Punteros Hash y Estructuras de Datos

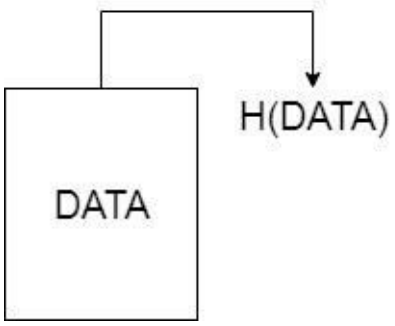
Introducción

Las estructuras de datos son básicamente técnicas para almacenar y/u ordenar datos. Cada una tiene ventajas/desventajas sobre otras y como se imaginan, es una cuestión del uso que se va tener de esa información la que determina la base de datos. Hay muchas estructuras de datos que pueden ser útiles en diversas situaciones, algunos ejemplos de estas son, una lista entrelazada, un árbol binario, una pila de datos, etc.

Un puntero hash es una estructura de datos en donde se almacena cierta información junto con un hash criptográfico de esa información. Detallaremos las estructuras más utilizadas en la mayoría de las criptomonedas.

Punteros Hash

Un puntero normal te brinda una manera de recuperar información de una estructura de datos, básicamente son coordenadas que nos indica la posición de una información. Un puntero hash además de indicar la posición de la información también te proporciona una forma de verificar que la información no haya sido alterada.



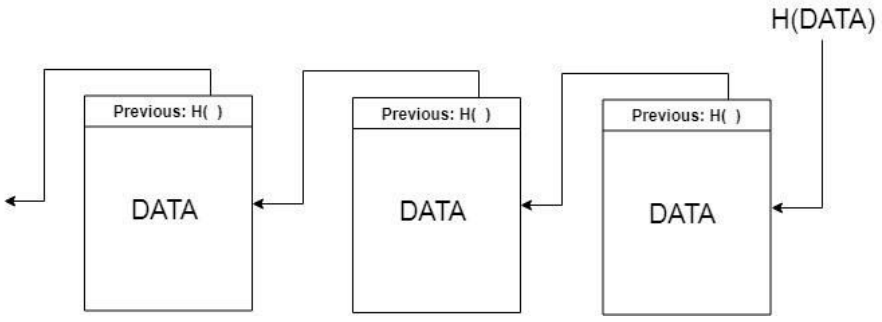
Un puntero hash es un puntero a donde los datos se almacenan junto con un hash criptográfico del valor de esos datos en algún punto fijo en el tiempo

En la sección anterior hablamos de las aplicaciones de las funciones hash y una de ellas era que funcionaban como resúmenes de mensajes o información. En el caso de Bitcoin, los punteros hash se utilizan como identificador de cada una de las transacciones y también como identificador los bloques.

Blockchain

Podemos usar punteros hash para construir todo tipo de estructuras de datos. Podemos tomar una estructura de datos como una lista entrelazada y utilizar los punteros hash como una forma de vinculación. Este tipo de estructura de datos se llama cadena de bloques o "blockchain", esta estructura tiene la propiedad de ser un registro en donde el intento de alteración de un registro anterior es evidente, siempre y cuando esta estructura de datos sea compartida por más de una computadora independientemente. Es decir, construimos una estructura de datos que puede almacenar una gran cantidad de datos y solamente nos permite

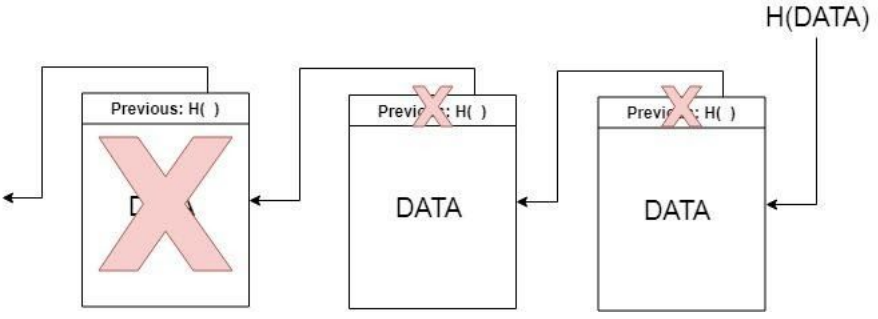
agregar datos al final del registro. Pero si alguien altera los datos que están antes en el registro, es posible de detectar.



Una cadena de bloques es una lista vinculada que se construye con punteros hash en lugar de punteros

Si un adversario intenta alterar los datos que se encuentran en el medio de la cadena tiene que hacerlo de tal manera que alguien que recuerde solo el puntero en la cabeza de la cadena de bloques no pueda detectar la alteración. Para lograr este objetivo, el adversario cambia los datos de algún bloque k. Dado que los datos han sido cambiados, el hash en el bloque k va a cambiar y por ende el hash del bloque k+1 también se va a haber alterado y no va a coincidir con lo que recordamos en nuestra estructura de datos.

Siempre que guardemos el puntero de hash del último registro agregado a la lista en un lugar donde el adversario no puede tener acceso, el adversario no podrá cambiar ningún bloque sin ser detectado.



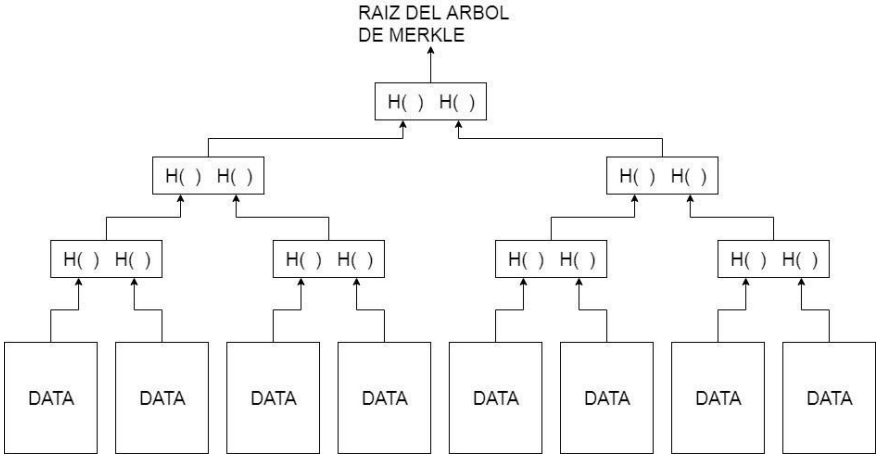
Una cadena de bloques es una lista vinculada que se construye con punteros hash en lugar de punteros

Esto hace que las cadenas de bloques que son públicas, de código abierto, neutrales, descentralizadas e incensurables tengan la propiedad de ser inmutables.

Blockchaindemo un simulador de blockchain en el cual se puede jugar a tener una base de datos con punteros hash donde podemos almacenar cualquier tipo de información, les propongo que creen sus propios registros y luego sus bloques. Por último, ¡intenten de cambiar un bloque más antiguo y observen que es lo que sucede!

Árboles de Merkle

Los árboles de Merkle son otra estructura de datos útil que podemos construir usando punteros hash, son un árbol binario pero construido con punteros hash y se conoce como árbol Merkle, gracias a su inventor Ralph Merkle. Supongamos que tenemos varios bloques que contienen datos. Estos bloques comprenden las hojas de nuestro árbol. Agrupamos estos bloques de datos en pares de dos, y luego para cada par, creamos una estructura de datos que tiene los punteros de hash, uno para cada uno de estos bloques. Estas estructuras de datos forman el siguiente nivel del árbol. Seguimos agrupando estos grupos en nuevos grupos de a dos hashes, y para cada par, se crea una nueva estructura de datos que contiene el hash del nivel inferior. Eventualmente se llega a un solo hash final que representa de toda la estructura, este último hash se llama raíz del Árbol de Merkle.



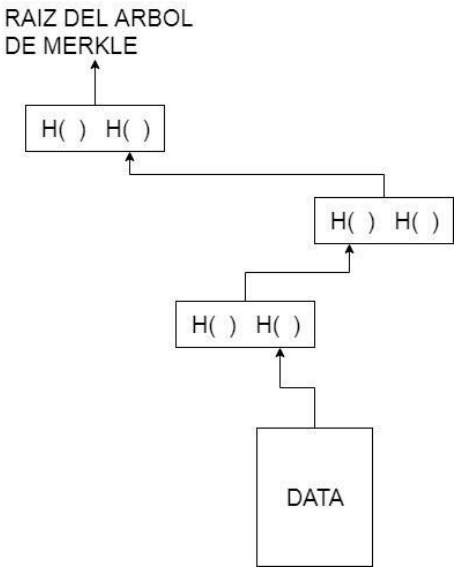
Los bloques de datos se agrupan en pares y el hash de cada uno de estos bloques se almacena en un nodo principal. Los nodos principales se agrupan a su vez en pares y sus hashes se

almacenan un nivel más arriba en el árbol. Esto continúa hasta el árbol hasta llegar a la raíz del árbol de Merkle.

Recordando solo el puntero hash de la parte superior del árbol (raíz del árbol de Merkle) nos permite asegurar que los datos no han sido alterados debido a que, como vimos con la cadena de bloques, si un adversario intenta manipular algún bloque de datos en la parte inferior del árbol, eso hará que el puntero que está un nivel arriba no coincida, el cambio se propagará a las capas superiores del árbol llegando a la raíz del árbol de Merkle. Si hemos almacenado la raíz del árbol en un lugar que un atacante no pueda acceder, podremos detectar el intento de manipulación de esta estructura.

Prueba de pertenencia

Además de la inalterabilidad, otra característica de los árboles de Merkle es que nos permite una prueba inobjetable de pertenencia o existencia a un árbol en particular. Supongamos que alguien quiere demostrar que un determinado bloque de información pertenece a un árbol Merkle en particular. Solamente con mostrarnos el bloque de datos original junto con a los hashes que se encuentran en la ruta desde el bloque de datos hasta la raíz del árbol de Merkle y se puede construir la prueba de pertenencia. Para nosotros, solo con recordar la raíz original nos es suficiente, podemos ignorar el resto del árbol, ya que los hashes en este camino son prueba suficiente válida para permitirnos verificar los hashes todo el camino hasta la raíz del árbol.



En las pruebas de existencia solo es necesario mostrar la información original (Data) y los hashes en la ruta desde la información hasta la raíz del árbol de Merkle.

¿Por qué son importantes?

Los árboles Merkle y las cadenas de bloques son piezas fundamentales en Bitcoin porque nos permiten construir pruebas de pertenencia inmutables en el tiempo. Además, gracias a cómo

se construyen estos árboles, es fácilmente computable que determinada información se ha incluido en un bloque en particular, y en qué orden.

También los árboles de Merkle nos permiten comprimir grandes conjuntos de datos eliminando todas las ramas superfluas y manteniendo solo las que necesitamos para establecer nuestra prueba. En el ecosistema de Bitcoin, criptomonedas & blockchain, los árboles de Merkle proporcionan las siguientes características fundamentales:

1. Posibilidad de verificar si una transacción está incluida en un bloque
2. Posibilidad de verificar el estado de la red
3. Clientes livianos (ya que no tenemos la necesidad de descargar toda la cadena)
4. Rendimiento general y escalabilidad
5. Verificación de pago simplificada (Simple Payment Verification, SPV)

Este artículo es una introducción a los fundamentos de cómo operan las estructuras de datos y los punteros hashes, una vez que comprendidos estos conceptos básicos, van a apreciar la seguridad y la confiabilidad de la Blockchain.