Diogo Trentini e Lauro Gripa Neto

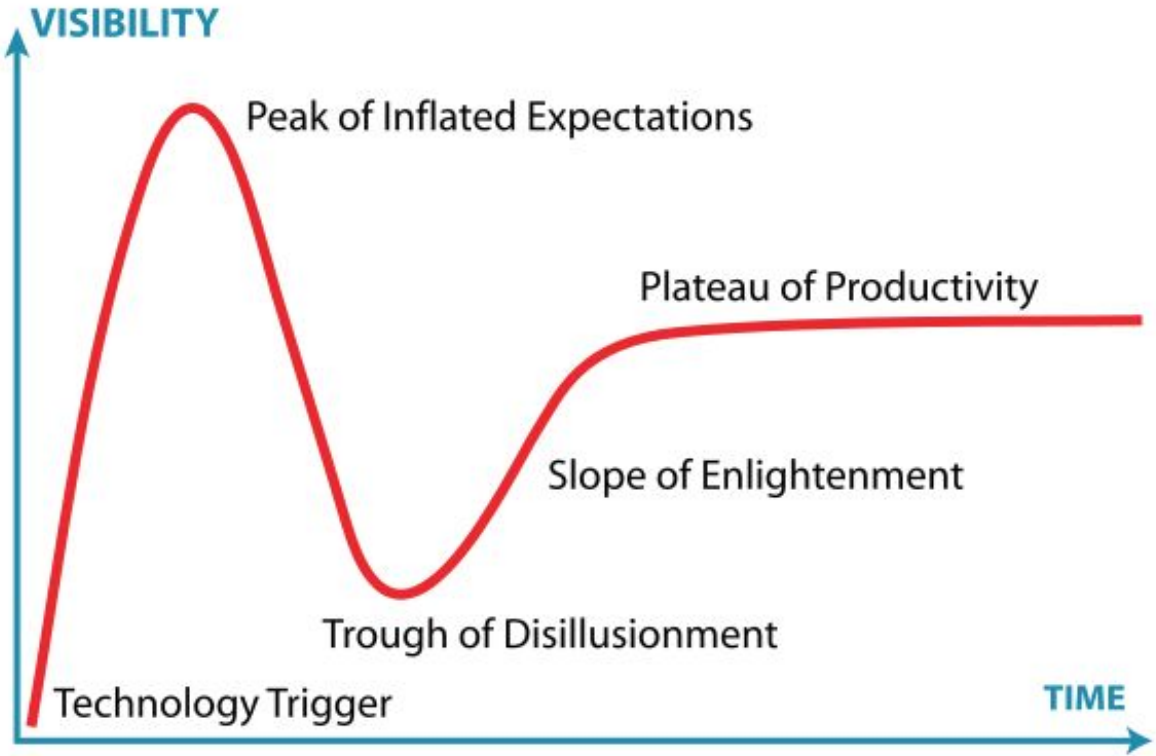
Introduction to **Blockchain**

MAGRATHEA LABS

www.magrathealabs.com

source: Scott Adams' Dilbert

Gartner Hype Cycle



source: Gartner Inc.

SUMMARY

1. Introduction
2. Theoretical concepts
3. Applications
4. Challenges
5. Conclusions


What’s a market?

The "Double-spending" problem

"**Double-spending** is the result of successfully spending some money more than once."

Combating Double-Spending Using Cooperative P2P Systems

4
Author(s)

Ivan Osipkov ; Eugene Y. Vasserman ; Nicholas Hopper ; Yongdae Kim

View All Authors

Abstract:
An electronic cash system allows users to withdraw coins, represented as bit strings, from a bank or broker, and spend those coins anonymously at participating merchants, so that the broker cannot link spent coins to the user who withdraws them. A variety of schemes with various security properties have been proposed for this purpose, but because strings of bits are inherently copyable, they must all deal with the problem of double-spending. In this paper, we present an electronic cash scheme that introduces a new peer-to-peer system architecture to prevent double-spending without requiring an on-line trusted party or tamper-resistant software or hardware. The scheme is easy to implement, computationally efficient, and provably secure. To demonstrate this, we report on a proof-of-concept implementation for Internet vendors along with a detailed complexity analysis and selected security proofs.

Published in: Distributed Computing Systems, 2007. ICDCS '07. 27th International Conference on

Date of Conference: 25-27 June 2007

Date Added to IEEE *Xplore*: 09 July 2007

CD-ROM ISBN: 0-7695-2837-3

Print ISSN: 1063-6927

INSPEC Accession Number: 10290180

DOI: 10.1109/ICDCS.2007.91

Publisher: IEEE

Conference Location: Toronto, ON, Canada

source: http://ieeexplore.ieee.org/document/4268195/

Distributed Double Spending Prevention*

Jaap-Henk Hoepman

TNO Information and Communication Technology
P.O. Box 1416, 9701 BK Groningen, The Netherlands
jaap-henk.hoepman@tno.nl
and
Institute for Computing and Information Sciences
Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, the Netherlands
jhh@cs.ru.nl

Abstract. We study the problem of *preventing* double spending in electronic payment schemes in a *distributed* fashion. This problem occurs, for instance, when the spending of electronic coins needs to be controlled by a large collection of nodes (e.g., in a peer-to-peer (P2P) system) instead of one central bank. Contrary to the commonly held belief that this is fundamentally impossible, we propose several solutions that do achieve a reasonable level of double spending prevention, and analyse their efficiency under varying assumptions.

1 Introduction

Many electronic payment schemes exist. For an overview, we refer to Asokan *et al.* [AJSW97] or O'Mahony *et al.* [OPT97]. Some of those are coin based, where some bitstring locally stored by a user represents a certain fixed value.

Coin based systems run the risk that many copies of the same bitstring are spent at different merchants. Therefore, these systems need to incorporate *double spending* prevention or detection techniques. To *prevent* double spending, a central bank is usually assumed which is involved in each and every transaction. In off-line scenarios (where such a connection to a central bank is not available), double spending *detection* techniques are used that will discover double spending at some later time, and that allow one to find the perpetrator of this illegal activity. A major drawback of double spending detection techniques is the risk that a dishonest user spends a single coin a million times in a short period of time before being detected. This is especially a problem if such a user cannot be punished for such behaviour afterwards, e.g., fined, penalised judicially, or being kicked from the system permanently.

* This research is/was partially supported by the research program Sentinels (www.sentinels.nl), project JASON (NIT.6677). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.
Id: double-spending.tex 18 2008-02-06 14:01:34Z jhh

source: https://bitcoin.org/bitcoin.pdf

Bitcoin: A Peer-to-Peer Electronic Cash System

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

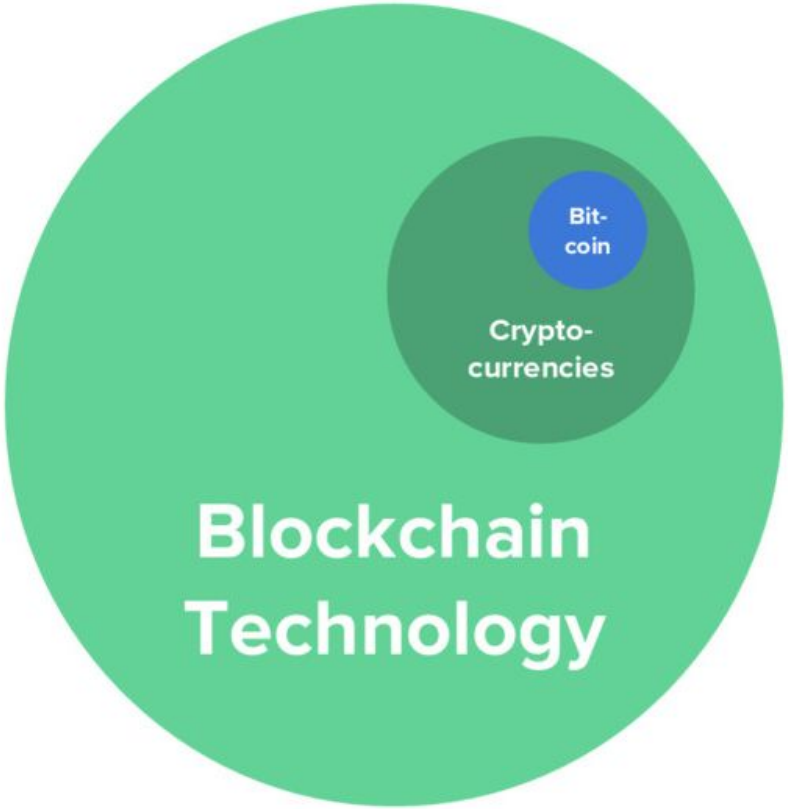
Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required 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.

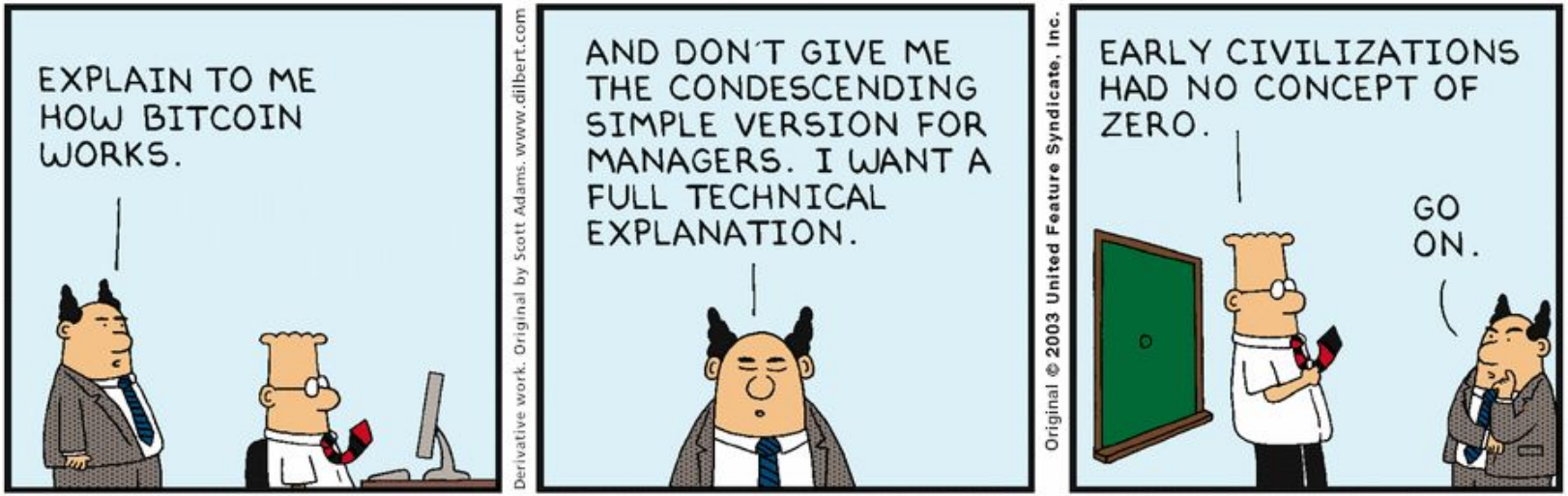
1. Introduction

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.

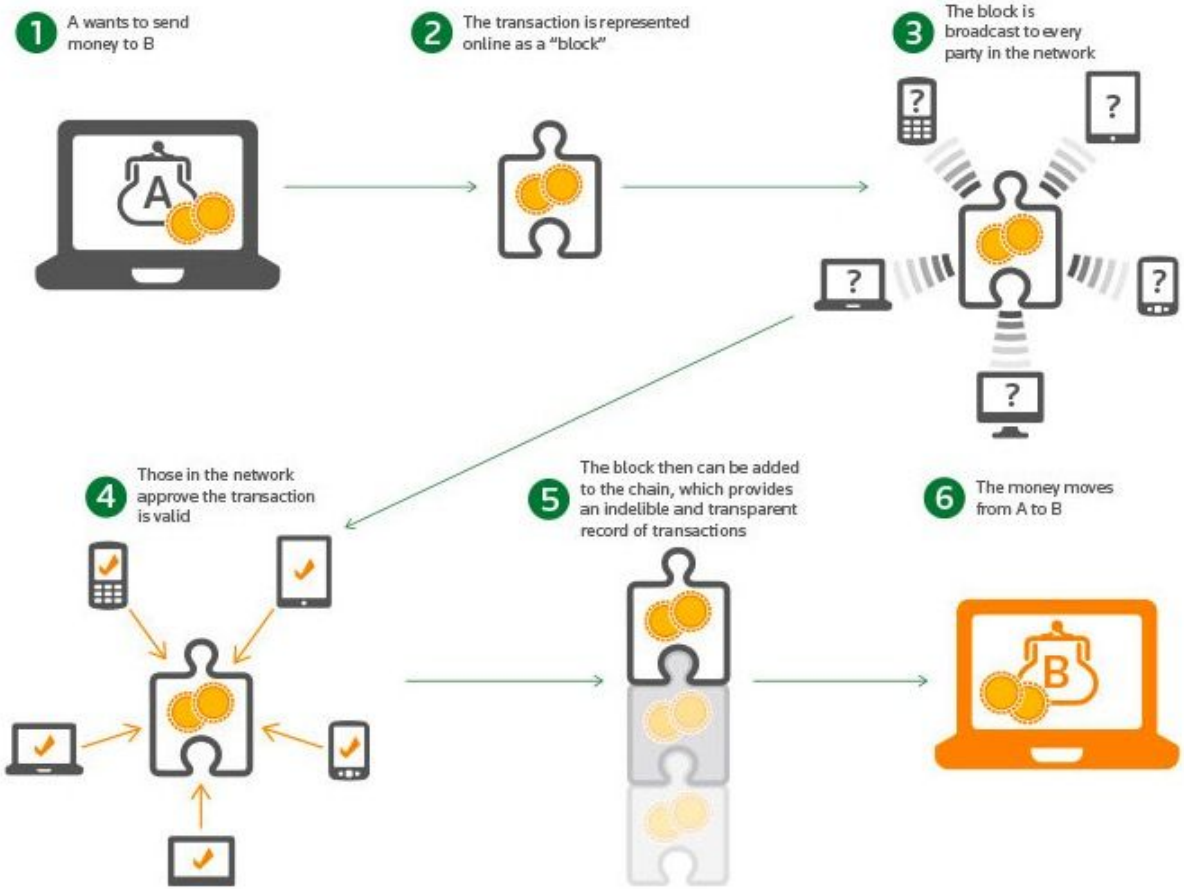
What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, 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.

BITCOIN ≠ BLOCKCHAIN


Bit-
coin
Crypto-
currencies
Blockchain
Technology



source: Scott Adams' Dilbert



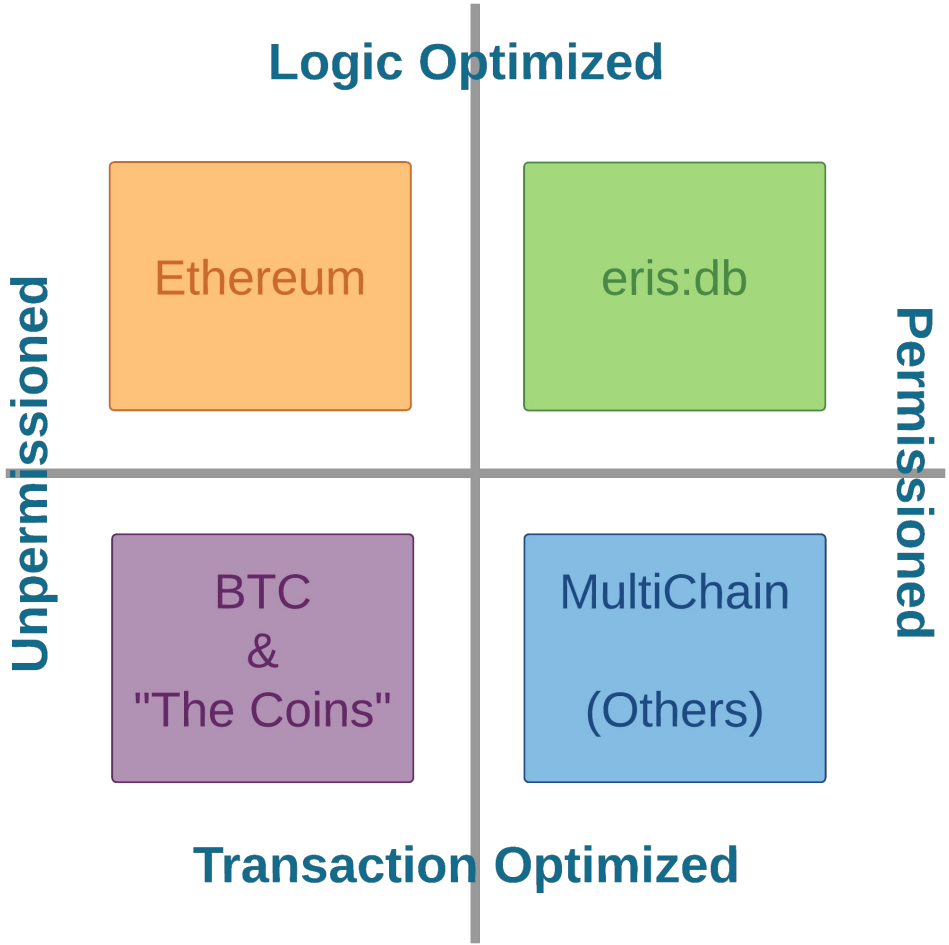
source: https://www.weforum.org/agenda/2015/11/how-will-blockchain-technology-transform-financial-services/

[DEMO]

https://anders.com/blockchain/

BLOCKCHAIN / BITCOIN COMPONENTS

1. Address (public keys) + Wallet (private keys)
2. Transactions / Contracts
3. Blocks
4. Ledger
5. Consensus Network + Consensus Algorithm



source: https://monax.io/explainers/blockchains/

source: envisioning.io/money

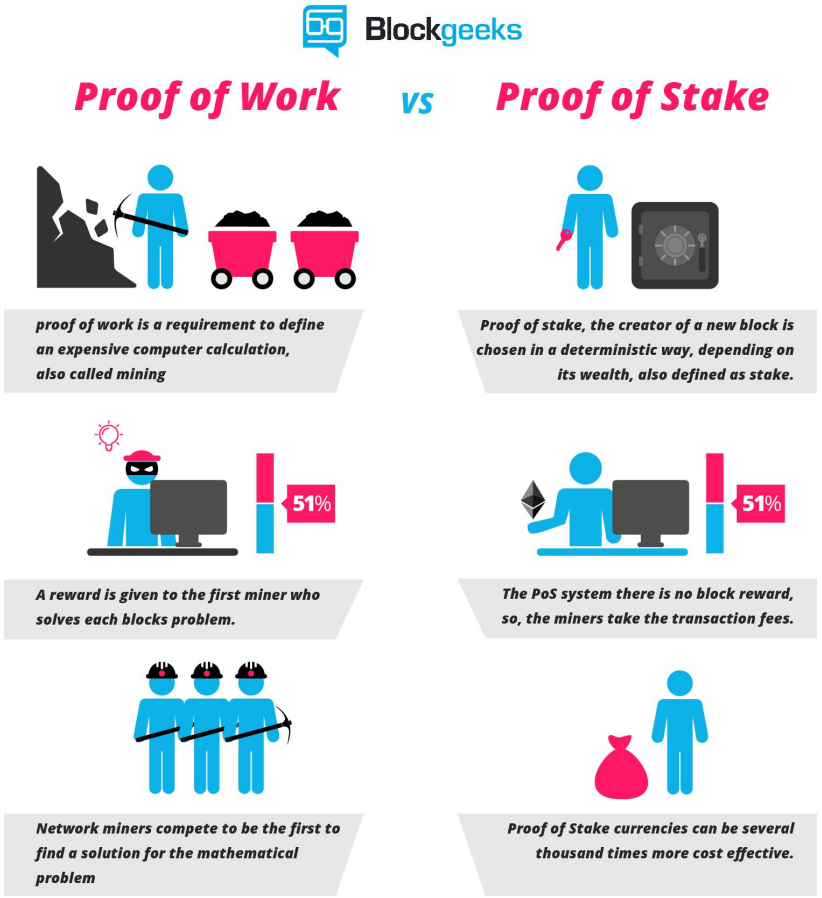
BUT... IT'S "JUST" A DATABASE

1. Fancy name: "Distributed Ledger"
2. CAP Theorem:
    a. "It's impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees: consistency, availability, and partition tolerance."

Game Theory

"The primary role of blockchains are to solve **coordination problems** among **multilateral agreements** between a **network of participants**."

source: https://bravenewcoin.com/assets/Whitepapers/OmiseGO-Whitepaper.pdf



source: https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/

APPLICATIONS

ETHEREUM

“Ethereum is a decentralized platform that runs smart contracts: **applications** that run exactly as programmed **without any possibility of downtime, censorship, fraud or third party interference**.”

source: https://ethereum.org/

GOLEM

“Golem is a global, open sourced, **decentralized supercomputer that anyone can access**. It's made up of the combined power of user's machines, from personal laptops to entire datacenters.”

source: https://golem.network/

NUMERAI

“Machine learning competitions are susceptible to intentional overfitting. Numerai proposes Numeraire, a new cryptographic token that can be used in a novel auction mechanism to **make overfitting economically irrational**.”

source: https://numer.ai/

OMISEGO

“OmiseGO is a public Ethereum-based financial technology for use in mainstream digital wallets, that enables **real-time, peer-to-peer value exchange** and payment services agnostically across jurisdictions and organizational silos, and across both fiat money and decentralized currencies.”

source: https://omg.omise.co/

CHALLENGES

1. Adoption / Regulation
2. Scalability
3. Energy consumption (when PoW)
4. Vulnerabilities:
    a. Social engineering
    b. 51% attack, sybil attack and such
    c. Bugs

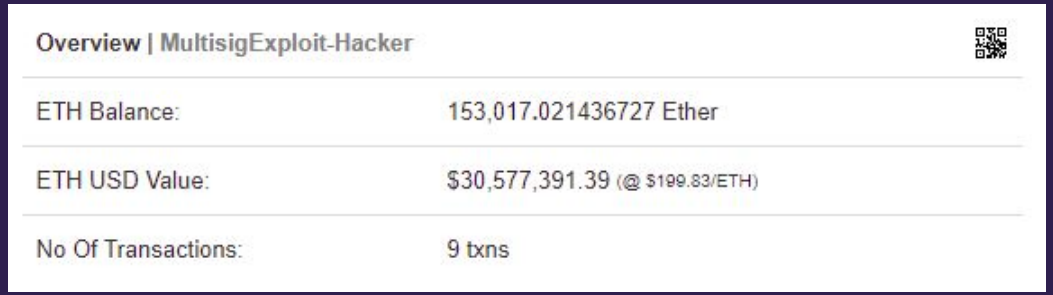
VULNERABILITIES

```
// constructor - just pass on the owner array to the
// multiowned and the limit to daylimit
function initWallet(address[] _owners, uint _required,
uint _daylimit) {
  initDaylimit(_daylimit);
  initMultiowned(_owners, _required);
}
```

VULNERABILITIES

```
function() payable {
  // just being sent some cash?
  if (msg.value > 0)
    Deposit(msg.sender, msg.value);
  else if (msg.data.length > 0)
    _walletLibrary.delegatecall(msg.data);
}
```

EXPLOIT ADDRESS



Overview | MultisigExploit-Hacker	
ETH Balance:	153,017.021436727 Ether
ETH USD Value:	$30,577,391.39 (@ $199.83/ETH)
No Of Transactions:	9 txns

WHITE HAT GROUP

WHITE HAT GROUP ADDRESS

Overview | MultisigExploit-WhiteHat

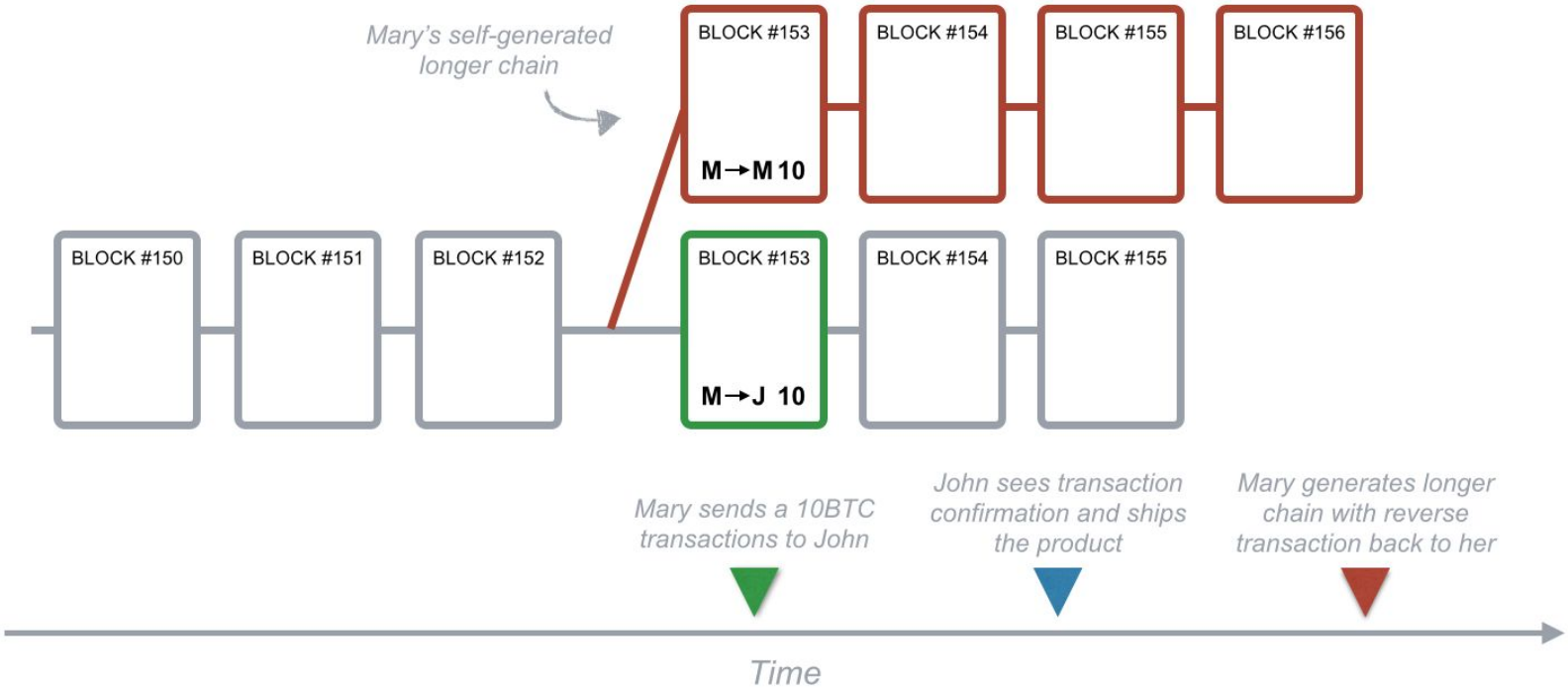
ETH Balance:	377,113.498729249311671493 Ether
ETH USD Value:	$75,528,291.53 (@ $200.28/ETH)
No Of Transactions:	2558 txns

FIXED COMMIT

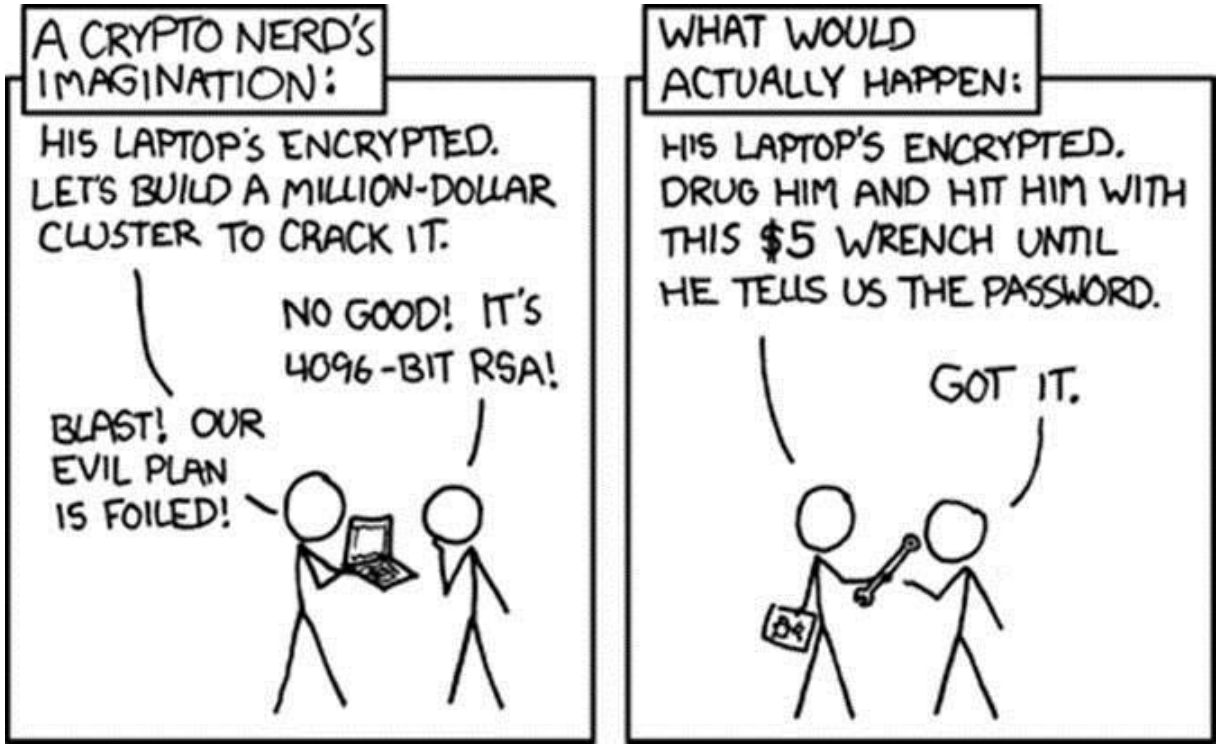
```
104
105     // constructor is given number of sigs required to do protected "onlymanyowners" transactions
106     // as well as the selection of addresses capable of confirming them.
107 +   function initMultiowned(address[] _owners, uint _required) internal {
108       m_numOwners = _owners.length + 1;
109       m_owners[1] = uint(msg.sender);
110       m_ownerIndex[uint(msg.sender)] = 1;

198     }
199
200     // constructor - stores initial daily limit and records the present day's index.
201 +   function initDaylimit(uint _limit) internal {
202       m_dailyLimit = _limit;
203       m_lastDay = today();
204     }

211       m_spentToday = 0;
212     }
213
214 +   // throw unless the contract is not yet initialized.
215 +   modifier only_uninitialized { if (m_numOwners > 0) throw; _; }
216 +
217     // constructor - just pass on the owner array to the multiowned and
218     // the limit to daylimit
219 +   function initWallet(address[] _owners, uint _required, uint _daylimit) only_uninitialized {
220       initDaylimit(_daylimit);
221       initMultiowned(_owners, _required);
222     }
```



source: https://medium.com/@micheledaliessi/how-does-the-blockchain-work-98c8cd01d2ae



source: xkcd.com/538

Old concepts, new tool

"The two differentiators of DLT in my opinion: (a) the control of the read/write access is **truly decentralized** and not logically centralized as for other distributed databases, and corollary (b) the ability to secure transactions in competing environments, **without trusted third parties**."

source: https://medium.com/@sbmeunier/e63d00781118

Decentralized, but trust is still needed

"While one of the most important features of blockchains is removing the need for trusted third parties, **most people don't have the time or background to thoroughly evaluate the software**, which means that trust is still needed: trust in the developers of the project, or someone else capable of evaluating the software."

source: https://medium.com/@neha/9a6a9ddc4367

Not a jack-of-all-trades

"Blockchain is **not a general-purpose solution for everything**. It should be considered as an **enabler to creating new decentralized services** and **solving specific business problems** (such as the double-spend problem in trustless P2P environments for the brilliant Bitcoin)."

source: https://medium.com/@sbmeunier/e63d00781118


Questions?
MAGRATHEA LABS
www.magrathealabs.com

https://www.meetup.com/
Joinville-Blockchain-Meetup

MAGRATHEA LABS

www.magrathealabs.com
contact@magrathealabs.com

Thank you!

MAGRATHEA LABS

www.magrathealabs.com