Underlying Assumptions

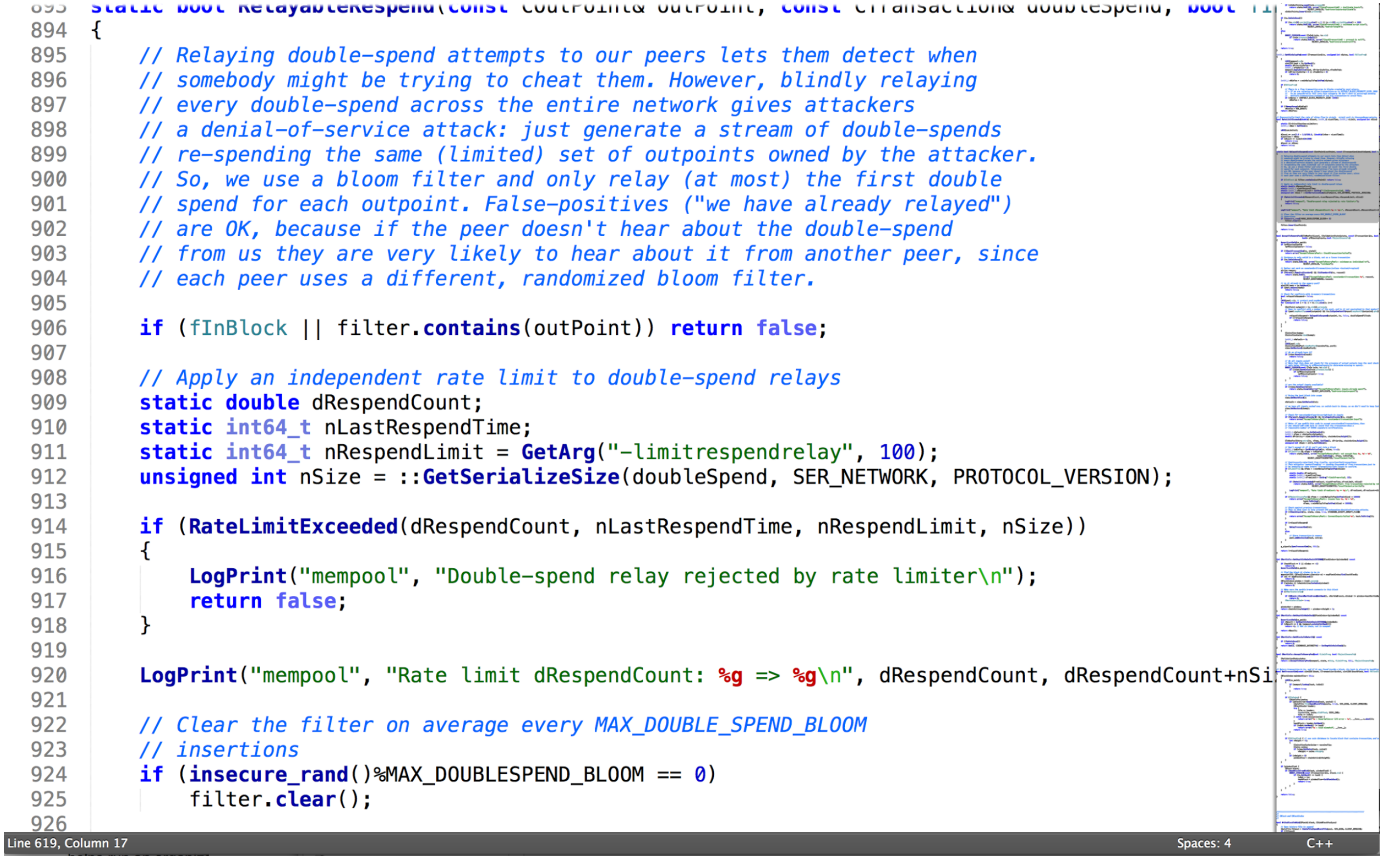
1. **Defining definition**
    a. To demarcate
    b. Minimax occam
2. **Defining Entrepreneur**
    a. The ability to anticipate demand
    b. Well performing entrepreneurship is recognised
3. Defining Business
    3.1. A sequence of activities

Defining Profit
    Progress

Risk is distributed

In short: whose ideas get to be brought in to the material world?
    Who gets the land, the money, the food, the security

The internet is an interconnected network
    It networks networks



```
894 {
895     // Relaying double-spend attempts to our peers lets them detect when
896     // somebody might be trying to cheat them. However, blindly relaying
897     // every double-spend across the entire network gives attackers
898     // a denial-of-service attack: just generate a stream of double-spends
899     // re-spending the same (limited) set of outpoints owned by the attacker.
900     // So, we use a bloom filter and only relay (at most) the first double
901     // spend for each outpoint. False-positives ("we have already relayed")
902     // are OK, because if the peer doesn't hear about the double-spend
903     // from us they are very likely to hear about it from another peer, since
904     // each peer uses a different, randomized bloom filter.
905
906     if (fInBlock || filter.contains(outPoint)) return false;
907
908     // Apply an independent rate limit to double-spend relays
909     static double dRespendCount;
910     static int64_t nLastRespendTime;
911     static int64_t nRespendLimit = GetArg("-limitrespendrelay", 100);
912     unsigned int nSize = ::GetSerializeSize(doubleSpend, SER_NETWORK, PROTOCOL_VERSION);
913
914     if (RateLimitExceeded(dRespendCount, nLastRespendTime, nRespendLimit, nSize))
915     {
916         LogPrint("mempool", "Double-spend relay rejected by rate limiter\n");
917         return false;
918     }
919
920     LogPrint("mempool", "Rate limit dRespendCount: %g => %g\n", dRespendCount, dRespendCount+nSi
921
922     // Clear the filter on average every MAX_DOUBLE_SPEND_BLOOM
923     // insertions
924     if (insecure_rand()%MAX_DOUBLESPEND_BLOOM == 0)
925         filter.clear();
926
```

Line 619, Column 17 Spaces: 4 C++

The regulation is in the code and the community

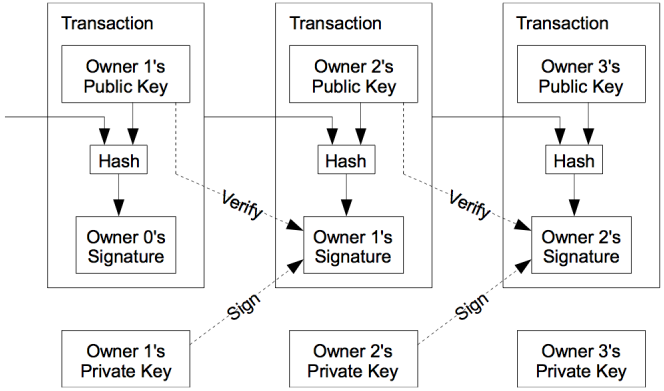
Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must



The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

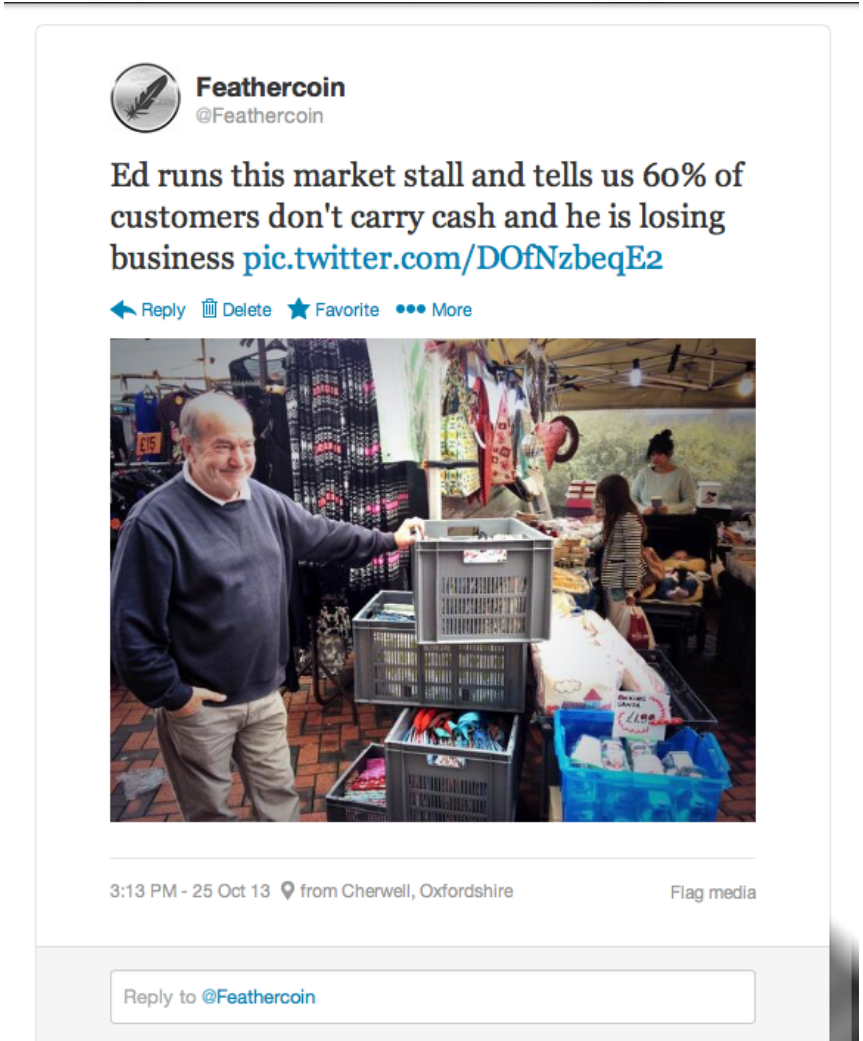
We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

3. Timestamp Server

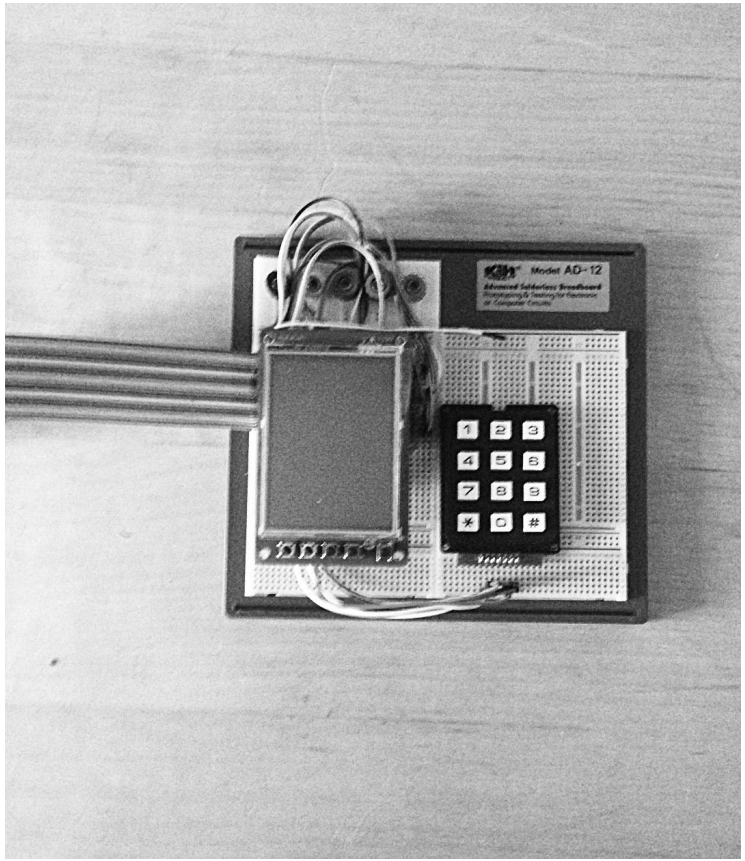
The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

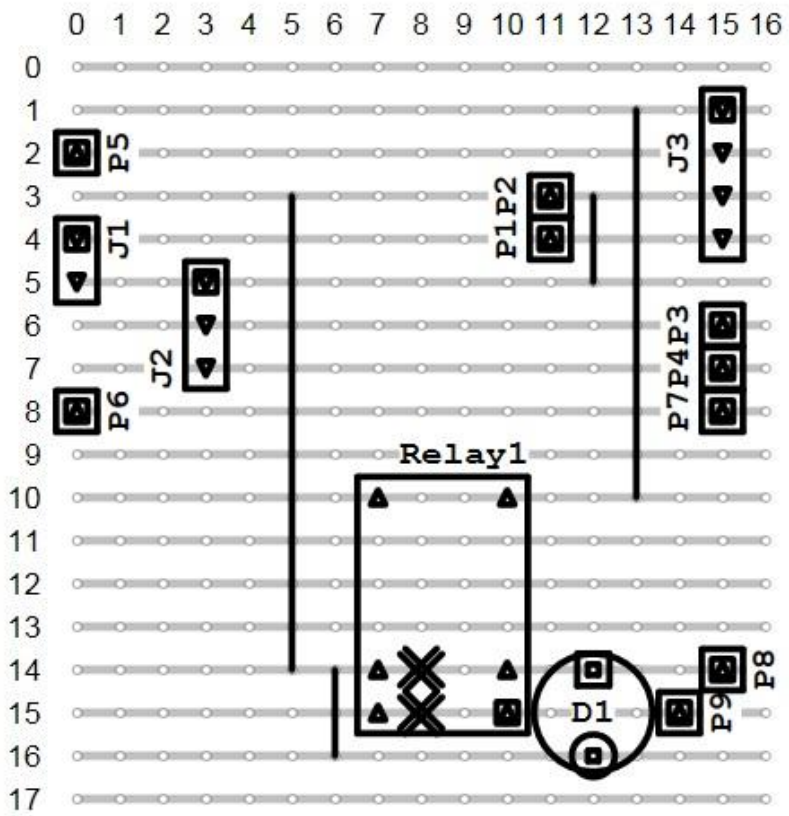
The truth is it’s not about the price or the code

It’s really about the people



Feathercoin Point of Sale





Hardware is Open Source