Introducción a blockchain

Pau Muñoz Pairet

Introducción

- **Blockchain es un tipo de estructura de datos distribuida usada para almacenar información sobre transacciones.**
- **Bitcoin es UNA de las muchas aplicaciones del concepto Blockchain, sin embargo resulta muy útil para entender el concepto.**

Ejemplo bitcoin

Sistema tradicional de transacciones económicas como analogía

- **Se confía en una entidad centralizadora.**
- **El dinero es un número.**
- **Se puede restar y sumar.**
- **Si la entidad cae, el sistema entero cae.**
- **La entidad puede cometer abusos.**
- **Sin embargo la entidad se encuentra muy protegida y supervisada.**
- **Es posible rastrear transacciones, identificar usuarios.**

Sistemas basados en “confianza”

- **Los sistemas bancarios tradicionales basan sus transacciones en la confianza hacia el banco.**
- **El banco conoce los datos del ambos clientes, ambos clientes conocen los datos del banco.**
- **A la hora del pago la conexión se cifra, normalmente usando TLS. Sistema de llaves pública/privada.**
- **El banco tiene un registro de todas las conexiones, las puede rastrear/monitorizar.**
- **Los paranoicos se preguntarán si pueden o no confiar en el banco.**

Alternativa P2P

- **Bicoin – mediante el uso de blockchain - busca ser una alternativa al sistema basado en confianza.**
- **Se substituye “el banco” o mejor dicho “la entidad central” por “la red”.**
- **Ninguna transacción pasa por una entidad central. Las transacciones se distribuyen mediante P2P a toda la red, que las valida mediante un algoritmo.**
- **Los partidarios dicen que así es más democrático.**

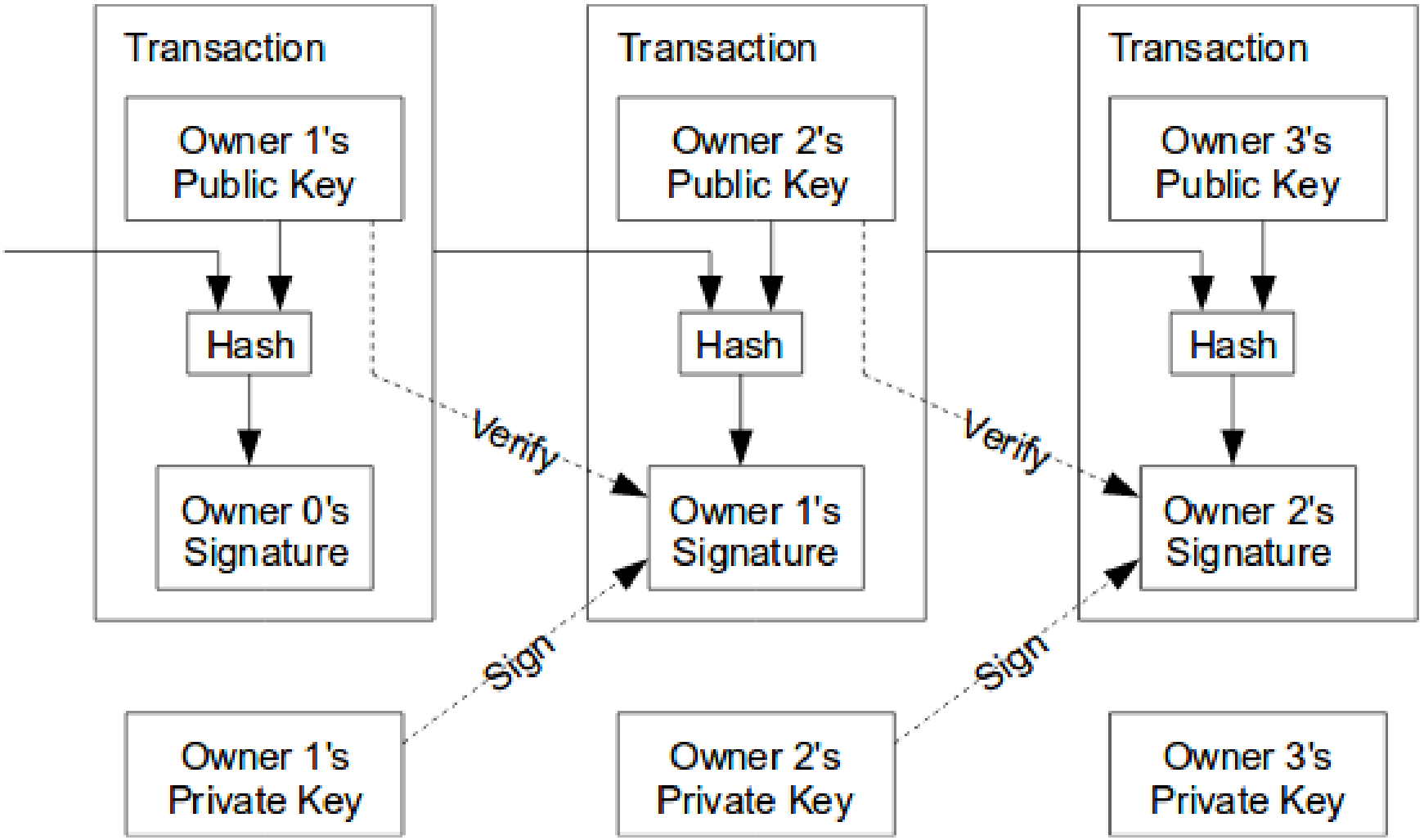
Problema del doble gasto

- **El dinero físico no se puede gastar dos veces. (o lo tengo yo o lo tienes tu). ¿Y el dinero “digital”?**
- **En sistemas tradicionales se confia en “entidades centralizadoras”**
- **Podemos definir Blockchain como un libro contable totalmente descentralizado (replicado en muchos ordenadores)**
- **Se usan técnicas de cifrado para comprobar quién y CUANDO ha hecho una transacción para evitar el doble gasto.**

Transacciones

- **Una moneda digital en bitcoin (o transacción en blockchain), es una cadena de firmas digitales.**
- **Cada propietario transfiere la moneda al siguiente mediante la firma de un HASH de la transacción previa y la llave pública del siguiente propietario, añadiéndo dicha información al final de la moneda.**
- **Esto facilita la verificación**

Transacciones en la cadena



Problema del doble gasto

- **Así pues, usando el sistema planteado ¿Como evitamos el problema del doble gasto?**
- **Una entidad central podría servir para verificar, pero no queremos eso.**
- **En bitcoin(BLOCKCHAIN) la ÚLTIMA transacción es la que cuenta. El dinero es visto como una serie de transacciones.**
- **Tu no “tienes” dinero, tu tienes el resultado de una serie de transacciones realizadas sobre un identificador. (por eso blockchain NO SOLO ES ÚTIL PARA PAGOS Y DINERO)**

Más transacciones en la cadena

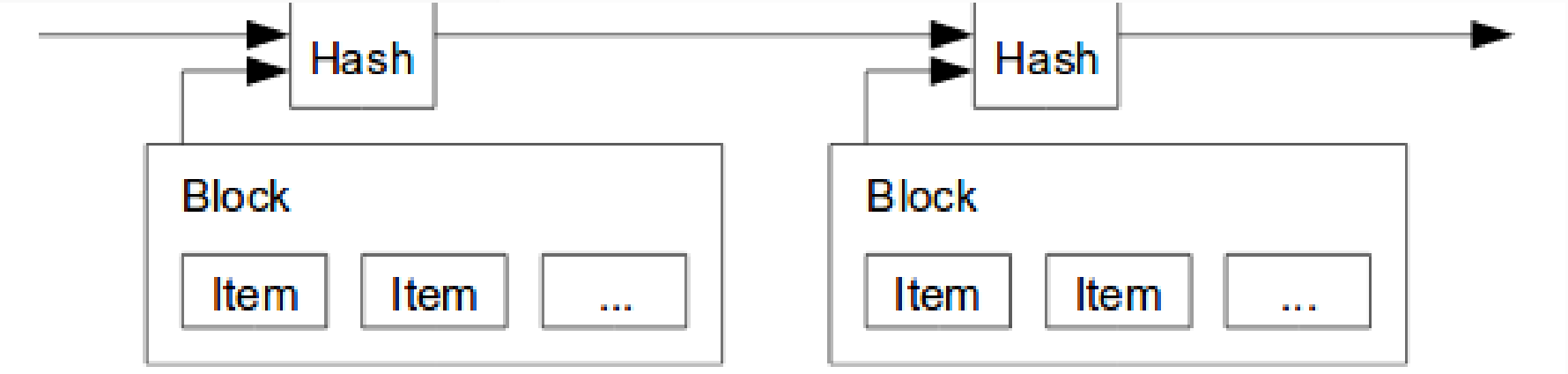
- **Tenemos que estar atentos a TODAS las transacciones de la cadena. ¿Como sabemos que transacción llega primero?**
- **Todas las transacciones son anunciadas públicamente y transferidas a la cadena con un SELLO DE TIEMPO.**
- **El receptor del dinero, necesita una PRUEBA de que A TAL HORA se realizó TAL TRANSACCIÓN y la MAYORÍA de nodos de la red estuvieron de acuerdo en la transacción.**

“proof of work” como alternativa

- **Un “proof of work” o prueba de trabajo, es un conjunto de información muy difícil de producir (en cuanto a tiempo) pero muy sencilla de verificar.**
- **Para verificar una transacción en bitcoin, las máquinas minadoras generan ataques de fuerza bruta para descubrir hashes SHA256 que simbolizan una transacción**
- **En bitcoin se usa el algoritmo “hashcash”.**

Servidor de sellado de tiempo “timestamp” (distribuido)

- **Un servidor de timestamp funciona generando un hash de un bloque de elementos a sellar y publicando el hash a la red.**
- **El sello de tiempo demuestra que la información había existido en un instante de tiempo, cada sello de tiempo incluye el anterior sello de tiempo formando una CADENA.**



Demostrar la veracidad de la información - Proof of work

- **Realizar el trabajo debe ser difícil, verificar el trabajo debe ser fácil. Realizar el trabajo = verificar una transacción y generar un nuevo bloque a la red.**
- **Para verificar una transacción, un “minero” genera un bloque con dicha transacción y lo añade a la cadena. La transacción se “sella” generando un hash con los datos del bloque actual + el hash de la transacción anterior (estructura merkle tree).**
- **En bitcoin cada hash (sha256) debe tener una cantidad de zeros delante.**

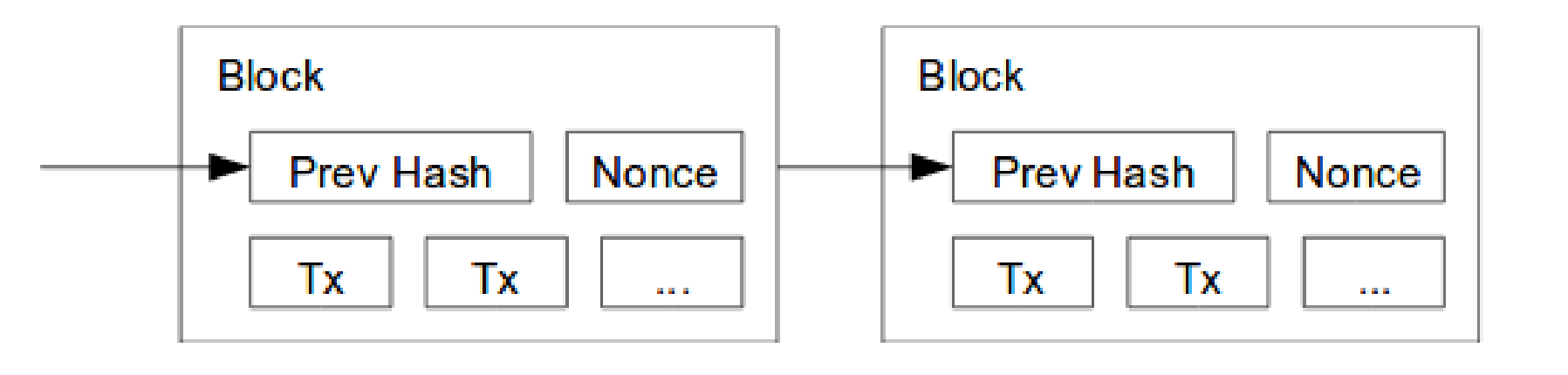
Un poco de minería

- **No hay manera de saber como será un hash antes de producirlo y si metemos otro dato en el mismo el hash será distinto.**
- **El minero utiliza un pedazo de información aleatorio en cada bloque llamado “nonce” para “jugar” con él y ir probando de generar un hash correcto.**
- **Se va “jugando con el nonce” hasta dar con un hash que tenga el número de 0’s requerido delante.**
- **El número de zeros en bitcoin se incrementa a medida que más “mineros” se unen a la red. Así aumenta la dificultad del minado.**

Un poco de minería

- **Puesto que verificar una transacción es muy simple.**
- **Una vez generado el hash correcto, los mineros dejan de minar, pues ya se ha generado el hash correcto, se ha verificado rápidamente y la transacción se considera “añadida a la cadena” y distribuída.**
- **Puesto que nos basamos el información firmada, verificad y con un sello de tiempo, tratar de modificar la cadena y sería inútil.**
- **NINGUNA TRANSACCIÓN REALIZADA EN LA RED BITCOIN PUEDE SER BORRADA**

Cadena de bloques, hash y nonce



Arquitectura de red Blockchain(Bitcoin)

- **La red dispone de nodos “origen” “hardcodeados” en cada nuevo cliente. Similar a DNS**
- **Nuevas transacciones son enviadas (broadcast) a todos los nodos.**
- **Cada nodo recoge transacciones en un bloque.**
- **Cada nodo trabaja buscando un “proof of work” difícil para el bloque generado**
- **Cuando un nodo encuentra un “proof of work” lo envía (broadcast) a toda la red**
- **Los nodos aceptan el nuevo bloque si pueden verificarlo**

Arqutiectura de la red Blockchain(bitcoin)

- **Los nodos de la red expresan su acuerdo dando validez al nuevo nodo usando su HASH para crear el nuevo bloque de la cadena.**
- **De este modo la transacción se considera aceptada y ya no podrá ser modificada.**
- **La estructura de datos funciona exactamente igual que una cadena irrompible.**
- **Se habla de criptomonedas debido al uso de sistemas de cifrado para mantener la integridad y la veracidad de la red.**

Arqutiectura de la red Blockchain(bitcoin)

- **Una nueva transacción no necesita ser aprobada por TODOS los nodos, necesita ser aprobada por la mayoría.**
- **Si un nodo no recibe un bloque determinado y sin embargo recibe el siguiente (¿Algo no cuadra aquí?) solicitará las partes anteriores de la cadena.**

Incentivos ¿De donde diablos sale tanto dinero?

- **En bitcoin, tradicionalmente la primera transacción de un bloque es especial y ofrece un bitcoin al creador del bloque.**
- **Si el valor del “dinero” que se emite en una transacción es superior al valor del “dinero” que va a recibir la otra parte, la diferencia es la “tasa de minado” que va directa al responsAble de la generación del bloque. Esto sirve de incentivo a mineros para verificar y acelerar transacciones.**
- **Cuanto más moneda hay en circulación mayor es el incentivo para los mineros.**

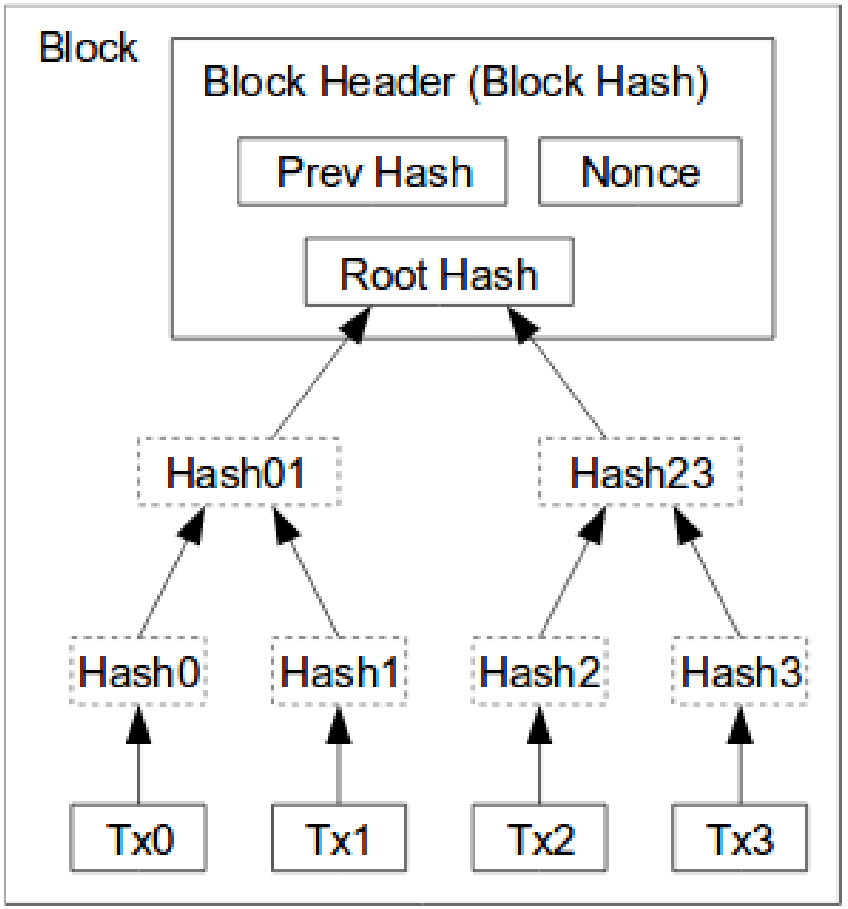
Incentivos ¿De donde diablos sale tanto dinero?

- **Si un atacante controla la mayor parte de potencia de CPU de la red, este debería poder “cascar” la red a base de bien.**
- **Sin embargo las ganancias que obtendría “minando” serían mucho mayores que las ganancias que obtendría “robando” el dinero de las transacciones.**

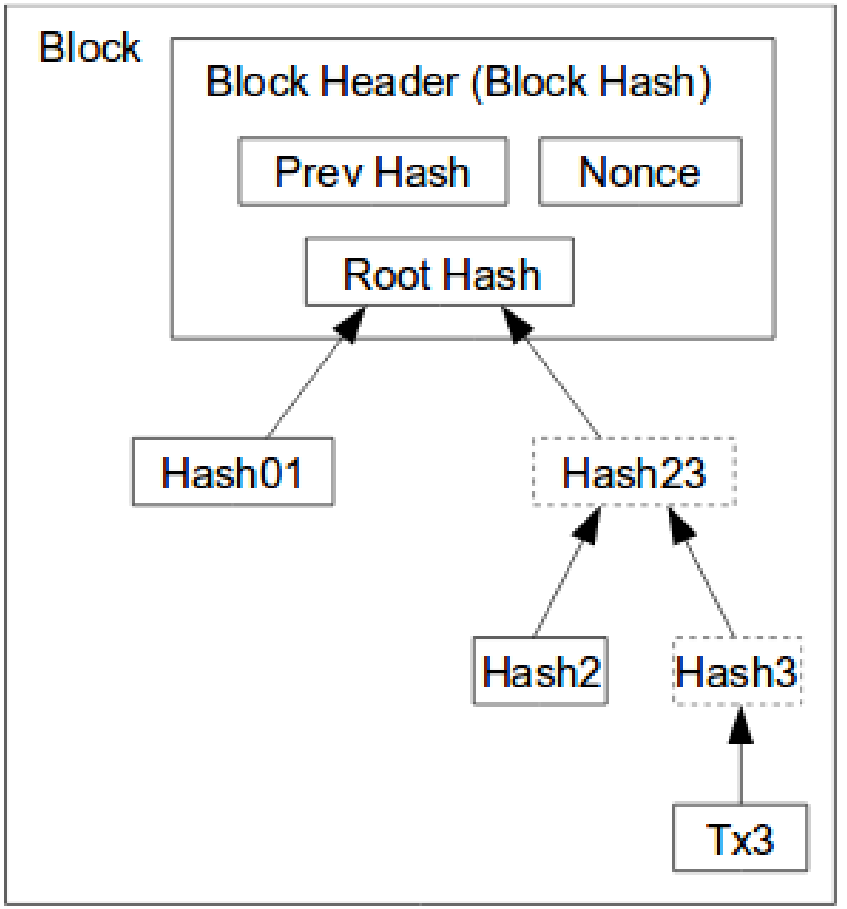
Espacio en disco

- **Para ahorrar espacio, podemos quedarnos con las últimas transacciones.**
- **Para no destrozar la cadena, las transacciones se guardan en una estructura de datos en forma de árbol MERKLE TREE.**
- **Podemos podar el árbol y quedanos con las ramas interesantes para ahorrar espacio.**

Merkle tree y bitcoin



Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

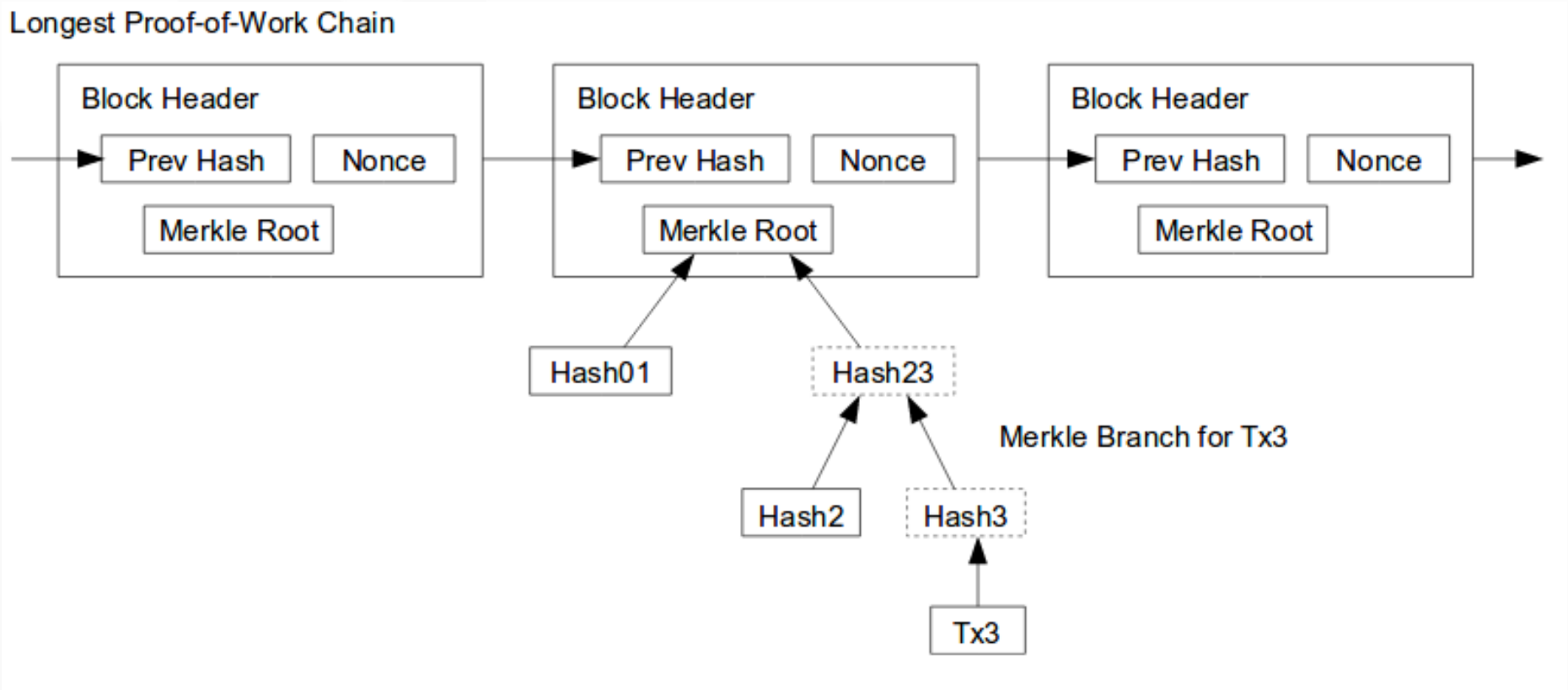
Espacio en disco

- **La cabecera de un bloque ocupa 80 bytes. Generando bloques cada 10 min, 80bytes * 6 * 24 * 365 = 4.2MB por año.**
- **Técnicamente (ley de moore) no deberíamos tener problemas para almacenar tanto dato, sin embargo las transacciones en bitcoin son cada vez más numerosas y lentas y eso es un hecho.**

Verificación ligera de transacciones

- **Un usuario solo necesita disponer de una copia de las cabeceras de bloque de la cadena de “proof of work” más larga.**
- **Puede disponer de dicha cadena “preguntando” a todos los nodos que pueda hasta estar convencido de que tiene la mayor cadena.**
- **Luego puede buscar la rama en el árbol merkle indicada y utilizar el algoritmo de hash para comprobar la veracidad**

Verificación ligera de transacciones

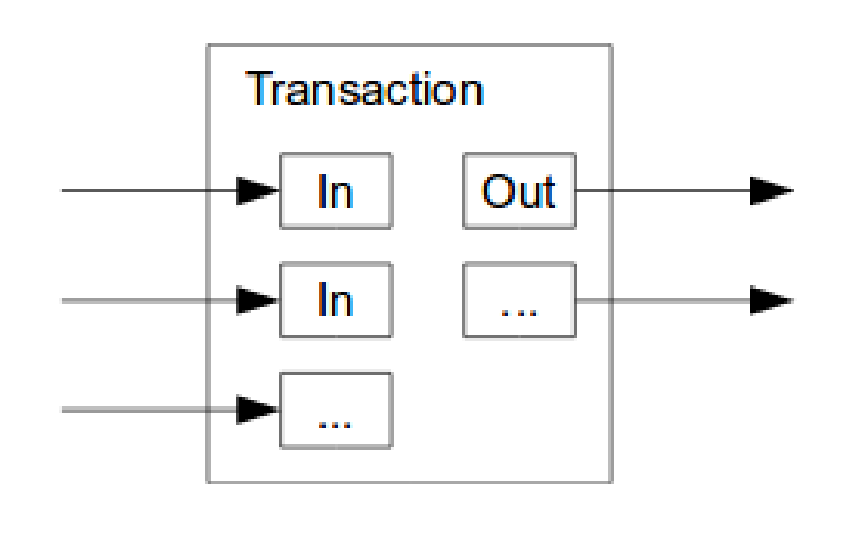


Verificación ligera de transacciones

- **Si la mayoría de nodos de la red son “legales” todo OK. Sin embargo si un atacante controla la mayoría de nodos de la red “te la pueden colar”.**
- **Un mecanismo de protección ante un ataque sería permitir que los nodos enviaran “alertas” en forma de broadcast, una vez detectaran una falsificación, obligando a los “clientes ligeros” a descargar TODA la cadena de bloques para realizar una verificación completa.**

Transacciones agregadas

- **En la red blochcain una misma transacción puede contener varias entradas y varias salidas.**



Privacidad

- **En la red bitcoin(blockchain) las llaves públicas se mantienen en privado.**
- **El público puede ver que una entidad A está enviado una cantidad de dinero a una entidad B pero no se revela información ninguna sobre A y B (nada enlaza a ellos)**
- **Por cada transacción se usa un par distinto de llaves.**
- **Se pueden llegar a usar “meta-datos” para “seguir la pista” de transacciones concretas (transacciones multi input)**

Seguridad

- **Un atacante puede intentar generar una “cadena alternativa” más rápido que la cadena “honesta”. Si controla suficiente poder de cálculo, podría.**
- **Sin embargo, solo podría recuperar su dinero gastado, como mucho. No podría generar dinero “del aire”. Los demás nodos no aceptarían transacciones no válidas.**
- **Una competición entre un atacante y la cadena honesta se puede modelar como un “paseo aleatorio binomial”.**

Binomial random walk

- **El evento de éxito sería que la cadena honesta consiguiera generar un nuevo bloque superando por uno a la cadena falsa.**
- **El fracaso sería que el atacante superara por un bloque a la cadena honesta.**
- **Modelo "Gambler's ruin": https://en.wikipedia.org/wiki/Gambler%27s_ruin**

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

Más matemáticas

- **A medida que la distancia entre el atacante y la cadena honesta sube la probabilidad de éxito del atacante baja drásticamente.**
- **Un atacante podría querer hacer creer al receptor que este ha cobrado, solo por un tiempo, luego transferir el dinero a él mismo modificando la cadena. El receptor sería alertado, sin embargo el atacante confía en que será “demasiado tarde” como para poder hacer algo (la cadena falsa ya estaría en circulación).**

Más matemáticas

- **El receptor genera un nuevo par de claves y envía la llave pública al emisor segundos antes de la firma.**
- **Esto evita que el atacante pueda trabajar “en secreto” generando una cadena falsa.**
- **Una vez la transacción se encuentra enviada, un emisor deshonesto podría empezar a trabajar “en secreto” en una cadena “paralela” con una versión “alternativa” de la transacción.**
- **El receptor esperará a que la transacción sea añadida a un bloque y luego “z bloques” sean enlazados a la cadena a partir de ahí.**

Más matemáticas

- **No se puede saber el progreso realizado por el atacante en su ataque. Sin embargo, suponiendo que los bloques honestos se tomen el tiempo esperado por bloque podemos definir el progreso del atacante mediante una distribución de Poisson.**

$$\lambda = z \frac{q}{p}$$

Más matemáticas

- **Para conseguir ver la probabilidad de que el atacante pueda llegar a lograr sus objetivos superando la cadena honesta. Multiplicamos la densidad de Poisson por cada una de la suma de progreso que pueda haber hecho por la probabilidad de que pueda haber llegado a “pillar” la cadena honesta desde ese mismo punto**

$$\sum_{k=0}^{\infty} \frac{\lambda^{k} e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & if\ k \leq z \\ 1 & if\ k > z \end{cases}$$

Resultados

- **Podemos expresar la suma así para evitar sumar la cola infinita de la distribución**

$$1-\sum_{k=0}^{z}\frac{\lambda^{k}e^{-\lambda}}{k!}\left(1-(q/p)^{(z-k)}\right)$$

- **Realizando una simulación podemos ver que no es nada fácil realizar un ataque a la red blockchain.**

Estructura de datos de bitcoin blockchain

- http://codesuppository.blogspot.com.es/2014/01/how-to-parse-bitcoin-blockchain.html

- **Todos los campos.**
- **159.999 MB**
- https://blockchain.info/es/charts/blocks-size
- **Todas las transacciones en blockchain.info**

Alternativas a “proof of work”

- **Proof of stake:** https://en.bitcoin.it/wiki/Proof_of_Stake
- **Proof of burn:** https://en.bitcoin.it/wiki/Proof_of_burn

Blockchains famosas

- **https://bitcoin.com**
- **https://www.ethereum.org/ (smart-contract)**
- **https://litecoin.com/es/**

Usos del bitcoin

- **Especulación.**
- **Evasión de capital y blanqueo.**
- **Financiación de grupos terroristas y actividades ilegales.**
- **Apoyo a operaciones clandestinas.**
- **Soporte a actividades de activismo político.**
- **En menor medida, uso legítimo, compra y venta de servicios y productos.**

Usos/posibilidades de blockchain

- **Transacciones económicas.**
- **Contratos digitales.**
- **Voto electrónico.**
- **Auto regulación de mercados/sistemas. (nonces y proof of work)**
- **Sistemas inter-gubernamentales para compartir información sobre seguridad mediante cadenas privadas. (anonimato Y certificación)**
- **Muchos otros**

Referencias

- https://bitcoin.org/bitcoin.pdf
- https://es.wikipedia.org/wiki/Hashcash
- https://blockchain.info/
- https://es.wikipedia.org/wiki/SHA-2
- https://es.wikipedia.org/wiki/Sistema_de_prueba_de_trabajo
- https://en.wikipedia.org/wiki/Trusted_timestamping
- https://en.wikipedia.org/wiki/Byzantine_fault_tolerance
- https://en.wikipedia.org/wiki/Merkle_tree
- https://es.wikipedia.org/wiki/Cadena_de_bloques