UNPSJB

LICENCIATURA EN SISTEMAS OPGCPI

ASPECTOS LEGALES Y PROFESIONALES

Blockchain

Una alternativa descentralizada para firma digital

Cátedra
Dr. Guillermo Cosentino
Lic. Bruno Damián Zappellini

Integrantes:
Luciano Serruya Aloisi

19 de diciembre de 2018

Índice

1. Introducción

El presente trabajo de investigación se tratará sobre la tecnología *blockchain*[1], haciendo foco en sus capacidades para implementar un mecanismo de firma digital sobre la misma.

La primer parte explicará de manera amplia conceptos de criptografía y los tipos que existen, y funciones *hash*. Luego, desarrollará sobre el concepto de firma digital (y la situación actual en la Argentina), y sobre la *Infraestructura de Clave Pública* (*PKI*).

Por último, el trabajo abarcará el concepto de *blockchain* y cómo se podría implementar una infraestructura de firma digital sobre blockchain.

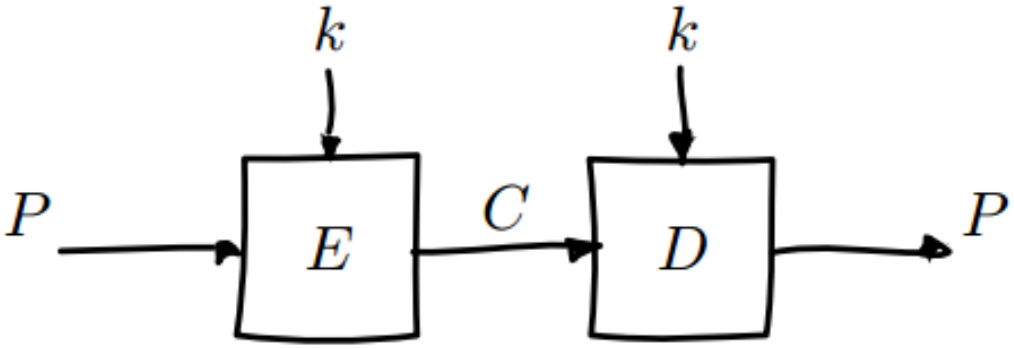
1.1. Criptografía

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 **encripción**, mientras que el proceso de conseguir el mensaje original a partir del cifrado se llama **desencripción** [11]

1.1.1. Simétrica

La criptograía simétrica (o *encripción 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.

[1] Infraestructura de computadoras sobre la que corre el protocolo Bitcoin, y otras criptomonedas y aplicaciones descentralizadas



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) [7]

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.

1.1.2. Asimétrica

La encriptación asimétrica (o *encripción 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 es que es computacionalmente inviable determinar la clave de desencripción solamente sabiendo el algoritmo de cifrado y la clave de encripción [12].

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

- Modo encripción: el emisor encripta el mensaje con la clave pública del receptor, de modo que sólo el receptor sea capaz que 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 los 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 encripción 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.

1.2. Funciones de *hash*

Otro elemento de la criptografía muy importante para blockchain 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" [8]. Una característica muy importante de estas funciones, es que es fácil computar, pero muy difícil (hasta imposible) revertir -conseguir la entrada que originó cierto *hash*-

2. Firma digital

El envío de un documento *digitalmente firmado* es necesario cuando se precisan las siguientes tres condiciones:

1. Verificar que el mensaje no fue alterado en el camino (*integridad*)
2. Verificar que el emisor realmente fue el que envió el mensaje (*autenticación*)
3. Verificar que el emisor quiso enviar el mensaje (*no repudio*)

Para esto, la firma digital consiste en utilizar la clave privada en el *modo autenticación* (como fue descripto en la subsección de *Criptografía asimétrica*).

El emisor genera un digesto del mensaje (aplicando una función de *hash* al mensaje), y dicho digesto es el que se encripta con la clave privada del emisor. El resultado de encriptar el digesto se conoce como **firma digital**; se la anexa al documento y se envía. El receptor genera el mismo *hash* del documento, desencripta la firma digital con la clave pública del emisor, y compara su resultado con el digesto: si son iguales, las tres condiciones anteriormente nombradas se cumplen.

2.1. Situación en Argentina

El 11 de Diciembre de 2001, el Senado y la Cámara de Diputados de la Nación Argentina sancionó con fuerza de ley la *Ley de Firma Digital* (Ley 25506). En su primer capítulo, en el artículo 2, se describe a la firma digital de la siguiente manera [5]:

> *Se entiende por firma digital al resultado de aplicar a un documento digital un procedimiento matemático que requiere información de exclusivo conocimiento del firmante, encontrándose ésta bajo su absoluto control. La firma digital debe ser susceptible de verificación por terceras partes, tal que dicha verificación simultáneamente permita identificar al firmante y detectar cualquier alteración del documento digital posterior a su firma*

Esta definición da a entender un procedimiento de firma digital similar al descripto previamente, donde el emisor firmaría con su *clave privada* (*conocimiento encontrado bajo absoluto control del firmante*) el mensaje o su digesto.

A lo largo del desarrollo de la ley, se habla también de conceptos relacionados con la tecnología *PKI*, tales como *Certificado*, *Certificador licenciado*, y *Ente licenciante*.

3. *PKI*: *Public Key Infrastructure*

La RFC 4949 define la *Infraestructura de Clave Pública* (*PKI*) como un conjunto de hardware, software, personas, políticas, y procedimientos necesarios para crear, manejar, almacenar, distribuir, y remover certificados digitales basados en criptografía asimétrica. El objetio principal para desarrollar PKI es establecer una forma segura, conveniente, y eficiente de adquirir claves públicas [13].

PKI se compone de varios componentes:

- Entidad final: concepto que engloba tanto usuarios finales, dispositivos o servidores, o cualquier otra entidad que necesite ser identificada en el ámbito de certificados de clave pública. Estas entidades mayormente consumen y/o soportan servicios relacionados con PKI
- Autoridad de Certificación (*Certification Authority - CA*): emisor de certificados y, usualmente, de las listas de revocación de certificados
- Autoridad de Registro (*Registration Authority - RA*): componente opcional (la CA puede llevar a cabo sus funciones) que se encarga de verificar que un usuario que solicita un certificado es quién dice ser.

- Emisor de CRL: componente opcional (la CA puede llevar a cabo sus funciones) que se encarga de emitir las *listas de revocación de certificados* (listas que indican todos los certificados que han sido revocados, y por lo tanto no se deberían aceptar)
- Repositorio: termino genérico para representar cualquier método para almacenar certificados y listas de revocación de certificados para que puedan ser accedidos por usuarios finales

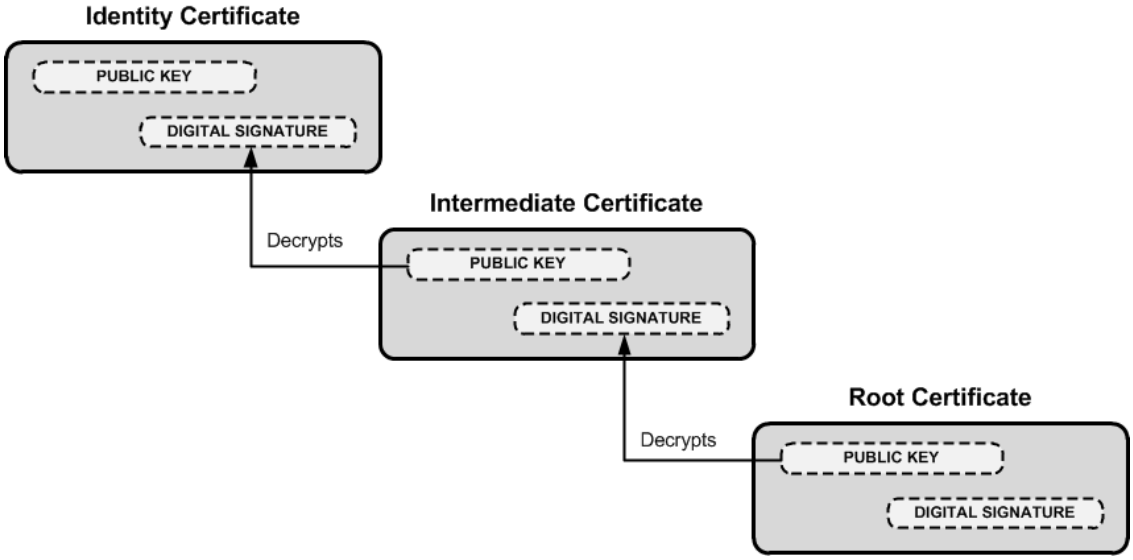
3.1. Funcionamiento de *PKI*

Cuando un cliente desea obtener un certificado de una CA, primero debe demostrar que es quién dice ser. En el caso de un servidor web, esta verificación de identidad podría consistir en demostrar que el dominio para el cual desea obtener el certificado realmente le corresponde. Estos pasos iniciales son llevados a cabo contra la RA (o la propia CA, en caso de que no haya RA).

Una vez demostrado que el servidor es quién dice ser, el cliente le debe enviar su clave pública a la CA; este último generará un certificado (con una fecha de vencimiento) en el cual incluye el nombre de dominio, la clave pública del servidor, y la firma digital de la CA (provee autenticidad de que la CA confía en el servidor). Teniendo este certificado, usuario finales se podrán conectar de manera segura con el servidor web.

Ahora bien, la CA que emitió el certificado para el servidor a su vez también cuenta con un certificado, el cual fue emitido por otra CA. Las **CA raíz** (***root CA***) son los primeros emisores de esta cadena de certificados. Los navegadores web ya cuentan con las claves públicas de varias CA raíz para validar los certificados.

Cuando un usuario se conecta al servidor, primero le solicita el certificado al servidor; éste le contesta enviándole tanto su certificado como el de su CA. Primero el cliente verifica el certificado de la CA validando la firma digital con la clave pública de la CA raíz que emitió el certificado (el navegador cuenta con dicha clave pública). Luego, repite el proceso con el certificado del servidor, y con la clave pública de la CA intermedia.



Verificación en cadena de certificados [9]

Una vez verificados todos los certificados, se puede establecer una conexión segura entre el cliente y el servidor.

4. *Blockchain*

Blockchain, o la *cadena de bloques*, es el "libro de cuentas" público, ordenado en el tiempo, e *inmutable* de todas las transacciones en la redes de criptomonedas. Cada bloque está identificado por un *hash* en la cadena y está vinculado con su bloque anterior referenciando su *hash*. Todos los bloques están relacionados con su bloque anterior, a excepción del primer bloque, también llamado bloque *génesis*. También se puede definir blockchain como una base de datos distribuida, en donde cada nodo tiene una copia entera de la base.

Bloques nuevo son añadidos a la cadena cada cierto tiempo. El protocolo de blockchain maneja una *dificultad*[2] para agregar nuevos bloques. Puede ir aumentando o disminuyendo según la capacidad de cómputo disponible en la red, para que se mantenga la frecuencia de un bloque nuevo cada el tiempo determinado (10 minutos en la blockchain de Bitcoin) [1].

[2]La dificultad signifca qué tan difícil es para los mineros generar un bloque nuevo

4.1. Minería de bloques

La minería de un bloques es una tarea que conlleva muchos recursos (de hardware y de electricidad) mediante el cual se agregan nuevos bloques a la red. Los bloques contienen las transacciones que son validadas mediante el proceso de minado (llevado a cabo por los nodos mineros).

Los *mineros* (computadoras que están corriendo el software necesario para ejecutar los algoritmos de minado) que generen correctamente el nuevo bloque, se llevan una recompensa (en Bitcoin, el minero se lleva una cantidad de bitcoins). De esta forma, existe un incentivo para que se dispongan computadoras y recursos en pos de que funcione el protocolo y de mantenerlo activo.

Cuando se genera un nuevo bloque, el minero que lo generó hace una *difusión* del bloque nuevo a toda la red para que el resto de los mineros lo verifique. Esto asegura el sistema contra fraudes y ataques de *doble gasto*[3].

El concepto de "minería" se debe a que las monedas (en el caso de las blockchain de criptomonedas) se crean o se "emiten" al crear un nuevo bloque; es una analogía a la minería de oro, en donde en base a un trabajo se obtiene algo de valor.

4.2. Prueba de trabajo

La prueba de trabajo (*Proof of Work*) es una demostración de que suficiente poder computacional fue empleado para construir un bloque válido. En este modelo, los nodos compiten para ser seleccionados en proporción a su capacidad computacional.

La prueba de trabajo consiste en calcular el *hash* del bloque. Éste se compone de la suma de todos los datos del bloque, y debe comenzar con *n* número de ceros (o que sea menor a cierto valor). La cantidad de ceros o el valor *objetivo* está definido por la *dificultad* de la red.

Debido a que es fácil calcular el *hash* de un valor, pero imposible conseguir el valor original a partir de un *hash*, la única forma de calcular el *hash* del bloque es **con fuerza bruta** (probar valores hasta encontrar un resultado). El campo `nonce` de la cabecera del bloque es el que tienen que ir incrementando los mineros hasta encontrar el *hash* indicado.

[3] Situación en la que se realizan dos o más transacciones con un mismo dinero

$$H(N||P\_hash||Tx_1||Tx_2||...||Tx_n) < Objetivo$$

Cálculo de la cabecera del bloque, donde `H` es la función de *hash*, `N` es el *nonce*, `P_hash` es el *hash* del bloque anterior, `Tx` son las transacciones que incluye el bloque, y `Objetivo` la dificultad

Una vez conseguido el *hash*, el bloque es inmediatamente difundido por la red. Debe ser aceptado por los otros mineros para ser agregado a la red [2].

Blockchain se dice que es "inmutable" debido a la dificultad que conllevaría modificar un bloque. En caso de modificar un bloque, el *hash* se vería modificado también, por lo tanto habría que recalcularlo (lo cual es una tarea muy costosa). No solo eso, sino que si el bloque no es el último de la cadena -tiene otros bloques por delante-, habría que modificar esos otros bloques también (ya que cada bloque tiene el *hash* del anterior, a modo de puntero). Por lo tanto, blockchain no es en sí inmutable, pero modificar un dato que ya se escribió en un bloque conllevaría demasiado trabajo, haciéndolo inviable.

Algoritmo de minado

La secuencia de pasos que llevan a cabo los mineros que crear un nuevo bloque es la siguiente [3]

- El bloque anterior se recupera de la red
- Se obtiene un conjunto de posibles transacciones que podrían conformar el bloque
- Computar el *hash* de la cabecera del bloque con un *nonce* = 0
- Si el *hash* obtenido es menor al objetivo, detener el proceso
- Si no es menor, repetir el proceso incrementando el *nonce*

4.3. *PKI* sobre *Blockchain*

La Infraestructura de Clave Publica provee un metodo seguro para autenticar identidades sobre Internet. Define las políticas y los procedimientos necesarios para emitir, gestionar, validar y distribuir certificados para poder usar encripción asimétrica de forma segura. El manejo de claves públicas de PKI usualmente está basado en el estándar de

certificados *X.509*, que provee verificación de posesión de una clave privada gracias a un ente externo (Autoridad Certificante).

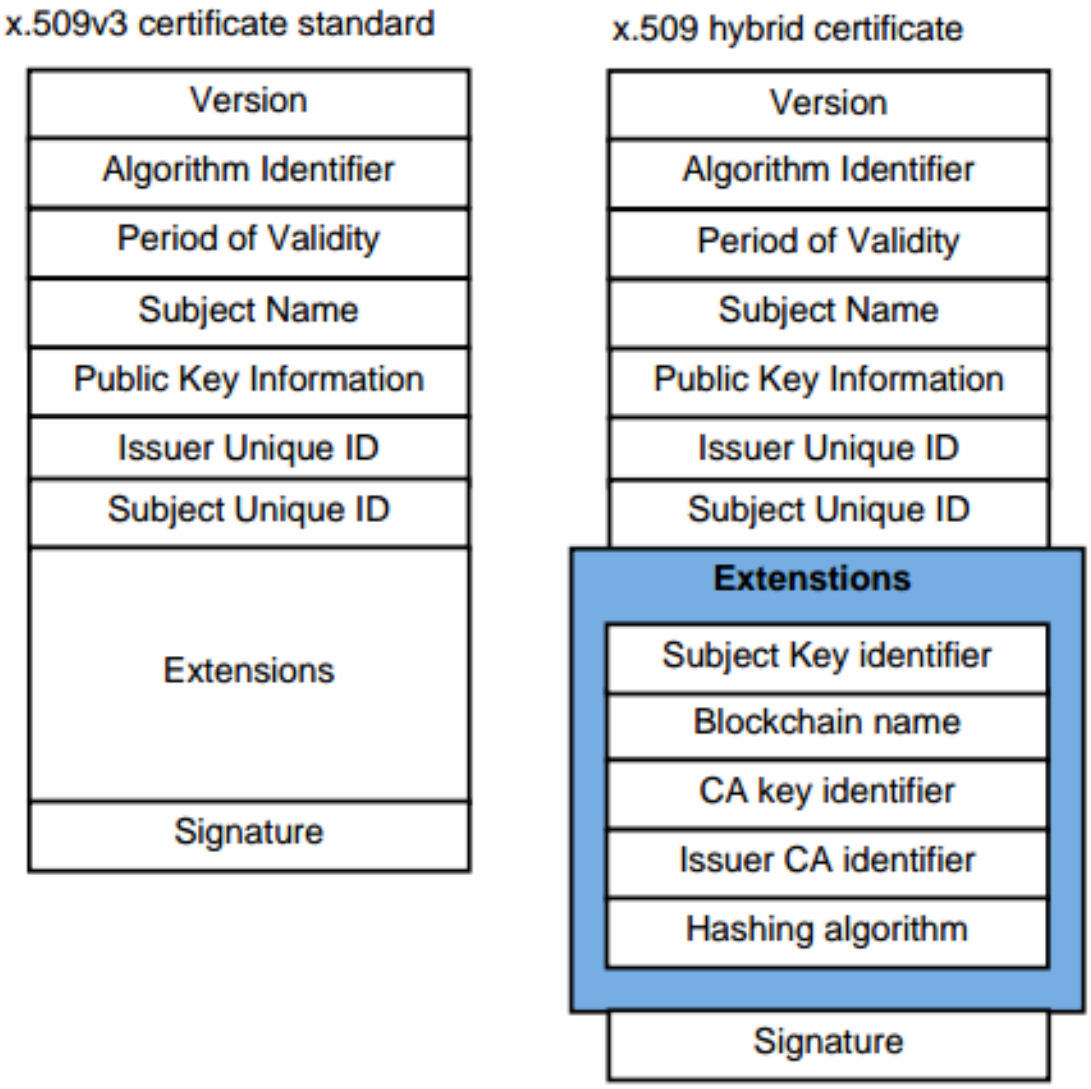
Sin embargo, las CA no dejan de ser *puntos únicos de fallo* (*single point of failure*), lo que significa que errores presentes (principalmente de seguridad, haciendo propenso a una CA al ataque de un *hacker*) en las CA significan emisiones de certificados sin autorizar [10]. Estos problemas no sólo afectan a las CA intermedias, sino que a las raíces también.

Para resolver este problema, se plantearon dos soluciones distintas. La primera de ellas consiste en una *Infraestructura de Clave Pública basada en logs* (*log-based PKI*), la cual consiste en implementar servidor con alta disponibilidad que registren y monitoreen todos los certificados emitidos por la CA. Los certificados no son considerados válidos hasta no ser registrados en el *log*, de esta manera se expone rápidamente cualquier emisión incorrecta. Ante la detección de cualquier comportamiento extraño por parte de la CA, es responsabilidad del dominio de actuar frente a los certificados no autorizados que han sido emitidos.

La otra solución planteada es un enfoque descentralizado: *Red de Confianza* (*Web of Trust*). En esta propuesta, los usuarios son encargados de indicar que otros usuarios son confiables firmando sus certificados. De esta forma, un usuario acumula varias firmas en su certificado, generando confianza (enfoque similar al de PGP). De esta forma se elimina el punto único de fallo, pero la desventaja que se introduce es que los nuevos usuarios que ingresen a la red no tendrán ninguna firma en su certificado, por lo tanto les resultará inútil.

En el contexto de PKI, blockchain puede proveer características de seguridad muy valiosas como revocación de certificados, eliminación de puntos de fallo, y un registro de transacciones fiable.

Existen varias implementaciones de PKI sobre blockchain; [14] plantea un *framework* gestionar un sistema de PKI sobre blockchain que soporta la revocación de certificados. Su diseño está basado en *certificados X.509 híbridos*.



Certificados X.509 híbridos [14]

La idea principal de esta implementación es que cada CA tiene un *contrato inteligente* (*smart contract*)[4] dedicado que incluye la siguiente información:

- Lista on los *hashes* de los certificados emitidos
- Lista con los *hashes* de los certificados revocados

Tanto los CA raíces como los CA intermedios se representan con un contrato dedicado.

[4] Programas que corren sobre la blockchain y modifican su estado [4]

Con esta estructura, crear una nueva CA consiste en crear un nuevo contrato (la CA padre crea el contrato, y agrega el *hash* del certificado de la CA hija a su lista de CA emitidos).

Como todos los nodos que componen la red tienen una copia de la base de datos (tienen una copia de la blockchain), verificar un certificado sigue siendo el mismo procedimiento que en la versión original de PKI - se debe validar toda la cadena de certificados, desde la CA raíz hasta el usuario final.

En el caso de que se deba revocar un certificado, sencillamente la CA encargada de emitir dicho certificado actualiza sus listas de *hashes* moviendo el *hash* del certificado a revocar de la lista de *hashes* emitidos a la de los revocados. Esta notificación será enviada al resto de los nodos instantáneamente, logrando una actualización de los certificados activos, y eliminando la necesidad de una CRL.

Esta característica también significa una protección frente a *atanque de hombre en el medio* (*Men-in-the-middle attacks*)[5]. En la versión original de PKI, en caso de que una CA se vea comprometida, los usuarios finales tendrían que esperar a la actualización de la CRL para recién anular el certificado de la CA en cuestión (mientras tanto podrían recibir certificados falsos firmados por la CA comprometida). Con PKI corriendo sobre blockchain, esta situación se vuelve virtualmente imposible.

5. Conclusión

Blockchain como tecnología moderna, sigue en una etapa de mucho auge. Debido a esto, muchas soluciones de software o arquitecturas se intentan plantear en blockchain, muchas sin éxito.

Particularmente con PKI, blockchain (complimentado también con los *smart contracts*) puede ser una buena alternativa, logrando tener una infraestructura que certifica la autenticidad de un ente, corriendo en una red descentralizada de computadoras sin contar con un esquema de confianza, manteniendo la seguridad y la integridad, y sin contar con un administrador central [6].

[5] Ataque a la seguridad que consiste en enviar un certificado falso, haciéndolo pasar por uno válido, para así interceptar la comunicación

Referencias

[1] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - Blockchain*.

[2] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - Proof of Work*.

[3] Imran Bashir. «Mastering Blockchain». En: primer edición. Packt Publishing, mar. de 2017. Cap. 4 - *Bitcoin - The mining algorithm*.

[4] Vitalik Buterin. *A Next-Generation Smart Contract and Decentralized Application Platform*. 2014. URL: https://github.com/ethereum/wiki/wiki/White-Paper.

[5] Senado y Cámara de Diputados de la Nación Argentina. *Ley 25506: Firma Digital*. Dic. de 2001.

[6] Gideon Greenspan. *Why Many Smart Contract Use Cases Are Simply Impossible*. Abr. de 2016. URL: https://www.coindesk.com/three-smart-contract-misconceptions.

[7] Laurens Van Houtven. «Crypto 101». En: Creative Commons, 2013. Cap. 6 - *Block ciphers*.

[8] Laurens Van Houtven. «Crypto 101». En: Creative Commons, 2013. Cap. 10.1 - *Hash functions - Description*.

[9] PacketPimp3. *PKI - Chain of Trust*. Mar. de 2012. URL: https://www.fir3net.com/Security/Concepts-and-Terminology/pki-chain-of-trust.html.

[10] H. Anada; J. Kawamoto; J. Weng; K. Sakurai. «Identity-embedding method for decentralized public-key infrastructure». En: *International Conference on Trusted Systems* (2014), págs. 1-14.

[11] Williams Stallings. «Cryptography and Network Security». En: septima edición. Pearson, 2017. Cap. 3 - *Classical Encryption Techniques*.

[12] Williams Stallings. «Cryptography and Network Security». En: septima edición. Pearson, 2017. Cap. 9.1 - *Principles of Public-Key Cryptosystems*.

[13] Williams Stallings. «Cryptography and Network Security». En: septima edición. Pearson, 2017. Cap. 14.5 - *Publica-Key Infrastructure*.

[14] Alexander Yakubov; Wazen M. Shbair; Anders Wallbom; David Sanda; Radu State. *A Blockchain-Based PKI Management Framework*.