Resumo artigo original do Bitcoin:
"Bitcoin: A peer-to-peer electronic cash system"

Lucas João Martins

Na época do artigo o comércio na internet estava muito dependente de maneira exclusiva de instituições financeiras que serviam de "trusted third party (TTP)" para o pagamento eletrônico. Com isso, havia um aumento no custo das transações. Além disso, isso era uma oportunidade, já que não existia mecanismo que possibilitasse o pagamento sem a necessidade de um TTP.

Era necessário um sistema de pagamento eletrônico baseado em prova criptográfica ao inves do TTP. Onde a solução proposta resolvia o problema de duplo pagamento, que é quando se usa a mesma unidade da moeda para realizar dois pagamentos diferentes, através de um servidor peer-to-peer distribuído que utiliza do timestamp para gerar prova computacional da ordem cronológica das transações. O sistema será seguro enquanto os nodos participantes honestos controlarem coletivamente o maior poder de CPU.

Uma moeda eletrônica é definida como uma corrente de assinaturas digitais. Cada proprietário transfere a moeda para o próximo assinando um hash da transação anterior e a chave pública do próximo proprietário, onde tudo isso é adicionado no fim da moeda. Qualquer interessado pode verificar as assinaturas.

O problema é que o interessado não consegue verificar se alguém realizou duplo pagamento. A solução é que as transações precisam ser anunciadas publicamente e os participantes do sistema precisam concordar com isso. Logo, a corrente com o maior número de concordância será considerada como a primeira recebida.

A solução do servidor timestamp pega um hash do bloco de itens que receberá o timestamp e divulga para todos. Cada timestamp possui o timestamp anterior ao seu hash.

O servidor timestamp é implementado em uma rede peer-to-peer. Onde o "proof-of-work" é um incremento de um "nonce" no bloco até chegar em um valor que fornece ao hash de bloco os bits zeros solicitados. Um bloco não pode ser mudado sem que o trabalho seja refeito. Portanto, se algum bloco for encadeado posteriormente, então significa que para mudar o primeiro será preciso retrabalhar nesses blocos adicionados.

Então a "proof-of-work" é basicamente: um CPU, um voto. A decisão da maioria é representada pela maior corrente, que vai possuir a maior "proof-of-work" relaizada. Com isso, a probabilidade de um ataque ter sucesso diminui exponencialmente com a adição de blocos subsequentes em uma corrente.

Na rede os nodos sempre irão considerar a corrente mais longa como sendo a correta e continuarão seu trabalho nela. Além disso, não é preciso que o "broadcast" de uma nova transação atinja todos os nodos.

Por convenção, a primeira transação em um bloco é uma transação especial que começa uma nova moeda pertencente ao criador do bloco. Isso é um incentivo para os nodos participarem da rede e um modo de colocar moedas em circulação, já que não há uma autoridade central para realizar isso. Esse cenário é análogo a mineradores gastando recursos para colocar ouro em circulação. Esse incentivo ajuda a manter os nodos honestos.

Para economizar espaço em disco e não corromper o hash de um bloco, as trasação são "hashed" em uma árvore Merkle. Aliás, todo o processo descrito até agora pode ser implementado de maneira mais simples, sem possuir uma rede inteira de nodos, e, só manter uma cópia do último bloco. Porém, haverá riscos de erros. Um negócio que recebe pagamentos constantemente provavelmente não irá optar por essa solução.

As transações podem conter múltiplas entradas e saídas. Normalmente será uma única entrada resultante de uma grande transação anterior ou múltiplas pequenas entradas combinada, e, no máximo duas saídas, o troco quando necessário e o pagamento em si. A dependência entre todas as transações não é um problema aqui: não é preciso extrair uma cópia do histórico de transação completo.

No âmbito da privacidade, a necessidade de ter que anunciar todas as transações não limita o que pessoas externas conseguem saber sobre uma transação, mas a privacidade consegue ser mantida da seguinte forma: chaves públicas são anônimas. Outra forma de ajudar a privacidade é fazer com que para cada transação um novo par de chaves tenha que ser gerado, assim dificulta associar determinada chave com determinada pessoa.

Sobre a segurança, nenhum nodo vai aceitar uma transação inválida como pagamento, além do que nodos honestos nunca vão aceitar um bloco que contenha esse tipo de transação. Um atacante só pode tentar mudar sua própria transação para tentar pegar o dinheiro que gastou nela.

A corrida entre uma corrente honesta e uma corrente do atacante pode ser caracterizada como uma "Binomial Random Walk". A probabilidade da corrente do atacante ter sucesso pode ser expressa como:

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

Onde:

- p = probabilidade de um nodo honesto encontrar o próximo bloco
- q = probabilidade de um atacante encontrar o próximo bloco
- q_z = probabilidade que um atacante vai ter se tiver z blocos atrás

Assumimos que p > q, então a probabilidade cai exponencialmente a medida que o número de blocos aumenta. Então a melhor chance do atacante é no começo. Aliás, o potencial de sucesso de um atacante conseguir mudar uma transação que ele acabou de realizar pode ser expresso como uma distribuição de Poisson.

Com isso, o bitcoin, modelo proposto aqui, é um framework de moedas criadas a partir de assinaturas digitais que não possui o problema de duplo pagamento.

Referência

NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.