Market Watch with RMH – Crypto Currencies (Bitcoin and others)

I wanted to thank all of you who asked over the years about gold as a safe haven, and now Bitcoin (Crypto Currencies). The words Bitcoin (BC) and Crypto Currencies (CC) will be used interchangeably.

First a little history, on October 31, 2008 **the name Satoshi Nakamoto (he has never been identified)** published a paper on a cryptography mailing list at metzdowd.com, describing a digital currency cryptocurrency, titled **"Bitcoin: A Peer to Peer Electronic Cash System".** On January 9, 2009 he released version 0.1 of the bitcoin software on Sourceforge, and embedded in the coinbase transaction was the following text: **"`The Times 03/Jan/2009 Chancellor on brink of second bailout for banks`",[17] referring to a headline in the UK newspaper *The Times* published on that date.[18] This note has been interpreted as both a timestamp of the genesis date and a derisive comment on the instability caused by fractional-reserve banking. The original paper can be found in Appendix 1 at the end of RMH Market Watch**

To put everyone on the same page, a few crucial definitions are in order, all come from www.bitcoin.org :

What is Bitcoin?

Bitcoin is a consensus network that enables a new payment system and a completely digital money. It is the first decentralized peer-to-peer payment network that is powered by its users with no central authority or middlemen. From a user perspective, Bitcoin is pretty much like cash for the Internet. Bitcoin can also be seen as the most prominent triple entry bookkeeping system in existence.

Who created Bitcoin?

Bitcoin is the first implementation of a concept called "cryptocurrency", which was first described in 1998 by Wei Dai on the cypherpunks mailing list, suggesting the idea of a new form of money that uses cryptography to control its creation and transactions, rather than a central authority. The first Bitcoin specification and proof of concept was published in 2009 in a cryptography mailing list by Satoshi Nakamoto. Satoshi left the project in late 2010 without revealing much about himself. The community has since grown exponentially with many developers working on Bitcoin.

Bitcoin, I freely admit to not knowing much about until recently, when I started to see published research from providers I trust, and the research process was started.

What caught my attention was the number of sophisticated investors that were now starting to dedicate a small percentage of the assets they managed to own bitcoin and other crypto assets as an uncorrelated asset with respect to stocks, bonds and alternative assets. **This might lead to the creation of a new asset class?**

On January 4, of this year, the Office of the Comptroller of the Currency issued a letter approving U.S. Banks to use public blockchain networks. The letter said these financial institutions can participate as nodes on a blockchain and store or validate payments.

I did a quick search (https://99bitcoins.com/bitcoin/who-accepts/) to see who accepts bitcoins as a form of payment. The list below is nowhere near complete, just a sample.

Wikipedia	Microsoft	AT&T
Burger King	Overstock	PayPal
Square	Miami Dolphins	Dallas Mavericks
Virgin Galactic	Norwegian Air	CheapAir

Initially one of my biggest concerns had to do with the custodial nature of Bitcoin. Where was it held, who monitored and controlled it, what were the rules, how was it created? All of these questions had to be answered. Schwab is the custodian of our firm for a reason, they are the leader in the Registered Investment Advisor (RIA) space, custody of $3T for RIA's and $3T for retail clients. The following comes from the web site www.bitcoin.org as to who controls the Bitcoin network and how it works:

Who controls the Bitcoin network?

Nobody owns the Bitcoin network much like no one owns the technology behind email. Bitcoin is controlled by all Bitcoin users around the world. While developers are improving the software, they can't force a change in the Bitcoin protocol because all users are free to choose what software and version they use. In order to stay compatible with each other, all users need to use software complying with the same rules. Bitcoin can only work correctly with a complete consensus among all users. Therefore, all users and developers have a strong incentive to protect this consensus.

How does Bitcoin work?

From a user perspective, Bitcoin is nothing more than a mobile app or computer program that provides a personal Bitcoin wallet and allows a user to send and receive bitcoins with them. This is how Bitcoin works for most users.

Behind the scenes, the Bitcoin network is sharing a public ledger called the "block chain". This ledger contains every transaction ever processed, allowing a user's computer to verify the validity of each transaction. The authenticity of each transaction is protected by digital signatures corresponding to the sending addresses, allowing all users to have full control over sending bitcoins from their own Bitcoin addresses. In addition, anyone can process transactions using the computing power of specialized hardware and earn a reward in bitcoins for this service. This is often called "mining". To learn more about Bitcoin, you can consult the dedicated page and the original paper.

Why do people trust Bitcoin?

Much of the trust in Bitcoin comes from the fact that it requires no trust at all. Bitcoin is fully open-source and decentralized. This means that anyone has access to the entire source code at any time. Any developer in the world can therefore verify exactly how Bitcoin works. All transactions and bitcoins issued into existence can be transparently consulted in real-time by anyone. All payments can be made without reliance on a third party and the whole system is protected by heavily peer-reviewed cryptographic algorithms like those used for online banking. No organization or individual can control Bitcoin, and the network remains secure even if not all of its users can be trusted.

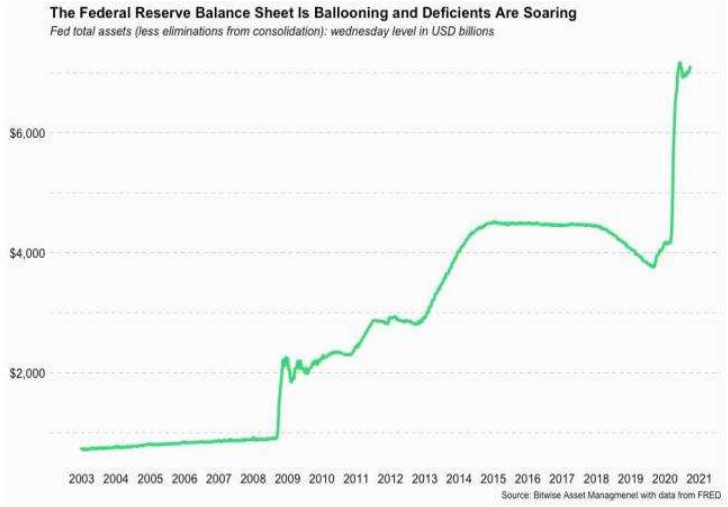
How are bitcoins created?

New bitcoins are generated by a competitive and decentralized process called "mining". This process involves that individuals are rewarded by the network for their services. Bitcoin miners are processing transactions and securing the network using specialized hardware and are collecting new bitcoins in exchange.

The Bitcoin protocol is designed in such a way that new bitcoins are created at a fixed rate. This makes Bitcoin mining a very competitive business. When more miners join the network, it becomes increasingly difficult to make a profit and miners must seek efficiency to cut their operating costs. No central authority or developer has any power to control or manipulate the system to increase their profits. Every Bitcoin node in the world will reject anything that does not comply with the rules it expects the system to follow.

Bitcoins are created at a decreasing and predictable rate. The number of new bitcoins created each year is automatically halved over time until bitcoin issuance halts completely with a total of 21 million bitcoins in existence. At this point, Bitcoin miners will probably be supported exclusively by numerous small transaction fees.

One of the trends in today's Monetary Policy (Mp) is the amount of debt outstanding in the U.S. and the world as a whole. It was not too long ago the investment industry was discussing Modern Monetary Theory (MMT), the direct monetization of massive federal spending, and how we could absorb all of the debt associated with it. Then March 2020 (Covid) came along, and all of a sudden MMT was here with no discussion and no disciplining response of rising market interest rate yields. Asset prices have gone up in value across all asset classes, **without the resulting inflation**. It is this potential rising inflation response that I expect to occur that will really cause problems. The chart below shows how much debt has been created to help combat Covid, borrowed by the Federal Government.



- The Federal Reserve's balance sheet has grown more this year than in any other year.
- The U.S. budget deficit is expected to reach almost $4 trillion in 2020, compared to less than $1 trillion in 2019.
- Investors are worried and searching for an efficient hedge against the risk of unexpected inflation

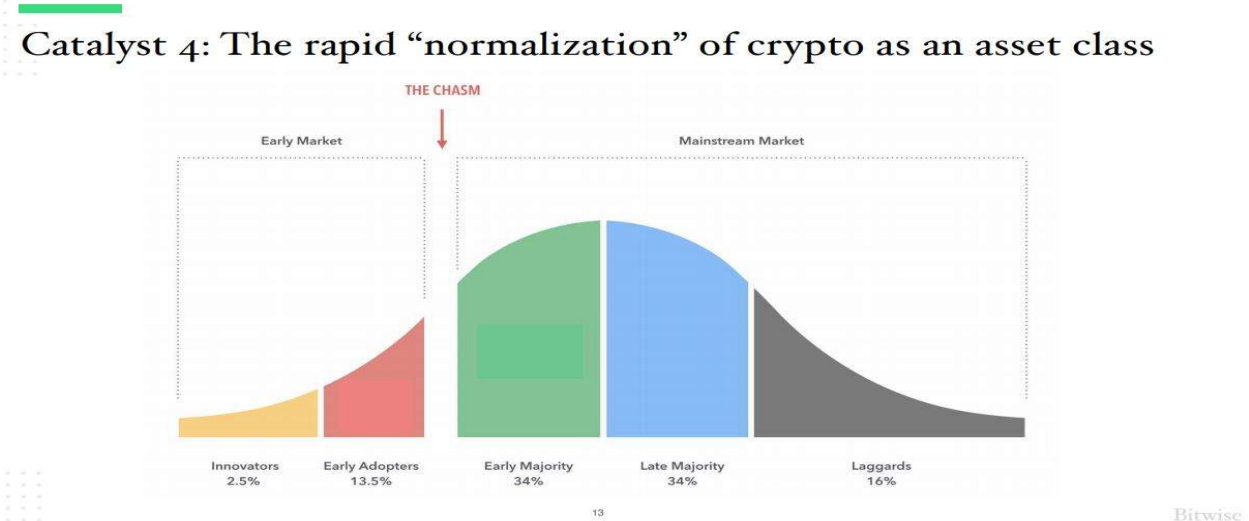
For those of you that have been with me as clients for at least 8 years, you will very clearly remember that each year for the last 8 years I have predicted interest rates to rise, and each year I have been surprised as they moved to the downside. Then there was last year (the year that no one wants to remember or say out loud), **and parts of the yield curve went to Zero!** This year to no one's surprise, my prediction for interest rates is that they will be higher than they were last year, on the way to moving higher over the coming years.

The question that needs to be asked, and what we at RMH are trying to prepare our clients for, **"What is an investor supposed to do?"** Traditional hedges like gold have done well in the past and we expect this to continue. Real assets such as real estate have a place in the portfolio as well. Might we see the rise of cryptocurrencies as playing a role in portfolio management?

From the Stone Ridge 2020 Annual Letter the following is very apt for today:

Risk management = Diversification + Humility

Going forward we need more diversification to protect our portfolio's for the long term.



From the chart above, Bitwise believes we are in the early innings of CC as an asset class. I would agree.

This analysis has been all one sided as to the benefits of CC in portfolios. Time for another thought from a firm whose research we enjoy reading, Research Affiliates. They published an article from one of their employees titled: "**Bitcoin: Magic Internet Money**". The article mentions, "Always know what you are investing in, Bitcoin is not a capital asset or store of value, and the price of BTC is nearly certainly a bubble and likely manipulated.

The full link to the article can be found here:

https://www.researchaffiliates.com/en_us/publications/articles/820-bitcoin-magic-internet-money.html

In conclusion, it is early in the cycle for crypto currency as an asset class for diversifying your portfolio, however, several things to keep in mind:

- Only 22,000,000 BTC will ever be made, scarcity is worth something.
- All governments can print unlimited amounts of money, there have been 56 cases of hyperinflation in the last 100 years, no scarcity.
- In the 1970's the U.S. Government took the US$ off of the gold standard ($35/oz.), **one year later, $125/oz. was the price.**
- While the CC's might be overpriced or in a bubble, will they always? We will know in the next three to five years.
- What is the cost of not having a small amount of portfolio diversification in your portfolio to smooth out the corrections?
- **The markets are clearing mechanisms, and Price is Price.** In the last 12 years we have not allowed the market to function properly, that is a concern as it allows excesses to build.

This was a fun RMH Market Watch to write, a lot to learn and try to disseminate in a readable form to a varied and different group of readers.

We have pdf copies of all of the articles we sourced for this RMH Market Watch, please feel free to ask for them.

What steps are we taking at RMH?

- We are looking at the portfolios and rebalancing where necessary.
- We are taking advantage of tax loss selling to lower future capital gains.
- We are looking at what insiders are doing with some of the stocks we have purchased.
- We are talking with portfolio managers on a one to one basis and participating in conference calls.

If there are ever any topics you wish for us to explore, please let us know. ***We are here to help and guide you through these times.***

We thank you all for taking the time and reading "Market Watch." It is meant as an educational piece on the always evolving markets. It is something we plan on providing every month, and your feedback is very important to us.

Tel: 520-314-2300 Email: richard@rmhinvestment.com 5151 E. Broadway Blvd. Suite 1600
Tucson AZ, 85711

On a personal note, RMH is now in the position to bring on new clients so please be sure to share this informational letter with whomever you wish. RMH's focus is on the customizable investment needs of individuals, families, and foundations. We enjoy working with our clients to better understand their goals, values, and passions for what is important in their lives. In expanding our client base, we look forward to working with people who share these same desires

Richard Mundinger, CFA

Sources:

Bitwise presentation, December 7, 2020 The Bitwise 10 Crypto Index Fund (BITW) to begin trading on OTCQX
https://en.wikipedia.org/wiki/Bitcoin
https://en.wikipedia.org/wiki/Satoshi_Nakamoto
Bitcoin: A Peer-to-Peer Electronic Cash System, Satoshi Nakamoto October 31, 2008
WWW.bitcoin.org
https://www.researchaffiliates.com/en_us/publications/articles/820-bitcoin-magic-internet-money.html
https://www.coindesk.com/cryptocurrency-markets-react-occ-banks-blockchains-price
Stone Ridge 2020 Annual Letter to Shareholders
Market Outlook – Macro Perspective Paul Jones & Lorenzo Giorgianni: March 2020
CFA Inst. Research Foundation: CryptoAssets The Guide to Bitcoin, Blockchain, and Cryptocurrency For Investment Professional.

Tel: 520-314-2300 Email: richard@rmhinvestment.com 5151 E. Broadway Blvd. Suite 1600
Tucson AZ, 85711

Appendix 1

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.

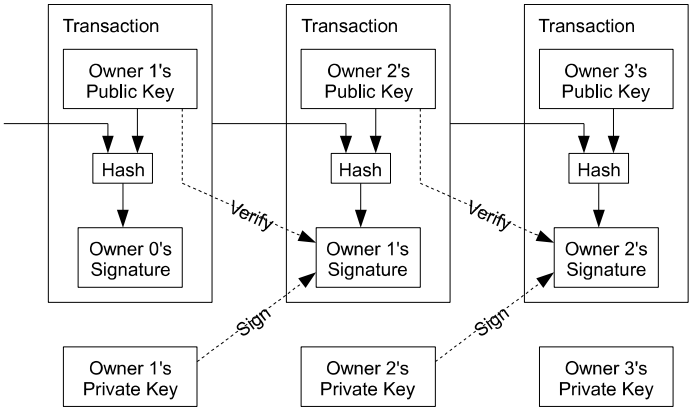
1. Introduction

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 be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

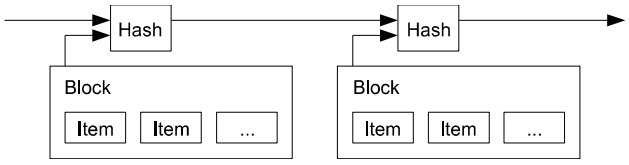


The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new 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

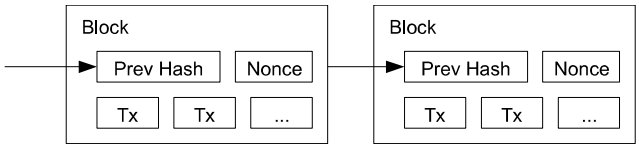
The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to 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.



4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

6. Incentive

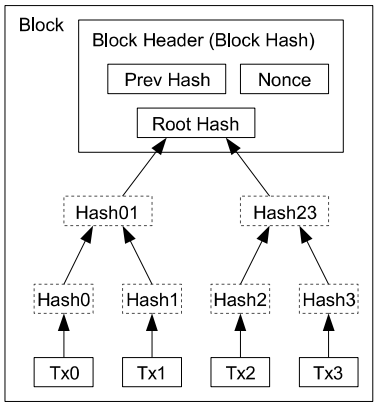
By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

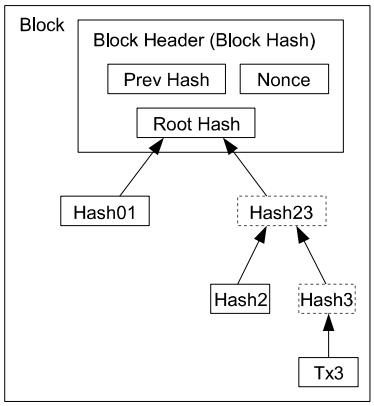
The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

7. Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.



Transactions Hashed in a Merkle Tree

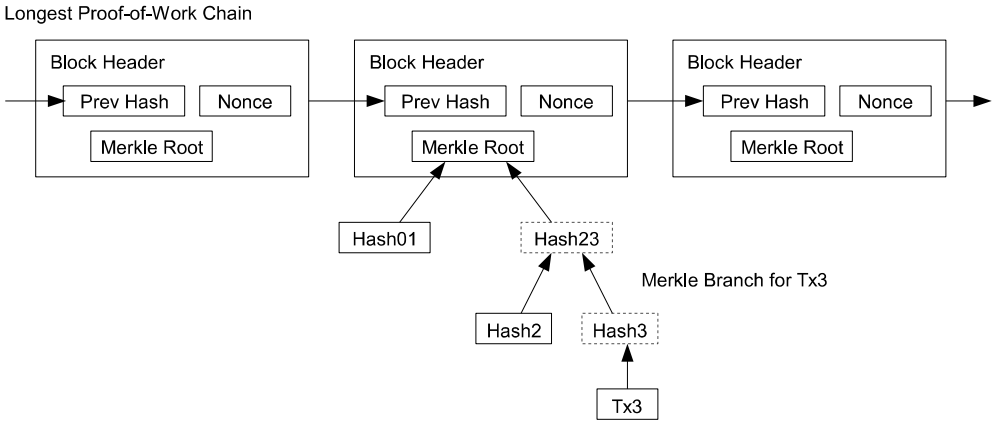


After Pruning Tx0-2 from the Block

A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

8. Simplified Payment Verification

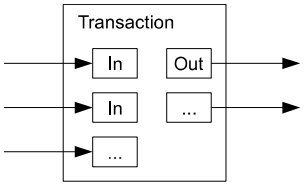
It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. Combining and Splitting Value

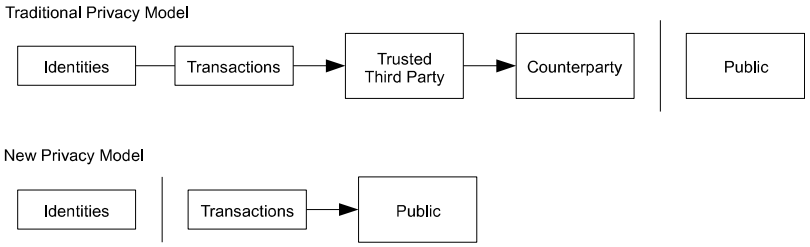
Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

11. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

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

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

Given our assumption that $p > q$, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

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

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

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

Rearranging to avoid summing the infinite tail of the distribution...

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

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Running some results, we can see the probability drop off exponentially with z.

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012

q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

Solving for P less than 0.1%...

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.