Integridad de archivos y su distribución en una red Blockchain

Sebastian Ponti, Lautaro Nahuel De León, Patricia Bazán
LINTI Facultad de Informática UNLP
sebaponti@gmail.com, laudleon@gmail.com, pbaz@ada.info.unlp.edu.ar

RESUMEN

En la actualidad, los métodos para garantizar la integridad de archivos tienen como problemática que el respaldo de dicha característica se apoya en un servidor central, con lo cual, es relativamente simple que alguien con malas intenciones pueda realizar alteraciones de estos datos, sin dejar rastros. En el año 2008, se publicó un paper titulado "Bitcoin: Un Sistema de Efectivo Electrónico Usuario-a-Usuario" [1] el cual explica el funcionamiento de un sistema monetario electrónico totalmente descentralizado, es decir, sin intervención de una entidad tercera que valide la autenticidad de las transacciones. El principio básico de este sistema es justamente distribuir la información entre todos los nodos dentro de la red de forma tal que cada uno de ellos mantenga en todo momento el mismo historial de transacciones realizadas y donde las mismas sean completamente inmutables, a fin de mantener la buscada integridad y confiabilidad de estos datos.
En este artículo se enuncia una alternativa de solución al problema mencionado a través de una aplicación concreta utilizando la plataforma Ethereum [2], la cual es una implementación de una blockchain que permite realizar cómputo distribuido y seguro sobre ella por medio de contratos inteligentes [3].

CONTEXTO

El presente trabajo se encuentra inserto en un área de innovación que parte originalmente del estudio de los sistemas distribuidos -particularmente los descentralizados-, pero que en los últimos años, dado el vertiginoso crecimiento de los fundamentos propios que caracterizan a este tipo de arquitecturas, las blockchain, ha divergido en una línea de investigación propia y sumamente fértil, con un vasto potencial para muchas aplicaciones.

1. INTRODUCCION

Desde el surgimiento de Internet en 1990 y el advenimiento de la descarga de archivos a través de esta red, siempre ha sido un problema garantizar la integridad de los mismos, es decir, poder probar mediante un método riguroso que estos no son alterados en ningún momento. Estas potenciales modificaciones pueden ser aleatorias, por ejemplo, debido a fallas del medio de transmisión, o bien, porque un agente malintencionado introduce un cambio indebido.
Dentro del marco de Internet, uno de los primeros algoritmos utilizados para verificar la integridad de los archivos transferidos fue MD5 [4], el cual pertenece a la familia de funciones de *hash* de una sola vía (*one-way function*). Este tipo de funciones tiene como objetivo asociar cada elemento del dominio que, en el presente caso, está caracterizado por el conjunto de archivos posibles (el cual es infinito) con una cadena pseudo-aleatoria y fija de caracteres, la cual recibe el nombre de huella digital. La particularidad de las funciones de una sola vía es que el costo computacional que se invierte para obtener dicha huella digital a partir de un archivo - o cualquier fuente en general- es muy barato, mientras que el proceso inverso es virtualmente imposible. Esta característica las hace especialmente idóneas para la verificación de integridad de archivos sobre redes de datos:

genéricamente, si en un nodo cualquiera de la red se computa la huella digital y se deja accesible de forma pública, entonces cualquier otro nodo de la misma red podría corroborar que el archivo transferido fue recibido de forma intacta computando la misma función sobre él, y luego, verificando que el resultado sea igual a la huella generada por el nodo origen; de lo contrario, se infiere que hubo algún tipo de alteración sobre él.

Aun en el caso de disponer de una función de una sola vía relativamente segura -en 2005 se demostró que MD5 era insegura [5]-, muchas veces el mayor problema recae en la arquitectura, particularmente, aquellas en la que la seguridad se concentra en un único punto o servidor. Además de los problemas de rendimiento conocidos, en donde un número reducido de servidores podría colapsar ante una gran cantidad de peticiones por parte de los usuarios o clientes [6], existen también problemas de seguridad asociados. Si, por ejemplo, un atacante consigue tomar control del/los servidor/es del sistema, entonces cualquier información alojada en ella dejaría de ser automáticamente confiable, en particular, las huellas digitales publicadas de los archivos ubicados allí. Al administrar el servidor, un agente malintencionado podría cambiar los archivos, recomputar sus huellas y volver a publicarlas al exterior y todo el circuito parecería confiable, pero sin embargo, el agente ha conseguido romper un eslabón de confiabilidad y eso podría ser muy costoso para el proceso o negocio involucrado.

Dadas las problemáticas de las redes centralizadas basadas en arquitecturas cliente-servidor, una variante desarrollada en pos de mitigar este inconveniente fue pensar las redes sin la noción típica de "servidor central", sino más bien, pensar en redes donde cada nodo participante de ella pueda requerir información o trabajo como así también suministrar información o trabajo. Cabe destacar que hay distintos niveles de descentralización; por ejemplo, lo que se describió previamente es una instancia de red descentralizada pura dado que todos los nodos actúan como clientes y servidores a la vez. Sin embargo, esto no necesariamente se da así en la realidad. En otros casos la red puede ser mixta, esto es, una red con un número equilibrado y bien distribuido -ya sea lógica o físicamente- de servidores y clientes. Estas características son cruciales para el presente trabajo puesto que la confianza ahora no dependerá solamente de un único punto sino de muchos otros, y esto, disminuirá exponencialmente la probabilidad de obtener todo el control de la red con el objeto de anunciar información espuria sobre ella.

Aproximadamente a partir de la década del 80 se han desarrollado una gran cantidad de ideas abogando por las bondades de la descentralización [7] [8] [9], llegando hasta el mencionado año 2008 en donde se publica el trabajo de Satoshi Nakamoto (pseudónimo) sobre la red Bitcoin, implementando el primer modelo completo de lo que hoy se conoce como Blockchain. En esta arquitectura, todas las transacciones son públicas y distribuidas, y una vez validadas, no pueden revertirse ni alterarse. Esto se consigue agrupando a las mismas en bloques, en donde cada una de estas últimas conforman los eslabones de una cadena en la cual cada uno se identifica con una huella digital que depende parcialmente de las huellas de los bloques previos, protegiendo así a la cadena completa de modificaciones, dado que, de lo contrario, la cadena de huellas quedaría completamente inválida [10] [11].

Por otro lado, con el objeto de evitar que cualquier nodo publique transacciones falsas, se exige que cada bloque lleve consigo una prueba de trabajo (*PoW*), en donde los nodos mineros o validadores de transacciones y bloques resuelvan un desafío computacional muy costoso de llevar a cabo pero muy sencillo de verificar con el propósito de que, justamente,

un nodo demuestre interés genuino en validar, y por ende, asegurar el buen funcionamiento de la red por medio de su inversión de trabajo [12].

Dado que Bitcoin únicamente sirve para transacciones monetarias, en el año 2014, sale a la luz la plataforma Ethereum, la cual implementa también una Blockchain con los mismos fundamentos que la de Bitcoin, pero con la gran diferencia y ventaja de que puede albergar fragmentos de código programables por los usuarios llamados "contratos inteligentes". Estos, al igual que una transacción, se pueden ejecutar y validar en cualquier nodo de la red, otorgando cómputo descentralizado.

En pos de resolver el problema previamente explicado en este artículo, se desarrollará una pequeña aplicación cuyo objetivo será el poder certificar archivos y distribuir dicha certificación -una huella digital del mismo, en términos simples- en la red Ethereum a través de los mencionados contratos inteligentes con el fin de mitigar considerablemente el riesgo de ataque y malversación del servicio que brinda la certificación.

La arquitectura de la aplicación contará con las siguientes componentes e interacciones:

- Frontend: una aplicación que brinde las operaciones necesarias al usuario como para listar, verificar y dar de alta certificaciones de archivos.
- Servidor de archivos: lugar donde residirán los archivos. Debería proveer por archivo una huella y una identificación unívoca y anónima.
- Blockchain (Backend): el corazón de la aplicación, en donde se persistirán y distribuirán las huellas de los archivos.

Lo que se intenta describir son los tres componentes básicos, el *frontend*, el servidor de archivos y el *backend* o blockchain, el cual necesariamente estará distribuido y replicado por su naturaleza intrínseca. Habrá interacciones bidireccionales entre el *frontend* y el servidor de archivos y entre el primero y la blockchain para las distintas operaciones a especificar. Como se ha dicho anteriormente, opcionalmente el servidor de archivos podría estar replicado y las aplicaciones clientes también, si en futuro, por ejemplo, fueran aplicaciones móviles.

A modo de ejemplo, se describe una de las operaciones del sistema, la de alta de una certificación, por medio de un diagrama de secuencia (ver figura 1).

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

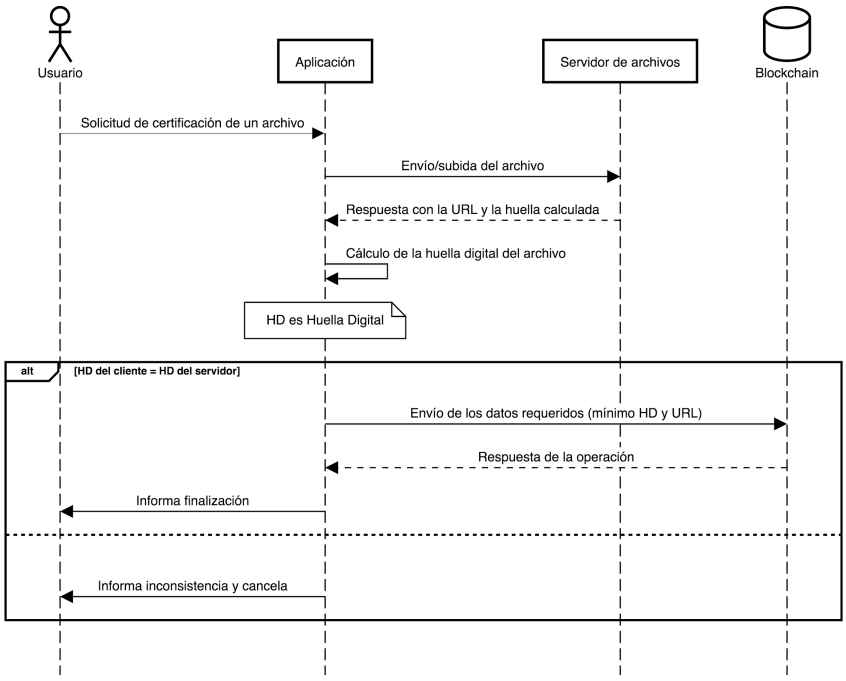

Figura 1: Diagrama de secuencia de alta de una certificación

El trabajo pone en manifiesto, por un lado, los conceptos de sistemas distribuidos, y particularmente, aquellos referidos a Blockchain y Ethereum: transacciones, consenso distribuido y contratos inteligentes. Por otro lado, se atacan cuestiones propias e inherentes al desarrollo de este tipo de aplicaciones, que interactúan con componentes desplegados en la Blockchain, como por ejemplo, diseño de la infraestructura, testing, análisis y evaluación de las diversas tecnologías presentes para llevar a cabo dichas tareas [13] [14] [15] [16] [17].

3. RESULTADOS OBTENIDOS/ESPERADOS

El resultado que intenta otorgar este trabajo es la capacidad de corroborar la existencia e integridad de un archivo en un momento determinado -prueba de existencia-, utilizando un medio de alojamiento en el cual no puede ser alterado o destruido.

Asimismo se hace foco en aportar un nivel de confianza y seguridad para el usuario que decida corroborar que, por ejemplo, evidencias fotográficas en un proceso de auditoría no fueron falsificadas luego de haberlas subido a un servidor centralizado.

4. FORMACION DE RECURSOS HUMANOS

La línea de investigación que se describe en el presente trabajo involucra el desarrollo de aplicaciones descentralizadas, la cual propone un nuevo paradigma a la arquitectura clásica que se conoce hoy en día en las aplicaciones orientadas a sistemas distribuidos. Debido a que la línea de investigación se estudia en una época de maduración temprana de la tecnología blockchain, el número de profesionales e investigadores involucrados en la misma es un grupo reducido.

En el presente trabajo se describe la línea de investigación que dio lugar a que en el año 2018 se formule la primera propuesta para tesina de grado haciendo uso de la tecnología blockchain como eje principal.

5. BIBLIOGRAFIA

[1] Satoshi Nakamoto. Bitcoin. A Peer-to-Peer Electronic Cash System, 2008.

[2] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. white paper, 2014.

[3] Nick Szabo. Formalizing and securing relationships on public networks. First Monday, 2(9), 1997.

[4] Ronald Rivest. The md5 message-digest algorithm. 1992.

[5] Xiaoyun Wang and Hongbo Yu. How to break md5 and other hash functions. In Annual international conference on the theory and applications of cryptographic techniques, pages 19–35. Springer, 2005.

[6] J. E. Neilson, C. M. Woodside, D. C. Petriu, and S. Majumdar. Software bottlenecking in client-server systems and rendezvous networks. IEEE Transactions on Software Engineering, 21(9):776–782, Sep 1995. ISSN 0098-5589. doi: 10.1109/32.464543.

[7] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM, 24(2):84–90, February 1981. ISSN 0001-0782. doi: 10.1145/358549.358563. URL http://doi.acm.org/ 10.1145/358549.358563.

[8] David Chaum. Computer systems established, maintained and trusted by mutually suspicious groups /. Jan 1982.

[9] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, Advances in Cryptology, pages 199–203, Boston, MA, 1983. Springer US. ISBN 978- 1-4757-0602-4.

[10] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. J. Cryptol., 3(2):99–111, January 1991. ISSN 0933-2790. doi: 10. 1007/BF00196791. URL http://dx.doi.org/10.1007/BF00196791.

[11] Dave Bayer, Stuart Haber, and W. Scott Stornetta. Improving the efficiency and reliability of digital time-stamping. pages 329–334, 1993.

[12] Adam Back. Hashcash - a denial of service counter-measure. August 2002.

[13] Ethereum. Solidity. https://solidity.readthedocs.io/, 2019.

[14] Node.js. Express.js. https://expressjs.com/, 2018.

[15] Facebook. React. https://reactjs.org/, 2018.

[16] Web3. Web3. https://web3js.readthedocs.io/en/1.0/, 2018.

[17] Metamask. Metamask. https://metamask.io/, 2018.