni la entidad de confianza ni la entidad verificadora ha obtenido el número secreto de identificación del firmante u. Sin embargo,

el anonimato puede ser garantizado si y solo si, solo podamos crear una firma en cada transacción, es decir, los cálculos de antes han de repetirse en cada transacción.

Ejemplos

En este apartado veremos dos ejemplos de una firma digital a ciegas.

Ejemplo 4.1: dinero digital

En este caso tenemos un ejemplo de dinero digital, es decir, el firmante será un emisor, la entidad de confianza será un banco y la entidad verificadora será un receptor.

Es importante recalcar que se trata de un ejemplo inseguro, puesto que las funciones H y H_0 no son funciones Hash, debido a que trabajar con éstas es bastante complicado. Además, los números p y q son de un tamaño ínfimo, puesto que si no los cálculos son inviables.

Una autoridad central escoge el número $p = 23$, por tanto $q = \frac{23-1}{2} = 11$, que también es primo, por tanto nos encontramos ante un valor de p válido. Se escoge una raíz primitiva de $\mathbb{Z}_{23}$, por ejemplo $g = 2$. Se escogen dos exponentes aleatorios $k_1 = 3$ y $k_2 = 6$, y se calcula $g_1 = 2^3 \ mod(23) = 8$ y $g_2 = 2^6 \ mod(23) = 18$.

A continuación, se eliminan los valores k_1 y k_2. Se publican los valores g, g_1 y g_2. Se escogen las dos funciones Hash (recordamos que en nuestro ejemplo no son funciones Hash), sean estas $H : \mathbb{Z}^5 \longrightarrow \mathbb{Z}_{11}$ dada por $H(a,b,c,d,e) = a + b + c + d + 2e \ mod(11)$ y $H_0 : \mathbb{Z}^4 \longrightarrow \mathbb{Z}_{11}$ dada por $H_0(a,b,c,d) = a + 2b + 3c + d \ mod(11)$.

La entidad de confianza escoge ahora un número secreto $x = 7$, y calcula $h \equiv 2^7 \ mod(23) \equiv 13 \ mod(23)$, $h_1 \equiv 8^7 \ mod(23) \equiv 12 \ mod(23)$ y $h_2 \equiv 18^7 \ mod(23) \equiv 6 \ mod(23)$.

El firmante escoge su número secreto $u = 8$, y calcula $I \equiv 8^8 \ mod(23) \equiv 4$. Envía el valor de I al banco, pero no el de u.

La entidad de confianza calcula y envía al firmante el valor de $z' \equiv (Ig_2)^x \ mod(p) = (4 \cdot 18)^7 \equiv 2 mod(23)$.

A continuación, la entidad verificadora escoge un número de identificación $M = 3$, y lo registra en la entidad de confianza.

Ahora se pasa a la parte de crear una moneda, la entidad de confianza escoge un número aleatorio $w = 9$, y calcula:

$$g_w \equiv g^w \ mod(p) = 2^9 \ mod(23) \equiv 6 \ mod(23) \text{ y}$$
$$\beta \equiv (Ig_2)^w \ mod(p) = (4 \cdot 18)^9 \ mod(23) \equiv 18 \ mod(23),$$

y le envía estos valores al firmante.

El firmante escoge una 5-tupla secreta y aleatoria de enteros $(s, x_1, x_2, \alpha_1, \alpha_2) = (3, 5, 9, 8, 2)$, y calcula:

$$A \equiv (Ig_2)^s \ \text{mód}\ (p) = (4 \cdot 18)^3 \ \text{mód}\ (23) \equiv 4 mod(23).$$
$$B \equiv g_1^{x_1} g_2^{x_2} = 8^5 \cdot 18^9 \ mod(23) \equiv 8 \ mod(23).$$
$$z = (z')^s \ \text{mód}\ (p) = 2^3 \ mod(23) \equiv 8 \ mod(23).$$
$$a = g_w^{\alpha_1} g^{\alpha_2} \ mod(p) = 6^8 \cdot 2^2 \ mod(23) \equiv 3 \ mod(23).$$
$$b \equiv \beta^{s\alpha_1} A^{\alpha_2} \ mod(p) = 18^{3 \cdot 8} \cdot 4^2 \ mod(23) \equiv 9 \ mod(23).$$

El firmante calcula también:

$$c \equiv \alpha_1^{-1} H(A, B, z, a, b) \ mod(q) = 8^{-1} H(4, 8, 9, 3, 9) \ mod(11) \equiv$$
$$7 \cdot (4 + 8 + 9 + 3 + 18) \ mod(11) \equiv 8 \ mod(11),$$

y le envía el valor al banco.

La entidad de confianza calcula $c_1 + w \ mod(q) = 8 \cdot 7 + 9 \ mod(11) \equiv 10 \ mod(11)$, y se lo envía al firmante.

Por último, el firmante calcula $r \equiv \alpha c_1 + \alpha_2 \ mod(q) = 8 \cdot 10 + 2 \ mod(11) \equiv 5 \ mod(11)$.

Por tanto, la firma obtenida es $(A, B, z, a, b, r) = (4, 8, 9, 3, 9, 5)$.

Pasamos ahora a la última parte: gastar una moneda.

El firmante entrega la moneda obtenida a la entidad verificadora, y la entidad verificadora comprueba el Teorema 4.1:

$$\begin{gathered} \text{i) } 9 \ mod(23) \equiv 2^5 \ mod(23) = g^r \ mod(p) \equiv a h^{H(A,B,z,a,b)} mod(p) = \\ 3 \cdot 13^{H(4,8,9,3,9)} \ mod(23) \equiv 3 \cdot 13^4 2 \ mod(23) \equiv 9 \ mod(23) \\ \text{ii) } 12 \ mod(23) \equiv 4^5 \ mod(23) = A^r \ mod(p) \equiv z^{H(A,B,z,a,b)} b \\ mod(p) \equiv 8^4 2 \cdot 9 \ mod(23) \equiv 12 \ mod(23). \end{gathered}$$

Como se cumplen ambas equivalencias, la moneda es válida, sin embargo, hacen falta más pasos para prevenir el doble gasto.

Se escoge ahora un número t, el cual representa la hora y fecha de la transacción, por ejemplo, imaginemos que es 3 de junio del año 2021, se tiene entonces que $t = 3 + 6 + 21 = 30$, nótese que t puede ser cuaquier número, en este caso hemos escogido este, pero podría funcionar con cualquiera.

La entidad verificadora calcula $d = H_0(A, B, M, t) = H_0(4, 8, 3, 30) = 4 + 2 \cdot 8 + 3 \cdot 3 + 30 \ mod(11) = 59 \ mod(11) = 4$, y le envía el valor al firmante. El firmante calcula:

$$\begin{gathered} r_1 \equiv dus + x_1 \ mod(q) = 59 \cdot 8 \cdot 3 + 5 \ mod(11) \equiv 2 \ mod(11). \\ r_2 \equiv ds + x_2 \ mod(q) = 59 \cdot 3 + 9 \ mod(11) \equiv 10 \ mod(11). \end{gathered}$$

El firmante envía los dos valores anteriores a la entidad verificadora, y por último, la entidad verificadora comrpueba el teorema 4.2:

$$\begin{gathered} 1 \ mod(23) \equiv 8^2 \cdot 18^1 0 \ mod(23) \equiv g_1^{r_1} g_2^{r_2} \ mod(p) \equiv A^d B \ mod(p) \\ \equiv 4^4 \cdot 8 \ mod(23) \equiv 1 \ mod(23). \end{gathered}$$

Por tanto la entidad verificadora acepta la moneda.

Para finalizar, si la entidad verificadora quiere depositar la moneda en la entidad de confianza, la entidad de confianza debe comprobar los teoremas 4.1 y 4.2, además de comprobar que la moneda nunca había sido utilizada. Si esto es así, la moneda se deposita en la entidad de confianza. Como ya hemos comprobado dichas hipótesis antes, la moneda se deposita en el banco.

Ejemplo 4.2: voto electrónico

Otra aplicación de las firmas digitales a ciegas es el voto electrónico. Cuando nosotros votamos digitalmente debemos tener asegurado el anonimato, además solo podemos votar una única vez. En este caso el firmante sería la persona que vota, la entidad de confianza sería la junta electoral y la entidad verificadora sería la mesa electoral.

5 Criptomonedas: Blockchain y Bitcoin

En este capítulo, explicaremos la base de uno de los temas más populares de los últimos 10 años: el Bitcoin y las criptomonedas. Pese a que todos hemos oído hablar de ellas, realmente el conocimiento acerca de su funcionamiento, su seguridad y sus propiedades es bastante limitado. En este capítulo comentaremos las principales características del Bitcoin y de las criptomonedas, como son el Blockchain, la minería, etc.

5.1 Las criptomonedas

Empecemos desde lo más básico, ¿qué es una criptomoneda?, esta palabra se divide en dos, por un lado cripto, haciendo referencia a la seguridad de esta, mediante algunos de los mecanismos que hemos visto anteriormente, y por otro lado, moneda, lo cual es un objeto (puede ser virtual), que cumple las siguientes propiedades: [7]

1. Almacena un determinado valor. Es importante remarcar que el valor no tiene por qué ser monetario, por ejemplo, si tenemos monedas de un parque de atracciones, estas no tienen valor real, puesto que, en teoría, no pueden ser intercambiadas por dinero.

2. Es un medio de intercambio. La moneda tiene que poder intercambiarse con cierta facilidad, es decir, la moneda tiene que tener un mercado financiero.

3. Tiene que ser una unidad de medida, es decir, tener un valor fijo (no tiene por qué ser monetario).

A día de hoy, no hay prácticamente ninguna moneda que cumpla estas tres condiciones, ya que la condición 3 restringe mucho las posibilidades, en el caso del Euro por ejemplo, su valor va cambiando cada día, por tanto no sirve como unidad de medida.

Las criptomonedas se basan en las propiedades 1 y 2, ya que, al igual que en el caso de las monedas normales, su valor no es fijo.

Pasemos ahora a explicar la diferencia entre moneda usual, moneda digital y criptomoneda:

La moneda usual es la que todos conocemos, no tiene por qué tener un formato físico, es decir, tu cuenta bancaria sigue estando formada por monedas usuales, pese a estar en un medio digital.

La moneda digital es prácticamente lo mismo que la usual, sin embargo, como veíamos en el capítulo 4, en este caso se garantiza el anonimato total de las personas implicadas en cada transacción, pero se mantiene la idea de una entidad que controle todas las transacciones, usualmente, un banco.

La criptomoneda elimina el concepto de entidad de autoridad, es decir, no hay ninguna entidad por encima de los usuarios que intervienen en las transacciones.

Pese a que no haya una entidad, las transacciones han de ser verificadas, el problema del doble gasto ha de ser evitado, es ahí donde aparece el concepto de mineros y blockchain.

5.2 Blockchain o cadenas de bloques

El Blockchain o cadena de bloques [7] es la tecnología que permite el intercambio de criptomonedas. Imaginémonos que un sujeto A, quiere enviar cierta información a B (usualmente dinero). Esta transacción se representa en forma de bloque, este bloque se comparte a una red de usuarios. Los usuarios de la red verifican que la transacción es correcta, es decir, el sujeto A dispone de dicha información, y el usuario B puede recibirla, pero esto se realiza de forma totalmente anónima, es decir, ningún usuario de la red sabe quién es el sujeto A ni el sujeto B. Una vez verificada la información, el bloque se añade a la cadena de bloques de esa red, la cual está formada por todos los bloques que se han hecho en esa red, es decir, todas las transacciones que se han realizado a lo largo de los años en esa red. Esta cadena de bloques es similar a lo que conocemos como libro de cuentas, en el cual se anotan todas las transacciones llevadas a cabo para que quede constancia de éstas. Una vez este bloque se añade a la cadena, la información pasa al usuario B.

Podemos notar que esta tecnología tiene muchas ventajas frente a otro tipo de mercados:

1. Al quedar constancia de todas las transacciones de la red, es prácticamente imposible que la red tenga fallos de seguridad relacionados con el doble gasto, ya que, si la red de bloques contiene todas las transacciones, es imposible que un sujeto A intente utilizar de nuevo cierta información.

2. El anonimato está garantizado. Esto ocurre gracias a que en ningún momento ni el sujeto A ni el sujeto B han tenido que dar ningún dato personal, así como ninguno de los usuarios de la red.

3. No hay una entidad que controle la información. La gran diferencia que los otros dos tipos de monedas mencionados anteriormente es esta, no existe una entidad que regule las transacciones, sino un conjunto de usuarios que únicamente verifican la transacción.

5.3 El Bitcoin

La criptomoneda más famosa es sin duda el Bitcoin, de la que todos hemos oído hablar. Su inventor fue Satoshi Nakamoto, nombre que es un pseudónimo de alguien cuya identidad es deconocida. La primera constancia de Bitcoin que se tiene fue en el artículo [9], en el cual nos hablaba de un nuevo tipo de moneda.

Es cierto que al Bitcoin siempre ha estado relacionado con negocios oscuros, llegándose a conocer como la moneda de la Deep Web. Esta afirmación, pese a ser cierta, no le ha hecho justicia al Bitcoin.

El anonimato y la ausencia de una entidad de confianza ha provocado que el Bitcoin haya sido usado en muchas ocasiones en transacciones fraudulentas: compra de armas, de drogas, etc. Pero tal vez sea este el precio a pagar por la existencia de una moneda que no se somete a la regulación de bancos centrales. Sin ánimo de entrar en este debate, esta es la principal desventaja del Bitcoin.

El valor del Bitcoin

El valor del Bitcoin es una de las características más llamativas de esta criptomoneda. Las primeras transacciones con Bitcoin de las que se tienen constancia se realizaron en el año 2009, sin embargo, en ese momento carecía de valor, de hecho, solo era empleado entre aficionados a la Criptografía. En el año 2010, adquiere por primera vez un valor de mercado, aproximadamente unos 0.003 $.[4]

A lo largo de los años fue aumentando su valor rápidamente, hasta que en Noviembre del año 2013, ascendió desde 350$ hasta los 1300$. A lo largo de los años ha ido elevando más y más su valor, hasta que este año, alcanzó su máximo histórico, con un valor de mercado de unos 62000$. [4]

Es importante remarcar la especulación que hay alrededor del Bitcoin, sin embargo, esta subida exponencial de valor tiene una justificación, y es lo que diferencia Bitcoin (y algunas criptomonedas) del resto de monedas: el número de Bitcoin es limitado.

Esto no quiere decir que no se sigan creando, de hecho cada año se crea un número determinado de Bitcoins, pero la clave es esa, el número es determinado, da igual el valor que tenga el Bitcoin, da igual el estado de la economía mundial, el número de Bitcoins creado es fijo, y sigue un mismo algoritmo desde su creación. Este algoritmo produce la mitad de Bitcoins cada año, es decir, su creación se va reduciendo cada vez más, y, alrededor del año 2130, su producción se detendrá completamente.

Esto hace que el Bitcoin tenga una oferta limitada, y, al ser la demanda tan abrumadora, su valor se dispara.

5.4 Funcionamiento del Bitcoin

En esta sección, explicaremos el funcionamiento del Bitcoin. El Bitcoin se basa en tres conceptos clave: funciones Hash (de las cuales hablamos en la sección 3.2), firmas digitales (de las cuales hablamos en los capítulos 3 y 4) y árboles de Merkle.

Funciones hash

Como veíamos en la sección 3.2, las funciones Hash son funciones de una vía que deben cumpliar varias propiedades. En la sección 3.2 dimos una definición general de las funciones Hash, sin embargo, cuando lo introducimos en el mundo de las criptomonedas, es necesario que las funciones Hash sean computacionalmente eficientes, es decir, el ordenador debe ser capaz de crear la función Hash en un tiempo prudencial. Es importante recalcar que cada transacción de Bitcoins tiene su propia función Hash. [1]

Las funciones Hash más empleadas son las denominadas funciones SHA (Secure Hash Algorithm), y de todas ellas, la más común es la función SHA-256, debido a que fue la propuesta por Satoshi Nakamoto en [9].

Árbol de Merkle

Las funciones Hash se encargan de cifrar la información de la transacción que queremos realizar, sin embargo, la información cifrada debe tener un orden.

Los árboles de Merkle intentan estructurar la información en forma de árbol invertido, es decir, intenta establecer un orden jerárquico a través de funciones Hash, partiendo de N bloques de información, e intentando reducir lo máximo posible dichos bloques, si es posible, se reduce a un bloque, denominado raíz de Merkle. [8]

La raíz de Merkle contiene toda la información de las demás ramas, es decir, conociendo únicamente la raíz, hemos almacenado toda la información desde el principio y, por tanto, las otras ramas de información pueden ser borradas.

A medida que se van produciendo transacciones en el mismo árbol de Merkle, su tamaño va aumentando. El árbol de Merkle se recorre de izquierda a derecha y de abajo a arriba. Se van haciendo parejas, por ejemplo, si el árbol tiene 7 transacciones, llamadas, A,B,C,D,E,F Y G, iremos juntando estas por parejas mediante funciones Hash, es decir, A con B, C con D y E con F, obteniendo así 3 hashes distintos, llamados AB, CD y EF. Volveríamos a aplicar esto, juntando AB con CD y EF con G mediante funciones Hash, obteniendo así ABCD y EFG. Por último, juntamos estos dos bloques, obteniendo el hash final.

Si se quiere añadir una nueva transacción H, se añade abajo a la derecha, así, los hashes ABCD y EF permanecen intactos, y solo deberíamos calculas los hashes en los q se vean involucrados la transacción H.

La principal utilidad de esto es reducir en gran medida la cantidad de información almacenada en la red de bloques. Pese a esto, las redes de bloques no pueden ser infinitas, por ello, una vez la red llega a su máxima capacidad, la red se clausura, y no se pueden realizar más transacciones a través de ésta.

Estructura de una transacción con Bitcoins

Por último, veamos como es la estructura de una transacción con Bitcoins. Imaginemos que un sujeto A quiere enviarle cierta cantidad de Bitcoins a un sujeto B, se siguen los siguientes pasos:[11]

1. El sujeto A quiere realizar una transacción, dicha transacción es firmada por éste, y es enviada a la red.

2. Los usuarios verificadores, también llamados mineros (los cuales se explican en el siguiente apartado), comprueban dicha firma, utilizando la función de verificado.

3. Si la firma es válida, los mineros anotan la transacción, es decir, añaden dicha transacción en la cadena de bloques (abajo a la derecha en el árbol de Merkle). Mediante funciones Hash obtienen la raíz de Merkle, y pasan al siguiente paso.

4. Los mineros calculan un número aleatorio P, llamado *nonce*. Además, anotan un número F que puede ser la fecha o la posición de esta transacción en la cadena de bloques, con el fin de poder localizarla facilmente. Por último, se emplea una función Hash en la que se introducen la raíz de Merkle, el número F y el número P.

5. Si el Hash obtenido es válido, es decir, si cumple los parámetros estipulados por la red, la transacción es válida, y el Bitcoin llega al sujeto B. Además, el bloque de la transacción queda siempre en la cadena de bloques, por tanto, queda constancia de dicha transacción.

Esta secuencia de pasos puede parecer sencilla, sin embargo, el coste computacional para hallar un *nonce* para el cual el Hash sea válido es muy elevado. Debemos tener en cuenta, que si se cambia un único número en la entrada de una función Hash, el Hash obtenido será totalmente distinto. Además, este Hash obtenido debe ser de una forma

concreta: ha de tener una longitud de 256 bits y ser menor que un determinado valor establecido por la propia red.

Usuarios verificadores o mineros

Como hemos visto antes, la red está compuesta por una serie de usuarios verificadores de información. Es importante recalcar la utilidad de estos, pues si no existiesen, no existiría ningún tipo de control en las redes de Blockchain.

El principal cálculo que tienen que realizar los mineros es obtener el número aleatorio P para que el hash sea válido, lo cual es computacionalmente costoso. Por ello son requeridos ordenadores muy potentes, con el fin de generar una gran cantidad de números aleatorios.[7]

Estos usuarios son conocidos como mineros y, a cambio de cada transacción verificada, obtienen una comisión de ésta. Por ello, cada vez son necesarios ordenadores más potentes, con el fin de obtener el número P más rápido, y poder validar más transacciones en un tiempo menor.

Nota: La tecnología Blockchain es esencialmente la misma que la del Bitcoin, es decir, en lugar de anotar transacciones comerciales, podemos hacer que cualquier transmisión de información a través de la red quede anotada en la cadena de bloques del mismo modo, tras verificar la firma correspondiente, al igual que ocurre con la transacción comercial llevada a cabo con bitcoins. La tecnología del Blockchain es aplicable a numerosos campos, ya que permite transferir cualquier tipo de información anónimamente, y dejando constancia de dicha transferencia de información en la cadena de bloques.

Bibliografía

[1] P. Caballero Gil, *Introducción a la Criptografía*, Ra-Ma, 2002.

[2] D. Chaum, *Blind Signatures for Untraceable Payments*, In: Chaum D., Rivest R.L., Sherman A.T. (eds) Advances in Cryptology, Springer, Boston, MA, 1983, 199-205.

[3] D. Chaum, V. Heijst, B. Pfitzmann, *Cryptographically Strong Undeniable Signatures*, In: Feigenbaum J. (Ed.): Advances in Cryptology-CRYPTO '91, LNCS 576, Springer-Verlag 1992, 470-484.

[4] U.W. Chohan, *A History of Bitcoin*, SSRN, 2017.

[5] W. Diffie, M.E. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, **22** (1976), 644-655.

[6] T. ElGamal, *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, **31** (1985), 469-472.

[7] A. Fauster Sabater, L. Hernández Encinas, F. Montoya Vitini, J. Muñoz Masque, A. Martín Muñoz, *Criptografía, protección de datos y aplicaciones. Una guía para estudiantes y profesionales*, Ra-Ma, 2012.

[8] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[9] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, Cryptography Mailing List, 2008.

[10] T. Okamoto, K. Ohta, *Universal Electronic Cash*, In: Feigenbaum J. (Ed.): Advances in Cryptology-CRYPTO '91, LNCS 576, Springer-Verlag 1992, 324-337.

[11] J. Pastor Franco, M. A. Sarasa López, *Criptografía digital: fundamentos y aplicaciones*, Prensas Universitarias de Zaragoza, 1998.

[12] M.O. Rabin, *Digital signatures and public-key functions as intractable as factorization*, Technical Report MIT/LCS/TR—212, 1978..

[13] R.L. Rivest, A. Shamir, and L. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*,Communications of the ACM,**21**(1978),120-126.

[14] C.P. Schnorr, *Efficient Signature Generation by Smart Cards*, J. Cryptology, (1991), 161-174.

[15] N.P. Smart, *Cryptography Made Simple*, Springer, 2016.

[16] D.R. Stinson, *Cryptography Theory and Practice*, Chapman & Hall/CRC, 2006.

[17] W.Trappe, L.C. Washington,*Introduction to Cryptography with Coding Theory*, Pearson Prentice Hall, 2006.

[18] FIPS 186-2, *Digital signature standart (DSS)*, Federal Information Processing Standarts Publication 186, U. S. Dept. of Commerce/National Institute of Standards and Technology, 2000.

[19] ANSI X9.62, *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, American National Standards Institute, 1999.

[20] Página web de descarga del programa Cryptool: https://www.cryptool.org/en/