

Verificación de Integridad de Datos

Miguel Angel Astor Romero

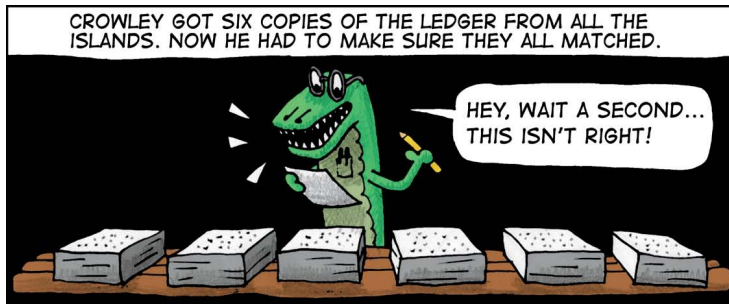
19 de julio de 2019

Agenda

- 1 Introducción
- 2 Mecanismos Básicos
- 3 Blockchain
- 4 Conclusiones

Introducción

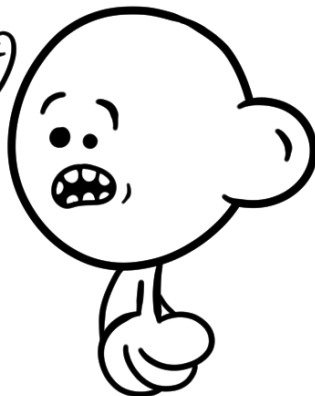
- La verificación de integridad de datos es el proceso mediante el cual se verifica que un bloque de datos transmitido por un medio no confiable no ha sido alterado en tránsito.
- Hay varias formas de resolver este problema, todas aplicando técnicas criptográficas.
- Las soluciones pueden aplicar técnicas centralizadas o descentralizadas.



verify the
checksum
before
running it!

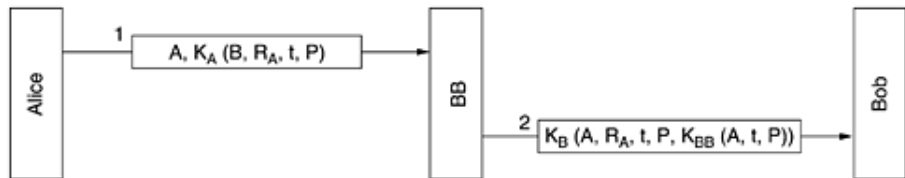


wget -O | bash



Una Solución Centralizada

- Dados Alice y Bob que desean comunicarse, se utiliza un tercero confiable (llamémoslo *Big Brother*) que comparte claves de cifrado simétrico con ambos.
- Alice envía el mensaje y una “firma” cifrada con su clave K_A a *Big Brother*.
- *Big Brother* verifica la firma y el mensaje y luego reenvía el mensaje más una firma a Bob cifrados con su clave K_B .
- La firma del segundo mensaje se genera con una clave que solo conoce *Big Brother*.



Sumas de Verificación o Compendios de Mensaje

Consiste en calcular un valor que identifique unívocamente un bloque de datos de entrada.

An SHA256 file containing checksums can be found in the same directory as the installation files. You can confirm that none of the downloaded files were mangled in transit using the [sha256\(1\)](#) command.

```
$ sha256 -C SHA256 miniroot*.fs  
(SHA256) minirootXX.fs: OK
```

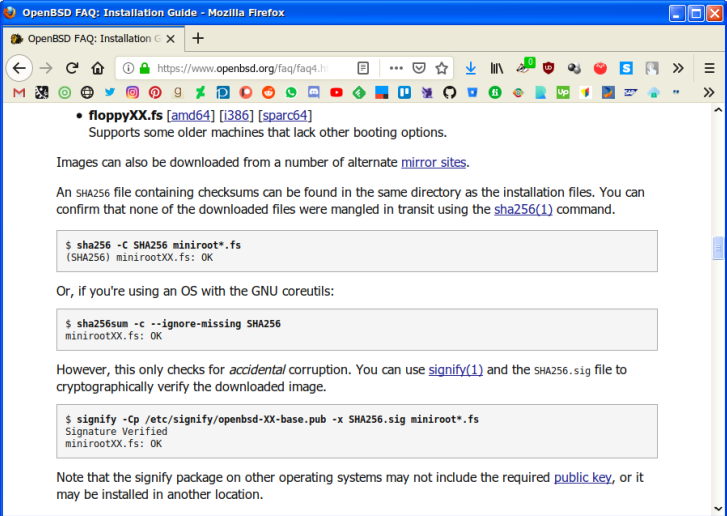
Or, if you're using an OS with the GNU coreutils:

```
$ sha256sum -c --ignore-missing SHA256  
minirootXX.fs: OK
```

Algoritmos de Suma de Verificación

- Chequeo de Redundancia Cíclica.
- Funciones *hash* (pe.: MD5, SHA256, etc.)

Uso de los Compendios de Mensaje en la Web



OpenBSD FAQ: Installation Guide - Mozilla Firefox

OpenBSD FAQ: Installation × +

<https://www.openbsd.org/faq/faq4.html>

- **floppyXX.fs** [\[amd64\]](#) [\[i386\]](#) [\[sparc64\]](#)
Supports some older machines that lack other booting options.

Images can also be downloaded from a number of alternate [mirror sites](#).

An SHA256 file containing checksums can be found in the same directory as the installation files. You can confirm that none of the downloaded files were mangled in transit using the [sha256\(1\)](#) command.

```
$ sha256 -C SHA256 miniroot*.fs  
(SHA256) minirootXX.fs: OK
```

Or, if you're using an OS with the GNU coreutils:

```
$ sha256sum -c --ignore-missing SHA256  
minirootXX.fs: OK
```

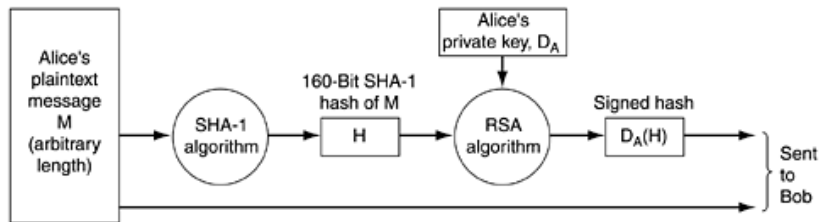
However, this only checks for *accidental* corruption. You can use [signify\(1\)](#) and the SHA256.sig file to cryptographically verify the downloaded image.

```
$ signify -Cp /etc/signify/openbsd-XX-base.pub -x SHA256.sig miniroot*.fs  
Signature Verified  
minirootXX.fs: OK
```

Note that the signify package on other operating systems may not include the required [public key](#), or it may be installed in another location.

Firmas Digitales

Los compendios de mensaje pueden verificar integridad pero son fáciles de falsificar. Para evitar esto se firman los compendios después de ser generados y antes de ser transmitidos.



Ataque del Cumpleaños

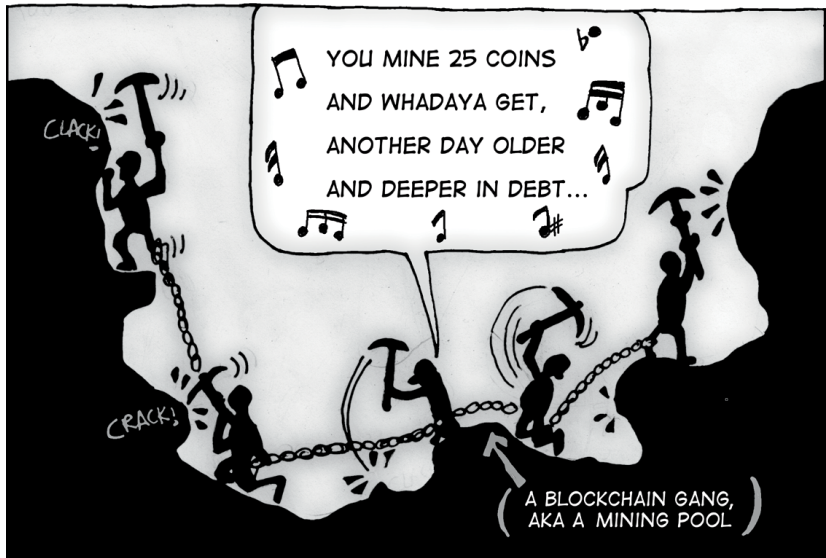
La verificación con firmas digitales no es 100 % segura

Si tenemos n entradas disponibles para una función *hash*

$f(x) = h \in H \subset \mathbb{Z}^+$ tal que $|H| = k$ y $n > \sqrt{k}$, entonces hay una buena probabilidad de encontrar un par de entradas n_1 y n_2 tales que $f(n_1) = f(n_2)$.

Revisar páginas 763 a 765 del libro “*Redes de Computadoras*” de Andrew Tanenbaum y el artículo “*How to Swindle Rabin*” de Gideon Yuval.





Antecedentes

- La idea de encadenar bloques de datos para verificación de integridad mediante criptografía data de (Ehrsam, 1978).
- Los algoritmos de Prueba-de-Trabajo datan de (Dwork, 1992) y (Back, 1997).
- En (Dai, 1998) se introduce el concepto de un sistema financiero semi-descentralizado donde los participantes son anónimos.
- El concepto de minería para verificación de transacciones y generación de valor fue desarrollado por (Szabo, 2005).

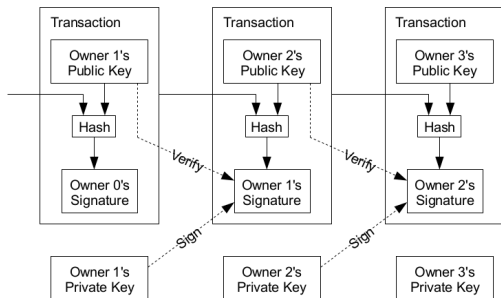
Bitcoin y los Orígenes de la Blockchain

- Primera criptomoneda.
- Creada en 2008 por Satoshi Nakamoto.
- Responde a la Gran Recesión económica del 2008.
- Asociada a las ideas del movimiento Cypherpunk.
- Especificada en (Nakamoto, 2008).



La Blockchain como Estructura de Datos

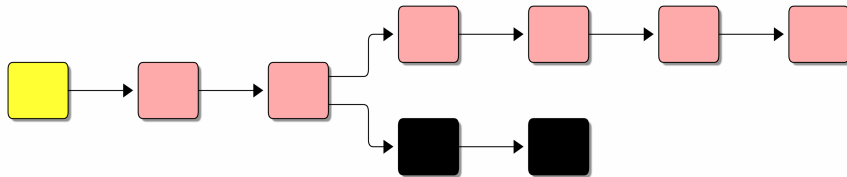
Dentro de una criptomoneda u otro sistema basado en Blockchain, la Blockchain en si cumple tres objetivos fundamentales:



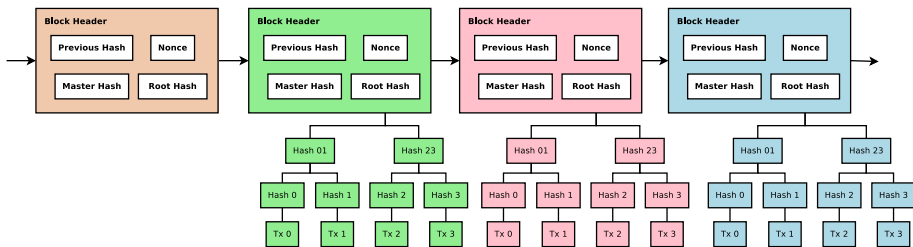
Funciones de la Blockchain

- 1 Coordinar el procesamiento de transacciones.
- 2 Asegurar su propia integridad.
- 3 Almacenar balances de los usuarios.

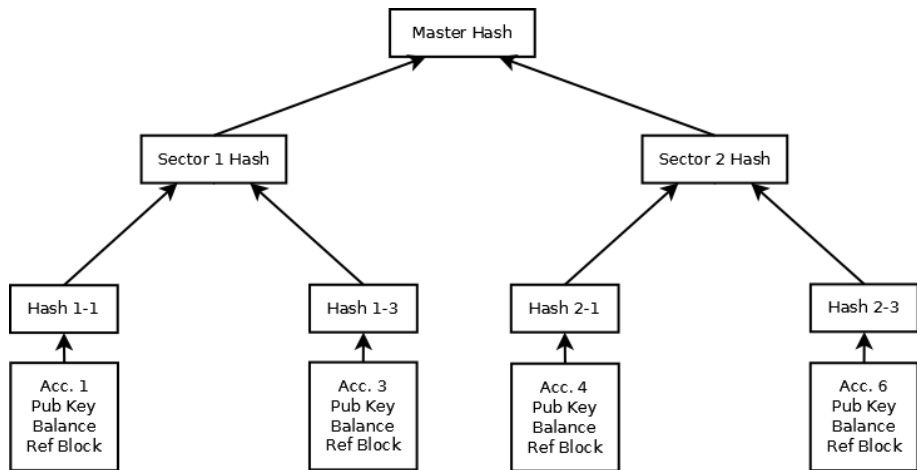
Cadena Principal y Cadenas Huérfanas



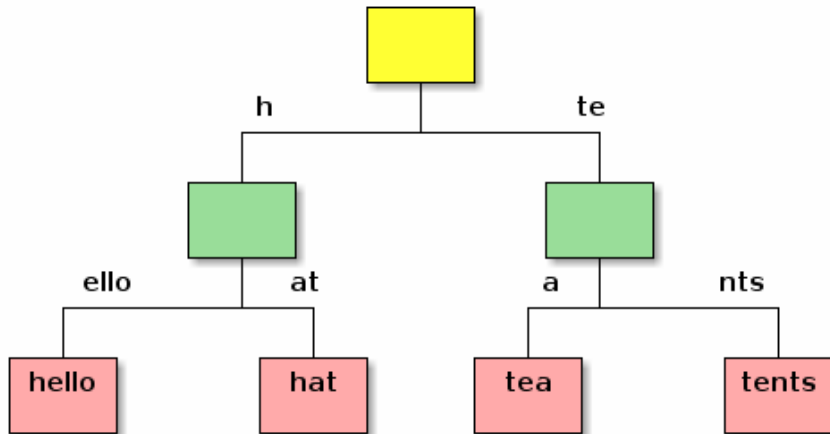
Funcionamiento de la Blockchain de Bitcoin



Árboles de Merkle

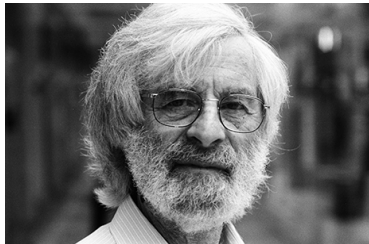


Árboles PATRICIA



Como Agregar Entradas a la Blockchain

- En todo el sistema debe existir una visión única de la Blockchain, pero no existe un ente central encargado de actualizarla.
- Esto define un problema de consenso distribuido llamado “problema de los generales bizantinos”, tipificado en (Lamport 1982).
- Se aplica un algoritmo de consenso distribuido. El algoritmo de Bitcoin se conoce como *Proof-of-Work*.

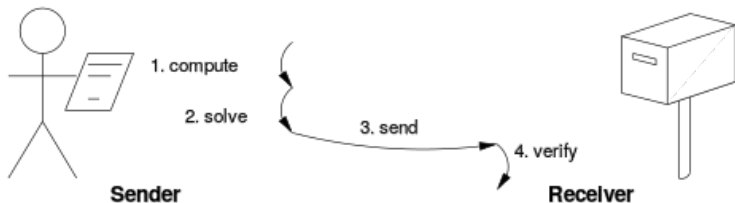


El Problema de los Generales Bizantinos

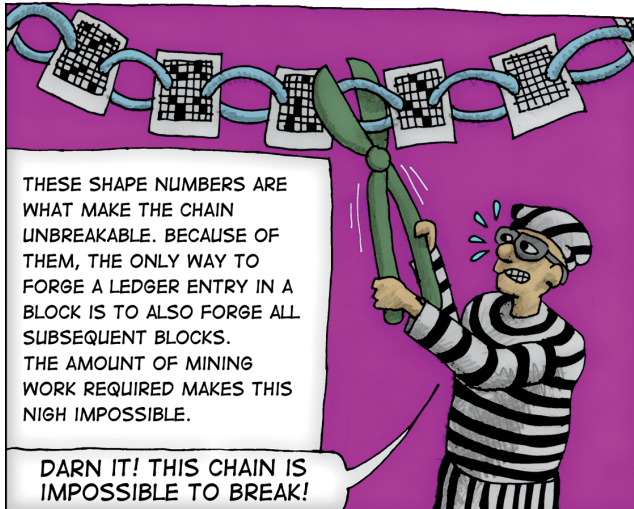


Minado por Prueba de Trabajo

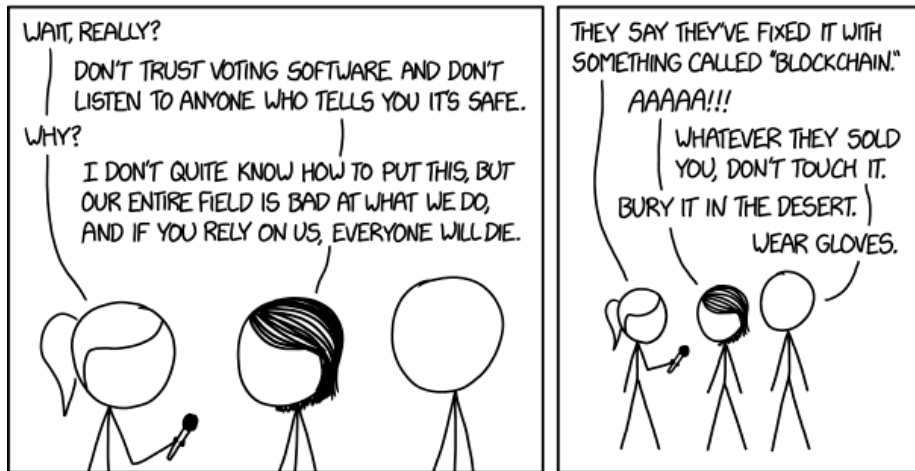
- 1 Los mineros reciben transacciones de la red y las acumulan en un bloque.
- 2 Al llenarse el bloque se le agrega un campo llamado *nonce* y se calcula su *hash* según el algoritmo de la red.
 - En Bitcoin el algoritmo es $SHA256(SHA256(bh))$, donde *bh* es la cabecera de un bloque.
- 3 Si el *hash* calculado es mayor que el parámetro de dificultad de la red se repite el proceso.

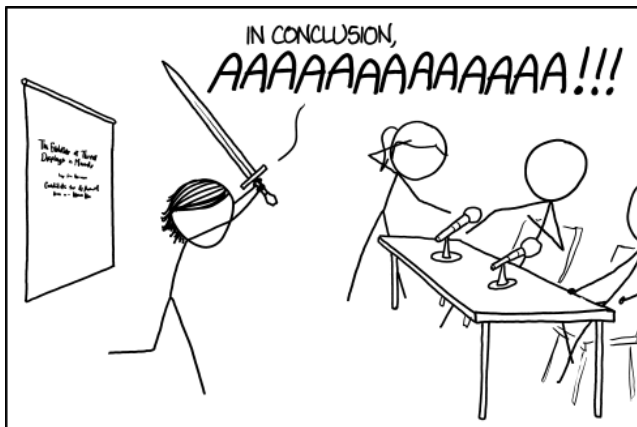


La Blockchain es Resistente a Fraudes . . .



... Pero no Resuelve Todos los Problemas.





THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

Conclusiones

- La verificación de integridad de datos se relaciona mucho con el problema de la autenticación y el problema del consenso distribuido.
- Es imposible tener un 100 % de confianza en la integridad de un bloque de datos.

Lectura Recomendada



Referencias

- Ehrsam, W. et al. (1978), “Message verification and transmission error detection by block chaining”.
- Lamport, L. et al. (1982), “The Byzantine generals problem”.
- Dwork, C. y Naor, M. (1992), “Pricing via processing or combatting junk mail”.
- Back, A. (1997). “Hashcash”, recuperado de:
<http://www.cypherspace.org/hashcash/>.
- Dai, W. (1998). “b-money”, recuperado de:
<http://www.weidai.com/bmoney.txt>.
- Szabo, N. (2005). “Bit Gold”, recuperado de:
<http://nakamotoinstitute.org/bit-gold/>.
- Nakamoto, S. (2008). “Bitcoin: A Peer-to-Peer Electronic Cash System”.

A Continuación

- Taller 3:
 - Certificados digitales con OpenSSL

¿Preguntas?

