

UNPSJB

LICENCIATURA EN SISTEMAS OPGCPI

ADMINISTRACIÓN DE REDES Y SEGURIDAD

---

# Bitcoin

**Qué elementos de criptografía utiliza la primer criptomoneda**

---

Cátedra

Lic. Bruno Damián Zappellini

Integrantes:

Luciano Serruya Aloisi

14 de diciembre de 2018



# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Qué es Bitcoin . . . . .	2
1.2. Criptografía . . . . .	3
1.2.1. Simétrica . . . . .	3
1.2.2. Asimétrica . . . . .	3
1.3. Funciones de <i>hash</i> . . . . .	5
1.3.1. <i>SHA256</i> . . . . .	5
1.3.2. <i>RIPEMD160</i> . . . . .	6
1.4. Codificaciones . . . . .	6
1.4.1. <i>Base58Check</i> . . . . .	6
<b>2. Transacciones</b>	<b>6</b>
2.1. Ciclo de vida . . . . .	7
2.2. Verificación de una transacción . . . . .	7
<b>3. <i>Blockchain</i></b>	<b>8</b>
3.1. Estructura de un bloque . . . . .	8
<b>4. Minería de bloques</b>	<b>8</b>
4.1. Prueba de trabajo . . . . .	8
<b>5. Billeteras</b>	<b>8</b>

# 1. Introducción

El presente trabajo de investigación se tratará sobre la *criptomoneda*<sup>1</sup> bitcoin, haciendo foco principalmente sobre los sistemas de criptografía que utiliza.

La primer parte explicará de manera amplia lo que es la criptografía, y hará hincapié en los tipos de criptografía que implementa el protocolo Bitcoin. También incluirá las funciones de *hash* que utiliza, y los algoritmos de codificación. Luego, desarrollará sobre las transacciones en Bitcoin (su ciclo de vida y cómo son validadas), siempre incluyendo los elementos de criptografía utilizados.

Por último, el trabajo explicará brevemente la red de comunicaciones en la que corre el protocolo Bitcoin -*blockchain*- y cómo es la inclusión de nuevos datos a la red (minería de bloques). También otro tema muy importante en el protocolo y con mucha presencia de la criptografía que son las *billeteras* de bitcoin y cómo se generan.

## 1.1. Qué es Bitcoin

Bitcoin es la primer aplicación de la tecnología blockchain. Comenzó una revolución con la introducción de la primer moneda digital totalmente descentralizada, y demostró ser extremadamente segura y estable [1]

El *paper* titulado *Bitcoin: A Peer-to-Peer Electronic Cash System*, escrito por *Satoshi Nakamoto*, introduce la idea de un *dinero digital* que no necesita un banco intermediario para transferir pagos entre pares.

Bitcoin se puede definir de varias maneras; es un **protocolo**, es una **moneda digital**, y es una **plataforma**. Es una combinación de redes *peer-to-peer*, protocolos de comunicación, y software que facilitan la creación y uso de la moneda digital llamada bitcoin. Nótese que Bitcoin con *B* mayúscula se refiere al protocolo, mientras que bitcoin con *b* minúscula se refiere a la moneda. Los nodos en la red *peer-to-peer* se comunican utilizando el protocolo Bitcoin [1].

Una de las principales ventajas de bitcoin frente a otros proyectos para generar un dinero electrónico, es la forma en la que soluciona el **problema del doble gasto**<sup>2</sup>.

---

<sup>1</sup> Moneda digital o virtual diseñada para funcionar como medio de intercambio. Utiliza la criptografía para asegurar y verificar transacciones [6]

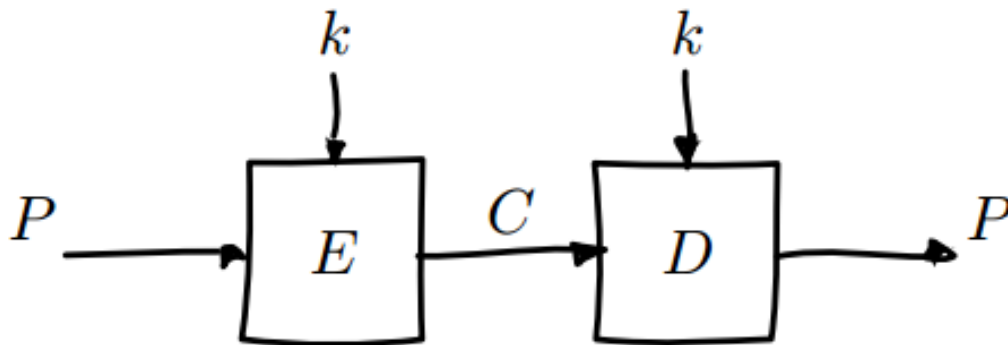
<sup>2</sup> Situación en la que se realizan dos o más transacciones con un mismo dinero

## 1.2. Criptografía

La criptografía es un área de estudio que consiste de varios esquemas y técnicas para transformar un mensaje en texto plano en un mensaje cifrado; este proceso de transformación se conoce como **encriptación**, mientras que el proceso de conseguir el mensaje original a partir del cifrado se llama **desencriptación** [13]

### 1.2.1. Simétrica

La criptografía simétrica (o *encriptación de clave secreta*) encripta un mensaje utilizando **una única llave** - la misma llave que encripta el mensaje desencripta el mensaje cifrado para obtener de nuevo el original.



Cifrado y descifrado simétrico (P representa el texto plano, C el mensaje cifrado, E y D las funciones de cifrado/descifrado respectivamente, y k la clave secreta) [9]

Los algoritmos de encriptación simétricos pueden trabajar con *bloques* (encriptando bloques de un mismo tamaño), o con *flujos* (encriptando flujos de datos, pueden ser flujos de 1 bit).

Este tipo de encriptación es muy performante y no incrementa el tamaño del mensaje, pero introduce una vulnerabilidad al tener que compartir la clave secreta entre las partes que se están queriendo comunicar.

### 1.2.2. Asimétrica

La encriptación asimétrica (o *encriptación de clave pública*), a diferencia de la simétrica que utiliza una única clave, necesita de dos llaves para funcionar: una **una privada** y una **pública**.

Este sistema de encriptación se basa en que se utiliza una clave para encriptar el mensaje, y otra clave (diferente de la primera, pero relacionada) para desencriptar el mensaje. Su característica principal se trata de que se computacionalmente inviable determinar la clave de desencripción solamente sabiendo el algoritmo de cifrado y la clave de encriptación [14].

Teniendo este par de claves, se puede operar de dos modos distintos:

- Modo encriptación: el emisor encripta el mensaje con la clave pública del receptor, de modo que sólo el receptor sea capaz de desencriptar el mensaje (utilizando su clave privada)
- Modo autenticación: el emisor encripta el mensaje (o un *digesto* del mensaje) con su clave privada y lo anexa al mensaje. El receptor desencripta este anexo con la clave pública del emisor, y compara el anexo desencriptado con el mensaje (o el *digesto* pudo generar el receptor). Si son iguales, entonces se garantiza que el mensaje fue enviado por el emisor, y que no fue alterado en el camino

Claramente este tipo de encriptación genera mayor seguridad en la comunicación, debido a que las claves para desencriptar los mensajes no se deben intercambiar entre las partes previamente a comenzar la comunicación (son privadas a cada cliente y no deben ser reveladas). Sin embargo, aumentan el tamaño del mensaje y no son algoritmos tan performantes como los simétricos.

## Criptografía de curva elíptica

Uno de los primeros algoritmos de clave pública que se diseñaron fue  $RSA^3$ , que se trataba de generar pares de claves privada/pública en base a la teoría de números primos. Para brindar mayor seguridad, el algoritmo precisa números cada vez más grandes; debido a que conseguir números primos resulta una tarea muy costosa para las computadoras, este tipo de algoritmos no es ideal para dispositivos con poco poder de cómputo o poca batería

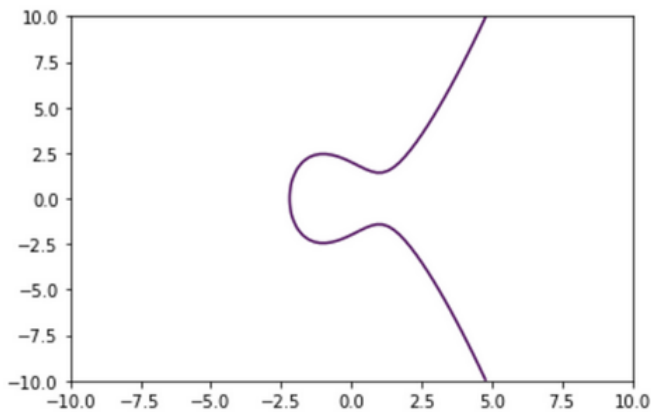
En 1985, *Neal Koblitz* y Victor Miller plantearon algoritmos de criptografía basados en *curvas elípticas* [8]. Una curva elíptica está definida por el conjunto de valores que genera la siguiente función [7]:

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

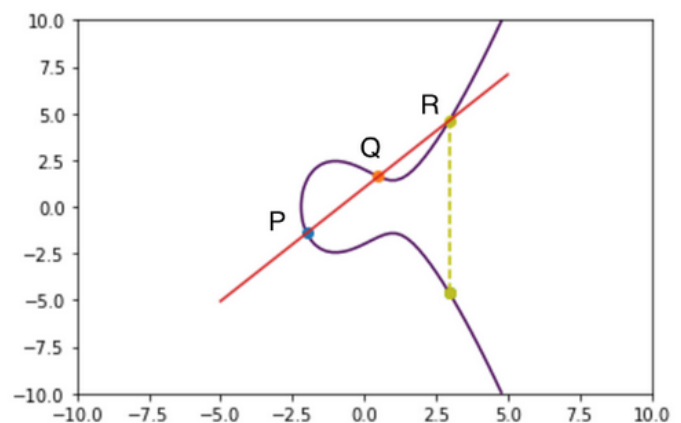
---

<sup>3</sup>Nombrado así por las iniciales de sus creadores (*Rivest-Shamir-Adleman*)

Una característica muy interesante que tienen este tipo de curvas (y que es en la que se basa el algoritmo), es que si una recta intersecta dos puntos de la curva, también intersectará un tercero



Curva elíptica [15]



Recta que atraviesa los puntos (P , Q); también atraviesa el punto (R) [15]

Nótese también que la curva es simétrica con respecto al eje Y.

Sin entrar en más detalles de la criptografía de curva elíptica, el par de claves privada/pública se genera gracias a esta característica de intersección de puntos.

### 1.3. Funciones de *hash*

Otro elemento de la criptografía muy importante para Bitcoin son las funciones de *hash* (la mayor carga de trabajo del protocolo se trata de calcular *hashes*).

Las funciones *hash* se tratan de funciones que toman una entrada de largo variable y lo convierten en un valor de largo fijo, también conocido como “digesto” [10]

#### 1.3.1. SHA256

La familia SHA (*Secure Hash Algorithm*, Algoritmo de *Hash* Seguro) es un sistema de funciones *hash* criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos (NSA) y publicadas por el National Institute of Standards and Technology (NIST) [19]. SHA256 es un *hash* de 64 dígitos hexadecimales *casi único* de un tamaño fijo de 256 bits (32 bytes) [11].

### 1.3.2. *RIPEMD160*

RIPEMD160 (acrónimo de *RACE Integrity Primitives Evaluation Message Digest*, primitivas de integridad del resumen del mensaje) es un algoritmo del resumen del mensaje de 160 bits desarrollado en Europa por *Hans Dobbertin, Antoon Bosselaers y Bart Preneel* [18].

RIPEMD160 fue diseñado en la comunidad académica abierta, en contraste con el algoritmo SHA-1, diseñado por la Agencia de Seguridad Nacional estadounidense (NSA). Por otra parte, RIPEMD160 es un diseño menos popular y no está tan estudiado como las funciones SHA. Los *hashes* de 160 bits RIPEMD (también llamados resúmenes RIPE del mensaje) se representan típicamente como números en código hexadecimal de 40 dígitos [12].

## 1.4. Codificaciones

Los algoritmos de codificación *binario-a-texto* transforman una secuencia de datos binarios a una secuencia de caracteres imprimibles. Estas codificaciones son necesarias cuando el canal de transmisión no acepta datos binarios [17]

### 1.4.1. *Base58Check*

El algoritmo de *Base58Check* es una versión modificada del *Base58*, y sirve para codificar arreglos de bytes en cadenas legibles para un humano [16]

## 2. Transacciones

Bitcoin utiliza un esquema de clave pública/privada para llevar a cabo la principal función del ecosistema de la moneda, que son las **transacciones**. Cada transacción se compone de al menos **una entrada y una salida**. Las entradas se pueden pensar como monedas siendo gastadas *que fueron creadas en transacciones anteriores*, y las salidas como monedas que están siendo creadas. Existe un caso especial, en la que una transacción no tiene entrada - estas transacciones son las que generan nuevas monedas, y son las que se registran como primera entrada en los bloques [2].

Si un usuario envía monedas a otro usuario, la transacción tiene que ser firmada con la clave privada del emisor, y también se requiere una referencia a las transacciones previas

para demostrar el origen de las monedas, y que todavía no han sido gastadas. De hecho, las monedas son *transacciones sin gastar*, representadas en *Satoshis*<sup>4</sup>.

Las transacciones no están encriptadas y son públicamente visibles.

## 2.1. Ciclo de vida

El ciclo de vida de una transacción es el siguiente [3]:

1. Un usuario genera una transacción usando algún software de Bitcoin
2. El software firma la transacción con la clave privada del usuario
3. La transacción es *difundida* a toda la red de Bitcoin (blockchain) usando un algoritmo de *inundación* (*flooding*)
4. Los nodos *mineros* incluyen la transacción en el próximo *bloque* a minar
5. El minado comienza una vez que el minero que resolvió la *prueba de trabajo* (*Proof of Work*) difunda el nuevo bloque a la red
6. Los otros nodos verifican el bloque y los siguen propagando
7. Finalmente, la confirmación de la transacción figura en la cuenta (*billetera*) del receptor, y luego de varias confirmaciones (estadísticamente, seis confirmaciones es suficiente), la transacción se considera confirmada y finalizada

## 2.2. Verificación de una transacción

Cada transacción tiene que ser verificada por los nodos que componen la red de Bitcoin. Los aspectos a verificar son los siguientes [4]:

1. La transacción respeta la sintaxis del lenguaje de *scripting*<sup>5</sup> de Bitcoin
2. Las entradas y las salidas no son vacías
3. El tamaño máximo de la transacción no supera el tamaño máximo del bloque (actualmente 1 MB)

---

<sup>4</sup>1/100000000 bitcoin

<sup>5</sup>Lenguaje basado en pilas llamado *script* que describe cómo las monedas pueden ser gastadas y transferidas [5]



4. El valor de las salidas no debe ser menor a 0 ni mayor a 21 millones de bitcoin
5. Todas las entradas deben tener especificada una salida anterior (salvo las transacciones que generan monedas)
6. No debe existir una transacción igual en ningún bloque, ni esperando ser confirmada
7. Las salidas que corresponden a las entradas de la transacción no deben figurar en ningún otra transacción (problema del *double spending*)
8. Para cada entrada, debe existir su correspondiente salida en alguna transacción
9. Para cada entrada, si la salida correspondiente referenciada es la salida que origina las monedas (llamada *coinbase transaction*), esta última debe tener al menos 100 confirmaciones
10. Para cada entrada, si la salida correspondiente referenciada no existe o ya fue gastada, la transacción es rechazada
11. La suma de las entradas debe ser mayor o igual al total de las salidas (si el valor de las entradas es mayor, el resto se considera *tarifa de la transacción* <sup>6</sup>)
12. La tarifa de la transacción debe ser mayor o igual a un valor mínimo establecido

### **3. Blockchain**

#### **3.1. Estructura de un bloque**

### **4. Minería de bloques**

#### **4.1. Prueba de trabajo**

### **5. Billeteras**

---

<sup>6</sup>Las tarifas de las transacciones definen la prioridad que tendrá la transacción al ser elegida por los mineros para agregarla a un nuevo bloque

## Referencias

- [1] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin*.
- [2] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - Transactions*.
- [3] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - The transaction life cycle*.
- [4] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - Transaction verification*.
- [5] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - The script language*.
- [6] Cointelegraph. *¿Qué es criptomoneda? Guía para principiantes*. URL: <https://es.cointelegraph.com/bitcoin-for-beginners/what-are-cryptocurrencies#futuro-de-las-criptomonedas>.
- [7] Andrea Corbellini. *Elliptic Curve Cryptography: a gentle introduction*. Mayo de 2015. URL: <http://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/>.
- [8] Grayblock. *Elliptic-Curve Cryptography*. Jun. de 2018. URL: <https://medium.com/coinmonks/elliptic-curve-cryptography-6de8fc748b8b>.
- [9] Laurens Van Houtven. «Crypto 101». En: Creative Commons, 2013. Cap. 6 - *Block ciphers*.
- [10] Laurens Van Houtven. «Crypto 101». En: Creative Commons, 2013. Cap. 10.1 - *Hash functions - Description*.
- [11] Alex Preukschat. *¿Qué es y de qué sirve el algoritmo SHA-256 en el protocolo Bitcoin? – Secure Hash Algorithm (VII)*. Ene. de 2014. URL: <https://www.oryfinanzas.com/2014/01/algoritmo-sha-256-protocolo-bitcoin-secure-hash-algorithm/>.
- [12] Alex Preukschat. *¿Qué es y por qué se utiliza el algoritmo RIPEMD-160 en la creación de claves públicas Bitcoin? (VIII)*. Ene. de 2014. URL: <https://www.oryfinanzas.com/2014/01/ripemd-160-bitcoin/>.
- [13] Williams Stallings. «Cryptography and Network Security». En: séptima edición. Pearson, 2017. Cap. 3 - *Classical Encryption Techniques*.

- [14] Williams Stallings. «Cryptography and Network Security». En: septima edición. Pearson, 2017. Cap. 9.1 - *Principles of Public-Key Cryptosystems*.
- [15] Hackernoon - Short Tech Stories. *Elliptic Curve Crypto, The Basics*. Jun. de 2017. URL: <https://hackernoon.com/eliptic-curve-crypto-the-basics-e8eb1e934dc5>.
- [16] Wikipedia. *Base58Check Encoding*. Nov. de 2017. URL: [https://en.bitcoin.it/wiki/Base58Check\\_encoding](https://en.bitcoin.it/wiki/Base58Check_encoding).
- [17] Wikipedia. *Binary-to-text encoding*. Dic. de 2018. URL: [https://en.wikipedia.org/wiki/Binary-to-text\\_encoding](https://en.wikipedia.org/wiki/Binary-to-text_encoding).
- [18] Wikipedia. *RIPEMD160*. Sep. de 2015. URL: <https://es.wikipedia.org/wiki/RIPEMD-160>.
- [19] Wikipedia. *Secure Hash Algorithm*. Nov. de 2018. URL: [https://es.wikipedia.org/wiki/Secure\\_Hash\\_Algorithm](https://es.wikipedia.org/wiki/Secure_Hash_Algorithm).