THE RISK AND TEST IN BITCOIN

As a thought experiment, imagine there was a base metal as scarce as gold but with the following properties:
- boring grey in colour
- not a good conductor of electricity
- not particularly strong, but not ductile or easily malleable either
- not useful for any practical or ornamental purpose

and one special, magical property:
- can be transported over a communications channel.

Satoshi Nakamoto

Abstract

If there were no risk of anything ever going wrong, there would be no case for Testing or Assurance. Unfortunately, superficial risk assessments can misdirect all the efforts that follow in their wake, while failing to control real risks.

Although Testing and Assurance activities are part of Risk Management, risk is still poorly understood and many mistakes are made by taking amateur approaches to risk assessments. Not speaking about risks that are difficult to understand does not make them go away. This paper describes common mistakes to avoid, and applies the theory of risk assessment to an important real-life example.

Bitcoin is used as an example because it is revolutionary, both in a technical and also political sense. Almost everyone has heard of Bitcoin and Blockchain, but not many understand how they work, or why they were created.

One of the key components to the success of Bitcoin has been the incorporation of a new type of machine-to-machine testing called 'Proof of Work' to validate transactions and establish consensus where there are no trusted third parties. Proof of Work and the variations such as proof of stake all depend upon an unproven assumption: that solving a hard problem takes longer than checking the result of the solution. The Clay Mathematics Institute will pay you $1 million if you can prove that it does, or that it doesn't.

How safe is that assumption?

Declan O'Riordan
Declan@TestingIT.co.uk

PART ONE – BITCOIN: THE ORIGINAL BLOCKCHAIN

However distasteful such an admission may be, we must recognise that we had before this war once again reached a stage where it is more important to clear away the obstacles with which human folly has encumbered our path and to release the creative energy of individuals than to devise further machinery for "guiding" and "directing" them – ***to create conditions favourable to progress rather than to "plan progress".***

Friedrich August von Hayek, The Road to Serfdom, 1943

A brief history of money:

Twenty years ago in the Rift Valley of Kenya, archaeologist Stanley Ambrose from the University of Illinois discovered a cache of beads made from ostrich eggshell fragments. They were argon dated to be at least 40,000 years old. Pierced animal teeth found in Spain have been dated to the same era, and perforated shells have also been recovered from early Palaeolithic sites in Lebanon. Regular shells, prepared as strung beads and dating to 75,000 BP (Before Present), were found in the Blombos cave in South Africa. Why did humans, often living on the brink of starvation, spend so much time making and enjoying necklaces when they could have been doing more hunting and gathering?

These collectibles had very specific attributes. They were not merely symbolic. While the concrete objects and attributes valued as collectible could vary between cultures, they were far from arbitrary. The primary and ultimate evolutionary function of collectibles was as a medium for storing and transferring wealth. Such proto-money overcame the inherent limitations of barter trade, provided the trading networks believed the collectables were valuable. The legacy remains visible into the present day when people wear necklaces and bracelets to display their wealth, as our ancestors did in ancient times.

5 Million Mark coin would have been worth $714.29 in Jan 1923, about 1 thousandth of one cent by Oct 1923.

Today Governments control the supply of money through fiat currency (from the Latin fiat "*let it be done*"). Fiat money is currency that a government has declared to be legal tender, but it is not backed by a physical commodity. The value of fiat money is derived from the relationship between supply and demand rather than the value of the material from which the money is made. Historically, attempts to produce alternative currencies have usually been punished harshly, sometimes by imposition of the death penalty. Fiat currencies are defended vigorously to maintain Government control of

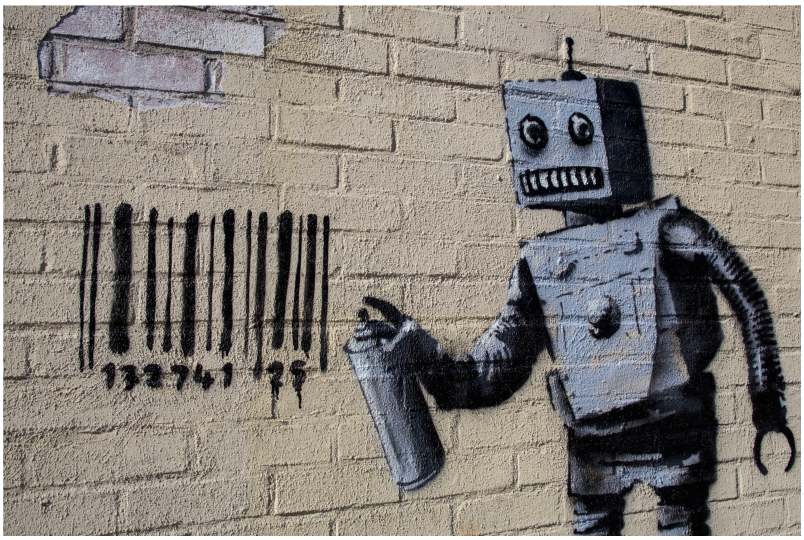
the tax base, yet they are prone to debasement and inflation through mismanagement of the economy. Some notable examples of hyperinflation are: Germany (*Nov. 1923: 29,525%),* Zimbabwe (*Nov. 2008: 79.6 billion%*), Venezuela (*expected to reach one million percent by December 2018*).

The Cypherpunks

In the 1970's and 80's a counter-culture movement began to emerge with strong interests in technology and promoting individual privacy. Topics discussed ranged across mathematics, cryptography, computer science, politics and philosophical discussion. One member of the group, Judith (Jude) Milhon combined the popular term 'cyberpunk' with cypher (an algorithm for performing encryption or decryption) and created a new word 'cypherpunk'' to describe her best friends. It is worth noting their approach to work radically differs from the established norms.

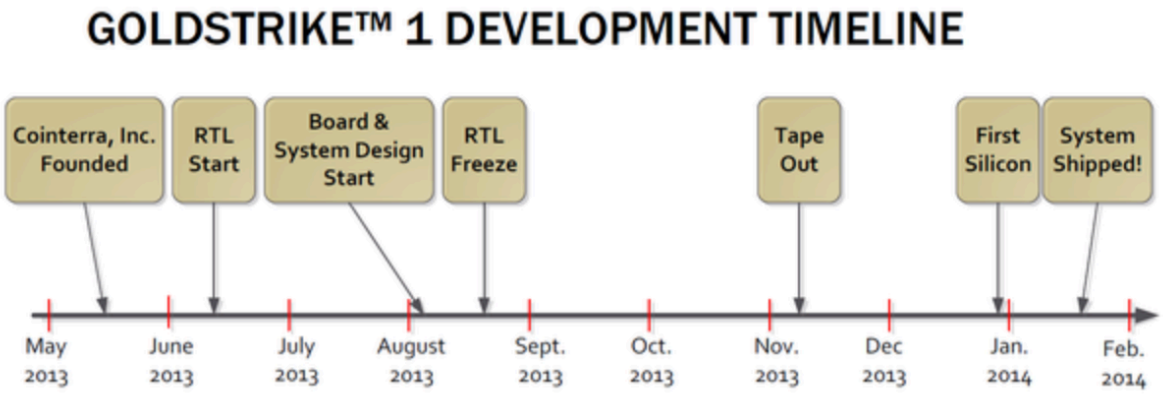
Methodologies such as the Rational Unified Process tell people **what** to do, **when** to do it, and **how**. **Frameworks** such as Agile tell people **what** to do and **when** but not how, in order to allow flexibility (for some people ambiguity). Cypherpunk developments are driven by **Philosophy.** The philosophy provides a body of knowledge and reasons **why**, for example why they should provide society with privacy-enhancing technologies. Originally, Friedrich August von Hayek and Ayn Rand influenced cypherpunks. More recently, Nick Land and Curtis Yarvin (A.K.A. Mencius Moldbug) are influencing their successors.

The cypherpunks rapidly produce revolutionary code and highly advanced hardware, often without conventionally recognisable management or planning activities such as assigning roles and responsibilities. We can see it works in practice, but we don't know if it works in theory because there isn't a theory. No label has yet been found to describe the distilled elements of Cypherpunk development but several Universities in the USA are currently researching the mining chip design, delivery and processing speed phenomenon.



"We examined the Bitcoin hardware movement, which led to the development of customized silicon ASICs without the support of any major company. The users self-organized and self-financed the hardware and software development, bore the risks and fiduciary issues, evaluated business plans, and braved the task of developing expensive chips on extremely low budgets. This is unheard of in modern times, where last-generation chip efforts are said to cost $100 million or more."

—Michael Bedford Taylor, University of California



The timeline above shows an example of Bitcoin mining hardware advancing from register-transfer level (RTL) design to 'tapeout', the point at which the graphic for the photomask of the circuit is sent to the fabrication facility in just four months. Normally this takes years.

"The amazing thing about Bitcoin ASICs is that, as hard as they were to design, analysts who have looked at this have said this may be the fastest turnaround time - essentially in the history of integrated circuits - for specifying a problem, which is mining Bitcoins, and turning it around to have a working chip in people's hands."
—Joseph Bonneau, Postdoctoral research associate, Princeton University

The Cypherpunk journey to Bitcoin and blockchain

In the 1970's a series of breakthroughs in cryptography occurred. Public-key cryptography provided unbreakable secret communications for the first time in history, but not for ordinary people. The relationship between citizens and State has always been highly unequal. Cypherpunks objected to Governments and large organisations treating individuals' data as their own property to collect, analyse and use however they liked. In 1991 cypherpunk Phil Zimmerman released PGP (Pretty Good Privacy) to enable the general public to make and receive private email communications. In 1993 Zimmerman became the formal target of a criminal investigation by the US Government for "munitions export without a license" because encryption was classified as a weapon.

On 9th March 1993 Eric Hughes published The Cypherpunk Manifesto containing the opening line: "*Privacy is necessary for an open society in the electronic age. Privacy is not secrecy. A private matter is something one doesn't want the whole world to know, but a secret matter is something one doesn't want anybody to know. Privacy is the power to selectively reveal oneself to the world.*" The closing statement provided a reason why cryptocurrency was needed: "*The act of encryption, in fact, removes information from the public realm. Even laws against cryptography reach only so far as a nation's border and the arm of its violence. Cryptography will ineluctably spread over the whole globe, and with it the* ***anonymous transactions systems*** *that it makes possible.*" Comparing the Manifesto with introduction of the EU General Data Protection Regulation 25 years later it is hard not to think the Cypherpunks created the perfect insurgency.

On 10th September 1994 Tim May published The Cyphernomicon, providing extensive details on a range of cypherpunk projects including a digital cash economy to ensure privacy through anonymous transactions systems. Some extracted headlines are shown below:

2.12. Digital Cash
2.12.1. "What is digital money?"
2.12.2. "What are the main uses of strong crypto for business and economic transactions?"
- Secure communications. Ensuring privacy of transaction records (avoiding eavesdroppes, competitors)
- Digital signatures on contracts (will someday be standard)
- Digital cash.
- Reputations.
- Data Havens.

10.8.8. "Will banking regulators allow digital cash?"
- Not easily, that's for sure. The maze of regulations, restrictions, tax laws, and legal rulings is daunting.
10.8.9. Legal obstacles to digital money. If governments don't want anonymous cash, they can make things tough.

> *+ As both Perry Metzger and Eric Hughes have said many times, regulations can make life very difficult. Compliance with laws is a major cost of doing business.*

10.9. Legality of Digital Banks and Digital Cash?
10.9.1. In terms of banking laws, cash reporting regulations, money laundering statutes, and the welter of laws connected with financial transactions of all sorts, the Cypherpunks themes and ideas are basically illegal. Illegal in the sense that anyone trying to set up his own bank, or alternative currency system, or the like would be shut down quickly.

The Cyphernomicon predictions were accurate. Before Bitcoin, 98 digital currencies were created and destroyed by attacks upon the central trust authority (e.g. imprisoning the owners and/or regulating their businesses out of existence), or by hackers 'double spending' the currency (copying the digital money file and re-spending it while corrupting the central authority). There was clearly a need for a better digital currency that prevented double-spending and removed the attack target presented by central control. These problems were the motivations for cypherpunks to work unpaid on delivering a new solution that resisted attack by a design unlike anything constructed before. Bitcoin does not comply with any banking standards or financial regulations, yet it was built, implemented, and rapidly grows.

The Genesis Block

The root of trust for the bitcoin blockchain is the genesis block (the first bitcoin block in the first blockchain). All subsequent blocks are linked back to the Genesis block in a chain designed to be unbreakable and immutable. It contains a hexadecimal encoded message within the 'coinbase' (the first transaction in every bitcoin block placed by the 'winning' miner to create new bitcoins as a reward for their successful mining work): `coinbase 04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e636 56c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f72 2062616e6b73)`
The decoded message reads:
"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks". This message serves two purposes: to demonstrate the genesis block could not have been created before 3rd January 2009; and to protest at Government mismanagement of the economy it was

supposed to control. It also provides a clue that the creator was an anglophile and read the Times newspaper produced in Britain.

A Cypherpunk using the pseudonym 'Satoshi Nakamoto' created the Genesis block. Satoshi Nakamoto has never been identified, except possibly by the United States National Security Agency (NSA) confidential use of stylometry to compare known Satoshi writings with the NSA database containing billions of document samples with identified authors. Edward Snowden, a defector from the NSA, revealed the method used to gather some of these documents is mass population surveillance. Most people have a writing style akin to digital fingerprints and Barack Obama wanted Satoshi identified during his presidency. Satoshi Nakamoto (who may be more than one person) used his pseudonym between August 2008 and December 2010 when communicating with fellow cryptocurrency cypherpunks Nick Szabo, Wei Dai, & Hal Finney (who unfortunately was diagnosed with motor neurone disease in 2009 and died in 2014). Apart from a brief final message in 2013, Satoshi disappeared, but not before publishing the bitcoin white paper in 2008 and developing the code to make bitcoin a reality on 3rd January 2009.


The National Security Agency. Source: Reuters

Bitcoin messages in transit and the public bitcoin transaction log are not encrypted because the security model is reversed from the traditional central control of trust, such as the 'layers of an onion' concentric model. All bitcoin nodes are responsible for establishing trust linked back to the Genesis block using a distributed peer-to-peer consensus network. Data is visible 'in the clear' to enable validation by all nodes in the network. The log shows Satoshi Nakamoto mined roughly one million bitcoins during January 2009. Except for test transactions these remain unspent. At bitcoin's peak traded value in December 2017 this hoard was worth over US $19 billion, making Nakamoto possibly the 44th richest person in the world at the time, if he is still alive.

To ask why Satoshi Nakamoto didn't convert the bitcoins into fiat currency is to fail to understand the basic motivations for creating bitcoin. Cyperpunks began, and are still implementing, an organised exit from economies controlled by intrusive Governments use of force. To confuse cypherpunks with cryptocurrency speculators is akin to seeing no difference between the Sex Pistols and punk-style poseurs who held no punk beliefs.

How Bitcoin Works **THIS IS AN IMPORTANT PART!!!**

Satoshi Nakamoto used cryptography to circumvent the mistakes that had previously led to digital currency failures. He avoided the legislative and security vulnerability of having centralised control by introducing a decentralised peer-to-peer network to verify transactions were valid and not 'double spends'.

The consensus mechanism includes an ingenious adoption of a cryptographic hash process called Hashcash. Using Hashcash-type algorithms is known generically as **Proof of Work**. All variations on Proof of Work contain a decision point: Is the hash solution valid or invalid? The decision point is a test. As we shall see later, this test is accurately executed at far higher speeds than any other computation in history (currently around 200 billion tests in the time a photon of light travels one metre, with exponentially increasing speeds every year).

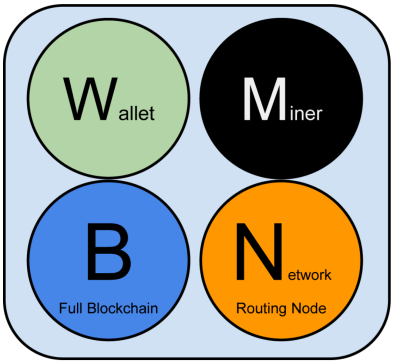
The crucial characteristic of Proof of Work is **computational asymmetry**. Solving a Proof of Work problem must be hard and time consuming, but checking the solution is always quick, like a game of Sudoku or completing a Jigsaw. When advances in computer processing power reduce the time taken to provide a solution, the difficulty (i.e. number of operations) to calculate a solution is increased. The validity of the harder solution can still be quickly checked, usually in one operation, even if billions of extra operations are added to finding the solution. Imagine playing Sudoku when the number of rows and columns are increased every time you learn how to solve the problems faster. Satoshi Nakamoto and the small group of Cypherpunks interested in cryptocurrency saw the potential to use Proof of Work in **Machine-to-Machine (M-2-M) Testing** and prevent invalid financial transactions being recorded in a ledger without the oversight of a trusted third party.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

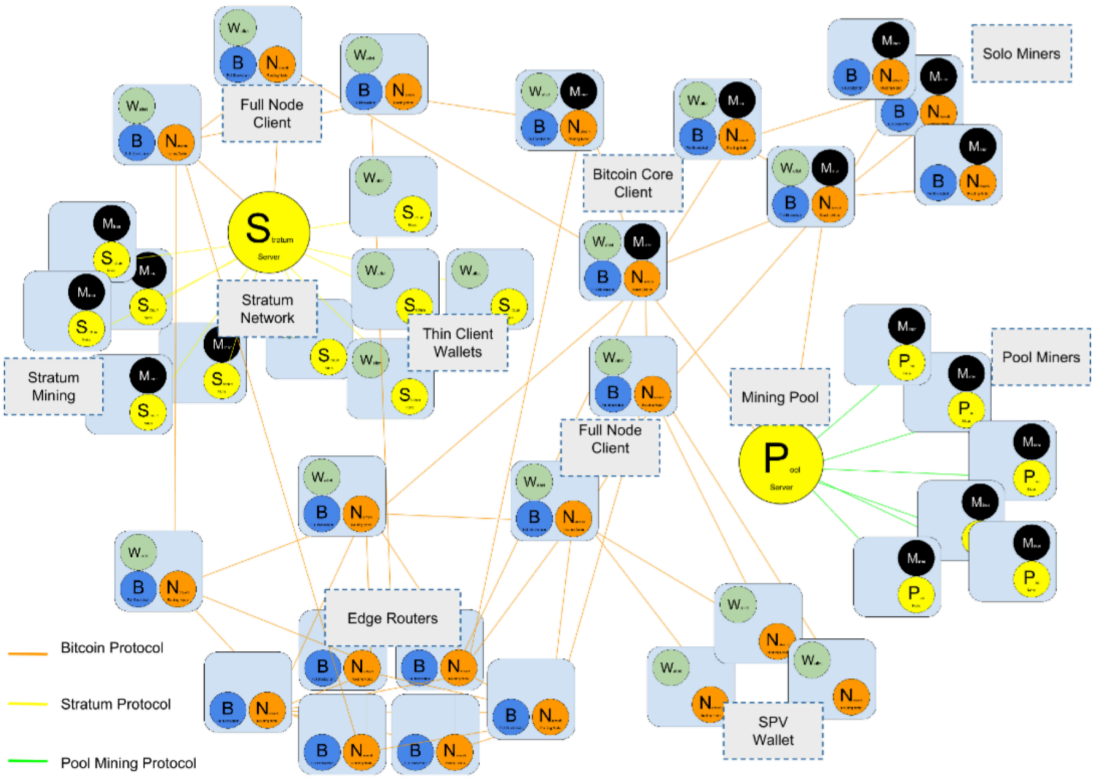
In the case of bitcoin, there is no central ledger. The bitcoin ledger is distributed as a copy to every full node in the peer-to-peer network and each miner races to complete a Proof of Work solution based upon the Hashcash algorithm. Satoshi Nakamoto's most important invention is the decentralised mechanism for **emergent consensus**. Thousands of independent nodes follow a common set of rules to reach majority agreement built upon four processes, continuously executed by Machine-to-Machine Testing:

1. Independent verification of each transaction through a comprehensive set of criteria such as syntax, data structure, size, value, unlocking cryptographic scripts to prove ownership, matching unique outputs and inputs etc. By verifying each transaction as it is received and before propagating it across the peer-to-peer network, every node



builds a pool of valid but unconfirmed transactions known as the transaction pool.

2. Mining nodes perform independent aggregation of transactions in the transaction pool and place valid transactions into new 'candidate blocks', coupled with demonstrated computation through the Proof-of-Work algorithm. The average block contains over 1900 transactions, plus a block header consisting of metadata as described in the 'Bitcoin Mining using SHA-256' section further below. Not all nodes with a full copy of the ledger are miners. When a miner constructs a candidate block they add a coinbase transaction detailing an output payment of new bitcoins (currently 12.5 bitcoin per block as an incentive to miners) plus transaction fees for validated transactions, all payable to a bitcoin address owned by the miner. The bitcoin issuance rate halves every 210,000 blocks (every four years) and will drop to 6.25 bitcoins at block number (block height) 630,000 in 2020. In 2137, after 32 'halvings' block 6,720,000 will issue a mining reward of the smallest currency unit, just one satoshi (1/100,000,000th of a bitcoin). In 2140, after 6.93 million blocks and issuance of 21 million bitcoins, no more bitcoins will ever be issued and miners will only be rewarded by transaction fees: sum{inputs} minus sum{outputs}.



3. New blocks are independently verified by every node and assembled into the chain of blocks. The verification criteria include: syntactically valid data structure; a proof-of-work test on the header hash; a timestamp less than two hours in the future; acceptable block size; only the first transaction is a coinbase transaction; and that all transactions in the block were validated using the transaction verification criteria. Every node validates blocks according to the same rules and any miner attempting to cheat has their block rejected, wasting their computational energy. Invalid blocks are rejected as soon as any one of the validation criteria fails and are never included in the blockchain.

4. The chain with the most cumulative computation demonstrated through Proof-of-Work is independently selected by every node in the peer-to-peer network. Once a node has validated a new block it attempts to assemble a chain by connecting the block to the existing blockchain. The 'main chain' is whichever valid chain of blocks has the most cumulative proof of work associated with it. Sometimes forks appear in the main chain because blocks arrive at different nodes at different times and copies of the ledger are temporarily inconsistent. Forks are resolved by each node selecting the chain of blocks with the most proof of work and then all nodes converge on a consistent ledger state. When mining nodes choose which chain to extend, their block represents a vote in favour of the chain they selected.

Secure Hashing Algorithms

The original 'Hashcash' proposal used Secure Hash Algorithm 1 (SHA-1), which was deprecated in 2010. Bitcoin uses Hashcash to execute the more secure SHA-256 algorithm.

In simplified terms, hashing is a bit like making mincemeat:

- It is a one-way function that cannot be reversed.
- You cannot tell what the original input looked like.
- No two outputs are identical (at a detailed level).

At a more detailed level, cryptographic hashing:

- Is deterministic: the same message always results in the same hash.
- Is quick to compute the hash value for any given message.
- It is infeasible to generate a message from its hash value except by trying all possible messages.
- A small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value.
- It is infeasible to find two different messages with the same hash value (strong collision resistance).
- Secure hash algorithms are iterative, one-way functions that can process a message to produce a condensed representation called a *message digest*.
- The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits long: SHA-224, **SHA-256**, SHA-384, SHA-512, SHA-512/224, SHA-512/256.
- Each algorithm can be described in two stages: pre-processing and hash computation.
    - Pre-processing involves padding a message, parsing the padded message into *m*-bit blocks, and setting initialization values to be used in the hash computation.

- The hash computation generates a *message schedule* from the padded message and uses that schedule, along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash value generated by the hash computation is used to determine the message digest.

Try Hashing

Using a website that provides free hashing such as http://www.hashemall.com let's explore how hashing works by selecting the SHA 256bit algorithm, then slightly varying inputs and examining the outputs. SHA-256 takes variable length inputs and produces binary output that is always 256 bits long. The outputs are then converted to hexadecimal (Base16) for display in a more compact format. If we used the same input every time we would see the same output every time. If you use the same inputs as me you will see the same results. Try it with your own name or another input of your choosing if you like.

For the purposes of demonstration, I'm incrementing a three-digit number as a suffix to my name. We call this number a nonce, short for 'number used once'. The nonce makes the output unique every time, even though most of the input is unchanging. Bitcoin miners have to find a SHA-256 hash of the block header in their proposed new block, but a difficulty target is set to produce a hash with a certain number of leading zeros. To vary input miners combine nonces with the header data. More on that later but for now let's try to find a hash of my input that has a single leading zero in the output.

Input 1 -> Declan O'Riordan 001

Output 1 - **Hash (sha256) of selected text** (0.005 seconds):

`64404BC1B3F8B789B47AF20171761A469996DE0E9C95EA6BC0E7BEFBEAD2CBBB`

Input 2 -> Declan O'Riordan 002

Output 2 - **Hash (sha256) of selected text** (0.005 seconds):

`244FC602AC259841857D95D1D8E48EAB0112F5ABC97C35124E62B7FF9ECA4160`

We changed the input nonce and got a completely different hash, but not with a leading zero.

Input 3 -> Declan O'Riordan 003

Output 3 - **Hash (sha256) of selected text** (0.004 seconds):

`9AFFD1B0D97A4E12843C46F1047E6EE59066DF2518A37A733B842D873617C04E`

In terms of probabilities, if the output of the hash function is evenly distributed we would expect to find a result with a single 0 as the hexadecimal prefix once every 16 hashes (one out of 16 hexadecimal digits 0 through F). But randomness is not evenly distributed!

Input 4 -> Declan O'Riordan 004

Output 4 - **Hash (sha256) of selected text** (0.004 seconds):

`3B5A1C6DAC4B097E69A04F0E7892418DB400203B8B89BE0ECFD4D7681A24D566`

Input 5 -> Declan O'Riordan 005

Output 5 - **Hash (sha256) of selected text** (0.003 seconds):

`A7A872A935F147CFB8583467D3972D7478FF37DE3A8F292F9127DAC1C98A2C89`

Input 6 -> Declan O'Riordan 006

Output 6 - **Hash (sha256) of selected text** (0.005 seconds):

`79BBB3FD7ED9603F6EF612D8BE225F0F40B942B546516F3E1BA30757F7B5889F`

Input 7 -> Declan O'Riordan 007

Output 7- **Hash (sha256) of selected text** (0.006 seconds):

0A70C950DFED554854EFCB8E2F92927D77F8D8576C3AF4CFB3008349916589AA

0A70C950DFED554854EFCB8E2F92927D77F8D8576C3AF4CFB3008349916589AA

I got lucky and found a leading zero after only seven tries. Next time it might take dozens of attempts. That's how randomness works. See how long it takes to find a leading zero using your own name.

Bitcoin Mining using SHA-256

In numerical terms, my goal in the hashing example above was to find a hash value less than 0x1000000000000000000000000000000000000000000000000000000000000000 (in *Hex*). We call this threshold the *target* and the goal is to find a hash that is numerically *less than the target*. **Each attempt is a Test**. Every time we decrease the target, the task of finding a hash that is less than the target becomes more difficult.

To help understand why decreasing the target increases the difficulty, imagine rolling a pair of dice. If your target is to score less than a total of twelve when the two dice scores are added together, anything except two sixes will be a success. Since there are six sides to each dice and the score on one dice does not influence the score on the other, there are six times six possibilities. The probability of rolling less than twelve is 35/36 or 97.22%.

Now imagine the target is reduced to a score less than three (i.e. both dice simultaneously showing one).

1 x 1 1

------- = ---

6 x 6 36

Probably only one throw out of every 36, or 2.78% of them will produce a winning result. The lower the target score, the lower the probability of seeing it occur. We can estimate the amount of work it would take to roll a dice target based on probability.

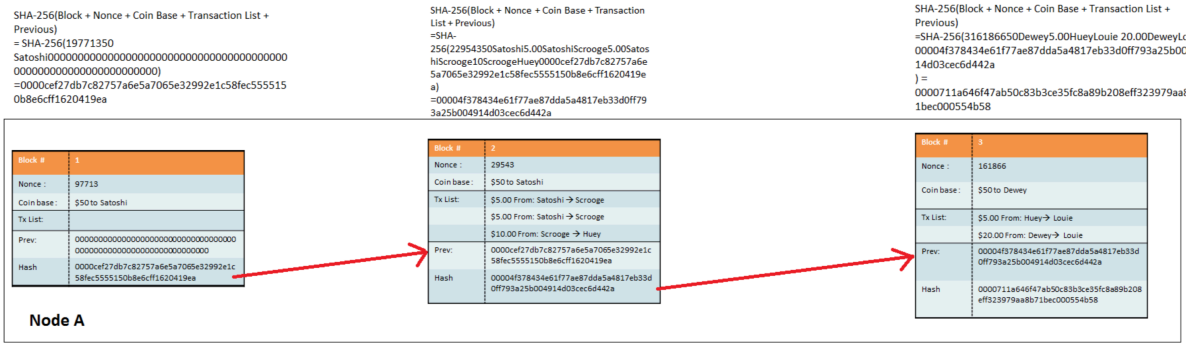
SHA-256 is a deterministic function and it is infeasible to generate an input message from its hash value except by trying all possible input messages, i.e. it is a one-way algorithm. If we set a target output, the only way to find the input that creates the target SHA-256 output is to start trying all possible input values until we get lucky or reach the last possibility. It takes a long time (and a lot of computing power) to work through enough possibilities to find a valid input, but once the matching input and output are presented, the validity of the result can be checked very quickly.

To produce a valid bitcoin block header hash and win the mining race, a miner needs to construct a candidate block filled with transactions, then use SHA-256 to calculate a hash of the block's header that is smaller than the current difficulty target. If they succeed, the miner includes the nonce that allowed the target hash to be achieved in their block header metadata. The nonce is then used by all the other nodes to quickly (in one operation) verify that nonce is the correct key to producing the target hash. If the miner fails to produce a hash less than the target they modify the nonce (usually incrementing it by one) and retry.

After the early days, one 4-byte nonce became insufficient to prevent miners cycling through all four billion values without finding a valid block header hash. Initially the block timestamp used in the block header was updated to increase the range of input values, but once mining hardware exceeded 4 GH/sec it became possible to exhaust all nonce values in less than one second. Advancing the timestamp more than two hours breaks the consensus ruleset described earlier. The solution was to use the variable coinbase transaction data (up to 100 bytes) as the source of extra nonce values.

On average, a bitcoin block is added to the blockchain every ten minutes. A scripted algorithm runs once every 2016 blocks (two weeks) to calculate if blocks are being added slower or faster than once every ten minutes. According to the result, the difficulty target for the Proof of Work algorithm is decreased or increased to maintain an average rate of one new block every ten minutes. The scripted equation in the bitcoin core code is:
New Difficulty = Old Difficulty x (Actual time of last 2016 blocks / 20160 minutes)

Every header references a previous block header hash to connect the new block to the previous block in the blockchain. By summarizing all previous transactions in the form of double-SHA256 hashes in the block header, a Merkle tree is built linking every block back to the Genesis block built in 2009.



Any attempt to cheat the blockchain by claiming a fraudulent transaction occurred can quickly be disproved by a Merkle path test, which connects specific transactions to the root of the tree if they have a valid hash. Establishing a fork in the Merkle tree requires a miner to win the race for Proof of Work and gain consensus from the majority of nodes. The further back in time the fork tries to establish itself the greater the depth of the blockchain and therefore the greater the difficulty to win multiple sequential Proof of Work races and alter established consensus that the main chain is valid. Currently it would be computationally infeasible to alter a block that was already six blocks deep in the blockchain because of the enormous work required to layer another five blocks on top again.

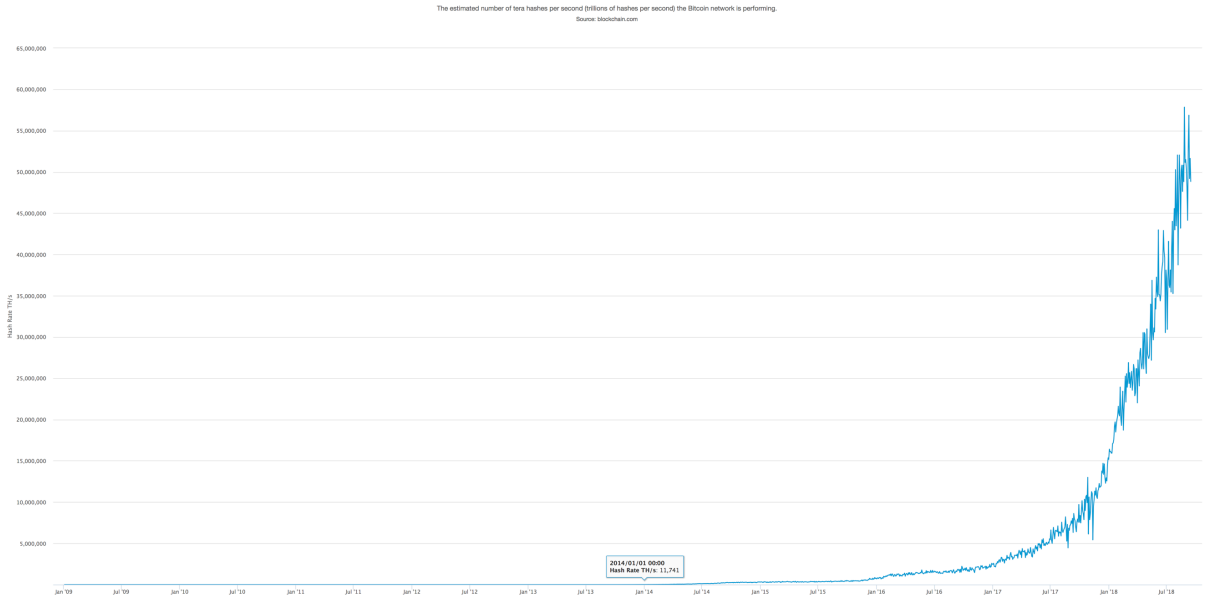
Mining speed

Bitcoin miners are not necessarily cypherpunks. Another of the ingenious bitcoin innovations was the creation of an incentive for complete strangers to invest energy in helping to build the blockchain by rewarding them with newly issued currency. Human nature and the desire of individuals to conduct commerce beyond the restrictions of Government control has driven innovation into new frontiers, as predicted by Nick Land's theory of Accelerationism. Solutions to problems that were recently estimated to be decades away have already been overcome by bitcoin developers and miners without any research funding, university collaboration, or large corporation support. Rumours circulate

that once entirely theoretical technology such as Instant Hypercube Routing and Infinite Sharding may have already been solved within blockchain communities. As classical computer components approach the nanoscale, the push inside atoms to Quantum Computing seems inevitable.

By any normal industry standard the growth in mining performance is extraordinary.
2009 - 0.5 MH/sec to 8 MH/sec (x 16 growth)
2010 - 8 MH/sec to 116 GH/sec (x 14,500 growth)
2011 - 16 GH/sec - 9 TH/sec (x 562 growth)
2012 - 9 TH/sec - 23 TH/sec (x 2.5 growth)
2013 - 23 TH/sec - 10 PH/sec (x 450 growth)
2014 - 10 PH/sec - 150 PH/sec in August (x 15 growth)

Looking at the current bitcoin hashing rate we should expect the incredible advances shown above to appear as huge spikes in the graph. Amazingly, everything earlier then mid-2014 appears to be a flat line compared to the recent exponential growth.



Initially in 2009, bitcoin miners used computers with conventional Central Processing Units (CPU) to calculate proof of work. Focusing on solving the Hashcash problem, miners switched to Graphical Processing Units (GPU) in 2010, then Field Programmable Gate Arrays (FGPA) in 2011. In 2013 the introduction of Application Specific Integrated Circuit (ASIC) mining lead to a giant leap in mining power, by placing the SHA-256 function directly on silicon chips specialized for the purpose of mining. The first such chips could deliver more mining power in a single box than the entire bitcoin network in 2010. Individual $3k ASICs now claim 16 TH/s (16 trillion cycles per second) performance, something that was not expected in 2012 to be achieved by research institutes before the year 2030 at the earliest.

Bitcoin and blockchain shine a light on future technology and development led by philosophy. What risks lie ahead?

PART TWO – RISKY ASSESSMENTS

For every difficult risk assessment, there is an answer that is clear, concise, and wrong.

The Root of the word Risk

The EU General Data Protection Regulation mentions 'risk' 75 times. For a word that has been around and widely used for a long while, there is surprisingly little common understanding of what the term risk really means. Most dictionaries define the English word *risk*, and also the Italian words *risico*, *risco*, *rischio*, the Spanish word *riesgo*, Portuguese *risco* and French *risqué,* as all deriving from the Latin words *resicum*, *risicum,* and *riscus* which mean *cliff* or *reef.*

The Latin words have Greek origins. In book 12 of Homer's epic tale, Odysseus is blown into unnavigable narrow waters between the six-headed monster Scylla and ever-thirsty whirlpool Charybdis. In the original cliff-hanger story, Odysseus survives by clinging to the roots of a fig tree hanging from one of the cliffs. Such roots were known as *rhiza* or *rhizikon* (possibly from Proto-Indo-European words existing about 5,500 years ago). The words rhiza and rhizikon then came to be associated with cliffs, and eventually took on the metaphorical meaning '*difficulty to avoid in the* sea' as the root symbolised Odysseus's predicament.

Over following centuries, the Arabic world adopted the Mediterranean term *rhiza* as *rizk*, meaning 'everything given by God for livelihood' i.e. something that cannot be totally controlled by mankind. By the time of the European Renaissance, risk had lost its seafaring meaning and the 16th century German business term 'rysigo' became 'to dare, to undertake enterprise, to hope for economic success'. Some etymologists believe the two Chinese symbols associated with risk are a combination of 'Danger' and 'Opportunity'.

What is Risk?

Leap forward to the 21st century and the International Organisation for Standardisation (ISO) 31000:2018 definition of risk is 'The effect of uncertainty on objectives'. But which effects, what uncertainties, and whose objectives? The objective of shareholders might be to see increased dividend payments, while the customers' objective is to pay as little as possible. The ISO definition is simultaneously irrefutable yet almost uselessly abstract. If risk is a type of uncertainty, we could at least narrow the scope of risk to 'uncertainty that matters'.

English is constantly evolving and the meanings of many words have changed over time. If they had not, the verb 'Test' would still be a noun meaning 'small vessel used in assaying precious metals'. Risk means many things to many people, but it cannot mean just anything.

A 2001 survey of risk professionals found 95% agreed that 'a risk' is an event. Nevertheless, I would argue that 'a risk' is actually an attribute of an uncertain event that matters. If an engine falls off an aeroplane it is an event, one risk attribute of that event is a possible rapid descent causing the loss of life to passengers, another risk attribute would be possible damage to anything or anyone the falling engine lands upon. If risks are considered to be just events, their causes are less connected to their potential consequences, making risks harder to control.

We might express risks as follows: "As a result of <*an event*>, <*attribute*> may occur, which would lead to <*negative or positive outcome to someone who matters*>". For example: "As a result of <*engine falling off aeroplane*>, <*rapid descent*> may occur, which would lead to <*physical harm to passengers and crew*>.

The ISO 31000:2018 standard tells us risk is usually expressed in terms of risk sources, potential events, their consequences, and their likelihood. Those terms are expanded as:

- Risk source - element that alone or in combination has the potential to give rise to risk.
- Event - occurrence or change of a particular set of circumstances. An event can also be something that is expected which does not happen, or something that is not expected which does happen.
- Consequence - outcome of an event affecting objectives, expressed qualitatively or quantitatively.
- Likelihood - chance of something happening: The word 'likelihood' is used to refer to the chance of something happening, whether defined, measured or determined objectively or subjectively, qualitatively or quantitatively, and described using general terms or mathematically, such as a probability or a frequency over a given time period.
    - The English term 'likelihood' does not have a direct equivalent in some languages; instead, the equivalent of the term "probability" is often used. However, in English, 'probability' is often narrowly interpreted as a mathematical term. Therefore, in risk management terminology, 'likelihood' is used with the intent that it should have the same broad interpretation as the term "probability" has in many languages other than English.

<u>Managing risk:</u>

The options available to manage risks are:

1. Avoid the risk by deciding not to start or continue with the activity that gives rise to the risk;
2. Accepting or increasing the risk in order to pursue an opportunity;
3. Removing the risk source;
4. Sharing the risk with another party or parties (including contracts and risk financing or insurance);
5. Retaining the risk by informed decision;

6. Changing the likelihood;
7. Changing the consequences.

Testing is clearly connected to options six and seven. By providing relevant and timely information, testing helps monitor the effectiveness of actions taken to control risks. Testing also helps identify risks to support all potential options. Testing alone is not equivalent to risk management.

Problems with risk assessments

On the face of it, risk management might appear a straightforward by-the-numbers business. Nevertheless, whether risk assessments are approached with a bottom-up component-focused methodology (e.g. ISO 27005), or a top-down system-focused holistic method (e.g. Attack Trees), it is important to recognise there are many limitations in all known approaches to risk.

For readers interested in taking a more professional approach to risk advice by reducing over-confidence and 'one size fits all' thinking, I have created a 'Top Ten' flaws in risk assessments.

1) Misidentifying risks

Having no clear understanding of Risk is likely to result in failures to identify real risks, and mistakenly identifying matters that are not risks. Some risk-related events are binary (they happen or they don't), while others involve multiple discreet events that vary across a range, and not necessarily in an intuitively determinable way.

While risk is a type of uncertainty, not all uncertainties are risks. For example, a possible change of Government in India is unlikely to affect an IT project in China. Conversely, a change of currency exchange rate could have a positive or negative outcome for an offshore project, depending upon which side of the transaction you sit. A risk management process could use financial derivatives such as options or futures contracts to manage exchange rate risk over the timescale of the project, or the risk could be accepted in order to pursue currency speculation. If the focus is on 'usual suspects', opportunities may be missed while real risks are unmanaged and left to chance.

When the identification step mis-identifies risks, subsequent stages are doomed to failure. Resources will be wasted managing irrelevant ideas and the credibility of risk management will suffer.

A note of caution here: Genuine risks are frequently identified and managed, but do not occur during an arbitrary time period. Over time, scepticism may lead to a mindset that such risks should be excluded from the risk management process. As Richard Feynman observed during the space shuttle Challenger explosion enquiry: "Much of the reasoning about risk at NASA effectively took the form that if disaster hadn't happened yet, it probably wouldn't happen next time either".

2) Masking uncertainty.

Unlike spatial distances between objects, we do not perceive time through a conventional sense, yet we notice time through the perception of other things. In humans, the subjective perception of time passing alters with increasing age and varies between individuals. Within five milliseconds (thousandths of a second) intervals, we perceive visual events to be simultaneous. Apart from when we look up into deep space and see the distant past, everything we experience as present time has occurred in the recent past and is therefore unreal. Our eyes only contribute around 10% of what we see. Our brains construct the other 90% because we cannot process all of reality. Instead we see 'meaning'.

The finite speeds of light and sound, plus the movement of signals from our senses through our minds create a delay between reality and our sense of the present. These considerations may seem irrelevant since light in a vacuum travels approximately one metre every 3.34 nanoseconds (billionths of a second). However, at the time of writing, bitcoin miners are testing 62 billion hashes per nanosecond and the rate is increasing exponentially. As we shall discuss in part 3 of this paper, time and time-complexity are too poorly understood to appear in conventional risk assessments and are therefore simplified to the extreme.

The past may influence the future, but the future cannot influence the past. We have mental mechanisms for recalling memories of past events (not entirely accurately), but we cannot sense the future. Despite these obvious truths, almost all risk assessments include a prediction of the future in the form of a 'likelihood' factor.

Risk calculations containing 'likelihood' in the expression airbrush over the practical difficulty of predicting the future. The more formulated the risk assessment, the more convincing the outputs become that it is actually possible to predict the causes and effects of all risks with a degree of certainty. It is implied that the same inputs will always lead to the same outputs (determinism). But because we are dealing with the interaction of people and technology we should not assume all interactions are predictable. There will always be some level of uncertainty about the outputs from any risk assessment technique.

The complex nature of modern technology systems means risks will often emerge that were not previously anticipated through assessment and analysis techniques. Failing to recognise this uncertainty (and the non-deterministic nature of risk) can lead to complacency and lack of preparation for emergent or changeable risks.

Continuing to use NASA as an example, the space shuttle Challenger inquiry revealed the engineers estimated the chance of disaster as one in one hundred, while managers thought them to be closer to one in one hundred thousand. Management over-ruled the engineers yet the loss of two shuttles (and 14 astronauts) in 135 flights highlights the inaccuracy of the dominant prediction.

Past events are not always a good predictor of future events. A statement of probability, (especially when guesses are expressed as percentages) can bias decision makers and lead them to place unfounded confidence in a prediction. Probability assertions should not be

read as announcements of surety; they are suggestions to reduce uncertainty in support of risk management decision-making. Eradicating uncertainty is unrealistic.

3) **Abstraction of reality.**

Whenever reality is reduced to labels, names, numbers, matrices, or any artificial construction, the subtlety and complexity of risk is often concealed by low resolution. A qualitative label such as 'High' is still vague and liable to various subjective interpretations. A quantitative label such as 4% of global turnover conceals the many associated costs and missed opportunities that a loss of that size would incur (e.g. redundancies, plant closure, R&D cutbacks, market retrenchment, reputational damage, loss of confidence, etc.) and the impact assessment would be subject to the bias of each decision maker.

Numerical labels are often perceived as being more reliable because they give the appearance of rigour. While this is sometimes true, numerical labels can promote bias because they are received with more confidence. For example, a risk labelled as 60% probable, will instil a greater sense of surety in decision makers, than if it were labelled 'medium-high'. While financial risks are usually best expressed in numerical terms, intangibles such as brand reputation generally translate poorly to quantitative risk assessment.

Using matrices to inform management decisions can hide the complexity of technology and true nature of the associated risks. Outputs are difficult to validate when a matrix hides the functions used to combine input components. The use of columns and rows creates an implicit impression that a scale exists, and a false notion that risks exist on a linear scale. For example, on a three-by-three risk matrix a 'High' impact appears to be three times worse than a 'Low' impact, with 'Medium' exactly half way between the extremes. In reality, the worse case scenario might be thousands of times more damaging than a 'Medium' impact.

Since there is no demonstrably valid axiom to guide the use of risk labels or matrices, anyone might create a risk matrix quite different from anyone else's matrix, while assessing the same risks. The greater the abstraction through labelling, the more meaning and context are lost.

4) **Losing risk signals in the 'noise'.**

Normal system operation generates noise in the form of alerts and notifications received through monitoring tools and via customer contact points. The volume of noise makes it difficult to separate acceptable activity from genuine risk signals. The problem can be compounded by situations where corresponding signals imply validation and are then input to risk decisions.

Noise can be created by misguided risk analysis based upon poor scoping and modelling. In extreme cases, the risk model may only contain noise and no genuine risk signals. This scenario may arise when a regulatory standard reduces the scope of a risk model to compliance alone.

Noise can also be generated through bias. The source of the bias may be the latest news headlines regarding a security vulnerability or disaster, causing the focus to be upon the noise while real risk signals are overlooked.

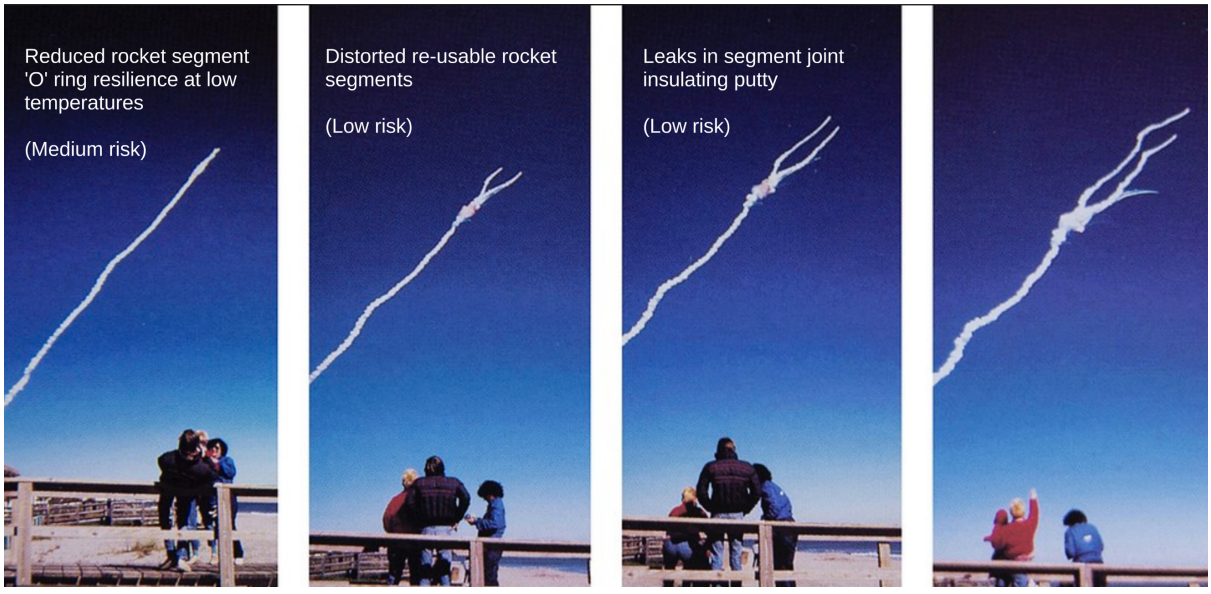
Filtering out normal system noise and ‘false-positives’ from real risk signals requires skill and good judgement. Too much filtering or modification of signals as they travel through the organisation creates information opacity, followed by loss of understanding. The greater the depth of the workforce information hierarchy and more complicated the signals, the more detached decision makers will become.

5) Missing the connections.

Typical risk assessments consider each risk in isolation and break down the components into source, event, consequence, and likelihood. This reductionist approach can lead to a fixation on individual parts while missing the risk at large.

When risks go bad there is seldom a single cause of disaster. Complex interactions between components can create a compounding effect whereby the total risk is greater than the sum of the parts. Using our space shuttle example again, two low risks (distorted re-usable rocket segments & leaks in segment joint insulating putty) combined with a medium risk (reduced rocket segment ‘O’ ring resilience at low temperatures). In isolation, each risk may have been tolerable, but in combination they were fatal.

Rather than dismiss medium and low risks in the short term, decision makers should look for relationships between risks and consider how these combinations affect their estimations.



6) Scope blindness.

Once the scope of a risk assessment has been decided, the assessment will inevitably consider risks within this perimeter. Until the 1990’s it was possible for many technology systems to exist in isolation. Now systems are highly interconnected and increasingly complex. A serious IT failure in one enterprise may affect both suppliers and customers in

the supply chain. Small suppliers are particularly vulnerable to problems within their major customer.

Sony had a well-planned campaign in 2004 to launch the PlayStation 2 before Christmas. Unfortunately, an oil tanker became stuck in the Suez Canal and blocked all ships from China, including those carrying the PlayStation consoles. By the time cargo planes started flying PS2s into Europe it was too late for the Christmas rush, sales were down 90%, gamers were disappointed or switched brands, and retailers were powerless to influence events.

When UK phone company TalkTalk experienced a serious data breach the customer compensation scheme did not extend to victims of fraudsters using the stolen data to gain the customers' confidence and extract funds from their bank accounts. Quantifying the embarrassment of staff and frustration of customers was probably beyond the risk assessment scope, and the record fine was possibly not within the expected range!

The true impact of risk consequences can extend far beyond the scope of an assessment.

7) The Effect of Time.

Estimations made at the outset of risk assessments are not always revisited to consider the effect time is having upon the components. Regrettably, risks tend not to have a constant probability distribution over time.

A system may be considered secure on day one but a new vulnerability is publicly disclosed on day two. The plan for applying a fix immediately becomes a race between organisational efficiency and attackers developing detection and exploit kits. If the systems are successfully attacked the cost might be correctly anticipated at €1m per day. However, if the attackers cannot be dislodged after many days the organisation may reach a tipping point and totally collapse, causing the eventual financial costs to far exceed the predicted daily rate.

Risk controls need to adapt in response to changes in, for example, threat, technology and business use. Most existing approaches to mitigation specify the application of a fixed control set which does not consider 'real world' feedback. This feedback is essential for the effective regulation of technology systems. Feedback can inform the amplification of mitigation activities in situations where increased assurance is required, and the dampening of mitigation activities in situations where they are becoming excessive.

Decision makers should ensure assessments are kept current to include the effect of time on risk.

8) Meet in the Middle.

Sticking one's neck out and bringing bad news, especially when there is uncertainty, can affect the messengers' career in a work environment hostile to pessimism. Without solid objectivity it is tempting to resort to safe subjectivity and rank most risks as 'medium'.

Unfortunately, a glut of medium risks hinders any attempt to prioritise treatment, assuming the assessment criteria were trustworthy in the first place. Granularity is essential for prioritisation of treatment and to avoid meaningless risk decisions. Effective risk

management is less about trying to calculate absolute values for risks and more about determining the optimum priorities when working with a limited budget. To be useful, models of risk must be honest.

9) Lop-sided variety.

Technology systems are built and delivered with increasing complexity, innovation, and variability. The options for controlling risk tend to evolve slowly and with less variability. The end result can be a lop-sided equation with a limited approach to mitigating risks versus an almost unlimited variety of technology.

While it may feel safe to stick with a control set recommended by an established risk assessment method, a more effective approach is to employ equivalent variety in risk mitigation as the dynamics of the technology system introducing the risks.

10) Treatment can create risks.

Interventions do not always deliver certain outcomes. Sometimes they can have adverse effects of their own. 'Fixing' system defects can cause regression. Removing Middle-Eastern tyrants can worsen already bad situations. The risk management policy might set deadlines for security patches to be applied within service level agreements, yet without adequate testing these patches may create new vulnerabilities, perhaps worse than the originals.

Risk assessments do not always consider the possible adverse effects of planned interventions. Look before you leap!

PART THREE – THE HIDDEN RISK TO BLOCKCHAIN

Parts three of this paper will consider how masking uncertainty, scope blindness, missing the connections, abstraction of reality, and the effects of time, hide a real risk of catastrophic failure to all blockchains. As an independent risk advisor I have no conflict of interest in my analysis. This paper does not provide any financial advice to speculators.

Some risks are difficult to understand and difficult to explain. Testers who hope those risks disappear if they don't think or talk about them have already taken the blue pill. Part three is for Testers considering taking the red pill and willing to test ideas. One day, the most important unsolved problem in computing might be solved and the answer will have a direct effect on the viability of machine-to-machine testing. If you could care about risks in the gap between computer science theory and practice, read on.

If you own a cryptocurrency such as bitcoin there are some domestic risks to be considered.

- Firstly, possession is ten-tenths of the law. NEVER EVER mix the new decentralised model of security with the old centralised model.
- If you trust a third party such as a relative, lawyer, or crypto-currency exchange to look after your private key you run a real risk of being robbed.
- If you lose the private key to your wallet you will never be able to spend the coins associated with that key.
- If you reveal you own bitcoins you may be targeted by thieves.
- If you die without telling anyone your private key, your cryptocurrency legacy can never be inherited or spent.
- Transaction fees may make smaller purchases too expensive to be viable.
- The energy consumption of mining rigs has an environmental impact.
- In Turkey, Egypt, Saudi Arabia, Kuwait and Palestine, purchasing bitcoin has been deemed un-Islamic.
- Theoretically there is an attack vector upon consensus if the attacker (e.g. several corrupt administrators controlling large mining pools) control more than 51% of the hash-rate. There are also other theoretical attack vectors such as the race attack, Finney attack, Sybil attack, and Vector76 attack.
- Code updates to bitcoin core may contain errors if volunteer Testers inadequately test the changes (there are opportunities for Testers to get involved here).
- Recently, Border Gateway Protocol (BGP) attacks on cryptocurrencies have become realistic.
- If Governments become serious about destroying cryptocurrencies they are likely to try interfering with the Domain Name System (DNS) to prevent routing between peer-to-peer traffic. Core Reference Clients are dependent upon centralised resources for bootstrapping, which therefore presents an attack vector.

There is however, a fundamental risk that could undermine Proof of Work and all variations used by blockchains based upon asymmetry. Proof of Work is totally dependent upon the existence of computational asymmetry. In bitcoin, the level of effort to solve a problem (calculate a valid hash and provide the nonce) must be high, but the level of effort to check the solution (use the nonce to check the hash is valid) must be low.

The risk begins with the assumption that Hashcash is a Nondeterministic Polynomial (NP) time problem. In other words, Hashcash belongs to the same complexity class as the Travelling Salesman Problem. NP problems have the characteristic of being hard to solve yet quick to check the result. Imagine a Sudoku game in which the number of rows and columns can be increased any number of times to ensure you spend a long time solving it, but once you have completed the task it is quick to check if your solution is correct. Jigsaw puzzles are also NP problems. The only way to be sure a pile of jigsaw pieces build a complete picture is to try fitting every piece into place. At the end of the task it is instantly obvious if the jigsaw is complete. Making the best move in a game of chess however is not an NP problem because it is hard to decide which move to make and also hard to verify it was the best move that could have been taken. Chess is an Exponential time problem.

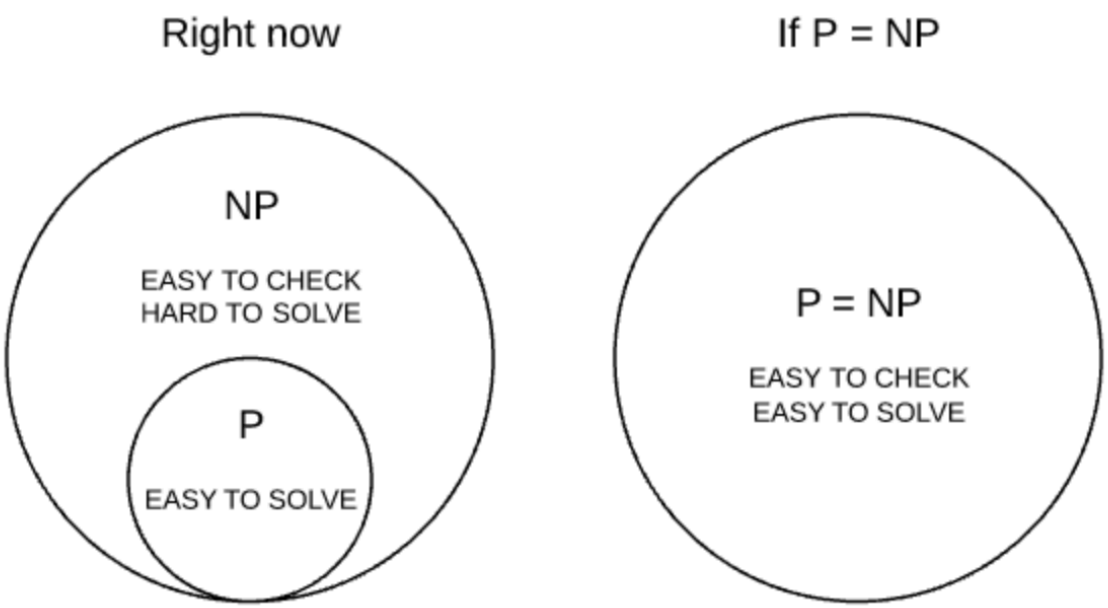
We shall follow the general assumption that Hashcash, and therefore Proof of Work, is an NP problem. Now comes the biggest assumption of all, one that is implicitly made by all blockchains:

P ≠ NP

P represents Polynomial complexity problems such as addition and multiplication, for which there exists a polynomial time algorithm that generates a solution. i.e. can be solved 'quickly'.

NP represents Nondeterministic Polynomial complexity problems such as Rubik's cube and prime number factorization, which consist of two phases: Firstly guess the solution in a non-deterministic way; secondly verify or reject the guess using a deterministic algorithm that is performed in polynomial time.

All P problems exist within the set NP, but no-one has been able to prove if P problems could be equal to NP, or definitely not equal to NP. The working assumption adopted by all blockchains is that P does not equal NP. It takes more time to solve Sudoku problems than check the answer, so surely P ≠ NP right?



P vs NP is not discussed by Testers, yet it is the greatest unsolved problem in computer science, and possibly all of mathematics. It is widely discussed in unexpected places. In The Simpsons episode 'Tree House of Horror 6' executive producer David X Cohen planted an image P = NP in the background. But in Futurama, staff writer Jeff Westbrook (a Yale Professor of Computer Science) placed two folders labelled P and NP on a shelf with a space between them, indicating he disagreed with Cohen and they are not equal.

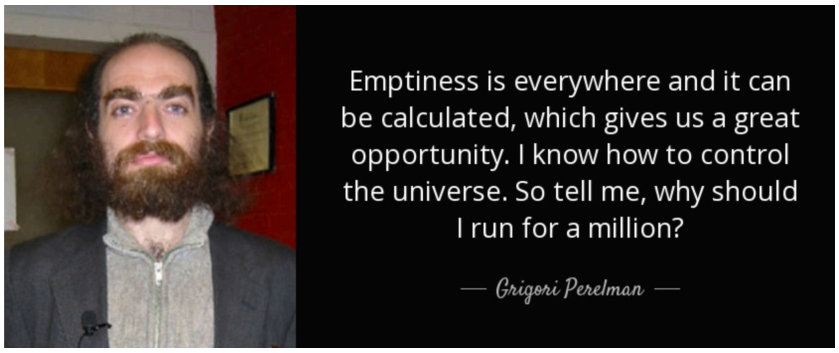
The P versus NP problem asks whether every problem whose solution can be quickly verified (technically, verified in polynomial time) can also be solved quickly (again, in polynomial time). Incredibly, if any NP-complete (i.e. harder) problem such as subset sum or the travelling salesman problem can be solved in polynomial time, then all NP problems can be solved in polynomial time and P = NP.

The implications are so enormous the Clay Mathematics Institute set a one million US dollar prize for providing a proof that either P = NP, or P ≠ NP. It is one of seven Millennium Prize Problems set on 24th May, 2000. The full set of problems are as follows:

- Poincaré conjecture (Solved)
- **P versus NP**

- Hodge conjecture
- Riemann hypothesis
- Yang–Mills existence and mass gap
- Navier–Stokes existence and smoothness
- Birch and Swinnerton-Dyer conjecture

The Poincaré conjecture was unsolved for one hundred years and thought to be unsolvable until Grigori Perelman announced he had an affirmative solution using Ricci Flow in 2003. He was awarded the Fields Medal (the equivalent of a Nobel prize for mathematics) and a $1 million Millennium prize but refused to accept them stating: "*I'm not interested in money or fame; I don't want to be on display like an animal in a zoo.*"



Grigori Perelman lives in poverty with his mother in a Soviet era apartment block in St. Petersburg. He provides an example of the extraordinary mental qualities associated with solving a Millennium Prize problem.

Solving P vs NP would be the hardest $1 million you will ever earn. Fortunately all six remaining Millennium Prize Problems are related and within the same complexity set. Therefore a proof that P = NP could be applied to solve the remaining five, netting you $6 million and landing you almost any job in the world you want to take.

Because P versus NP has been outstanding since the 1950's an increasing majority of mathematicians and computer scientists believe either P ≠ NP, or no proof will ever be found. There may be a similarity with Fermat's Last Theorem which was unsolved for 358 years and resisted every attempt at solution, leading to a consensus among mathematicians that it was unsolvable. Andrew Wiles read about Fermat's last theorem in a library when he was ten years old and devoted his life to finding the solution. Aged 41 and as a Professor of mathematics, he presented his proof and ensured a place in history. A solution to P vs NP is likely to stand alongside Isaac Newton's formula $E = mc^2$ in terms of significance to science.

What if P = NP?

Proof of Work depends upon computational asymmetry and is believed to be in the NP complexity set. Anyone able to solve Proof of Work in Polynomial time can avoid the cost and effort of working through all possible solutions by arriving at the target in a single step.

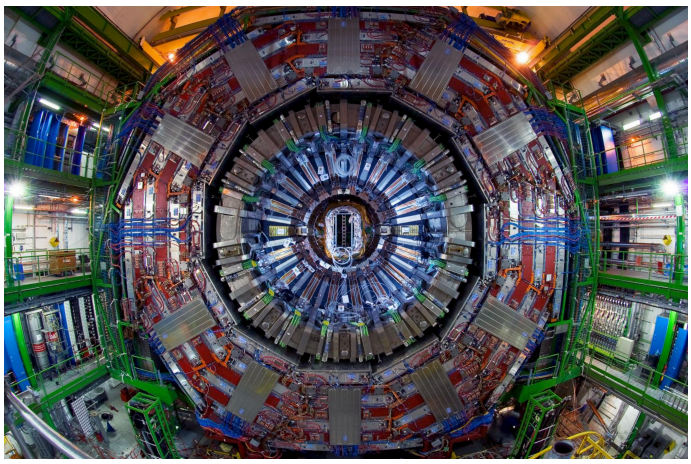
If attackers can submit valid-looking blocks to nodes with the correct SHA-256 header hash as fast as the blocks can be tested (i.e. in Polynomial time), emergent consensus is defeated. Multiple forks will appear in the blockchain, many containing fraudulent transactions created by attackers. With a powerful solution, attackers could create forks back in time and revoke 'immutable' transactions from the blockchain. Without computational asymmetry in Proof of Work it becomes simple to bombard blockchains with

denial of service attacks, or to win the mining race every time if one attacker has monopoly control of the P = NP solution.

There are further ramifications. If P = NP, every public key cryptosystem we have becomes solvable in Polynomial time. That would mean the end of privacy and secrecy as we know it. It would also be the beginning of a new era for technology, commerce, medicine, and science, as problems such as protein folding that would currently take computers millions of years to process could be solved inside a day.

How might P vs NP be solved?

With current mathematics it is unlikely a solution to P vs NP will be found. That still leaves the possibility that an extraordinary genius such as Grigori Perelman or Andrew Wiles will present a proof using sheer human ingenuity. NP problems are like looking for a needle in a haystack, which conventionally requires looking through the entire haystack until the needle is found. A P = NP solution does not require faster searching, it requires an approach that doesn't involve searching at all. Metaphorically speaking, a solution would be like pulling a needle from a haystack using a super-powerful magnet. If you can't yet imagine a magnet powerful enough to do that, you can either give up or remove all limits from your imagination until you have big enough magnet!

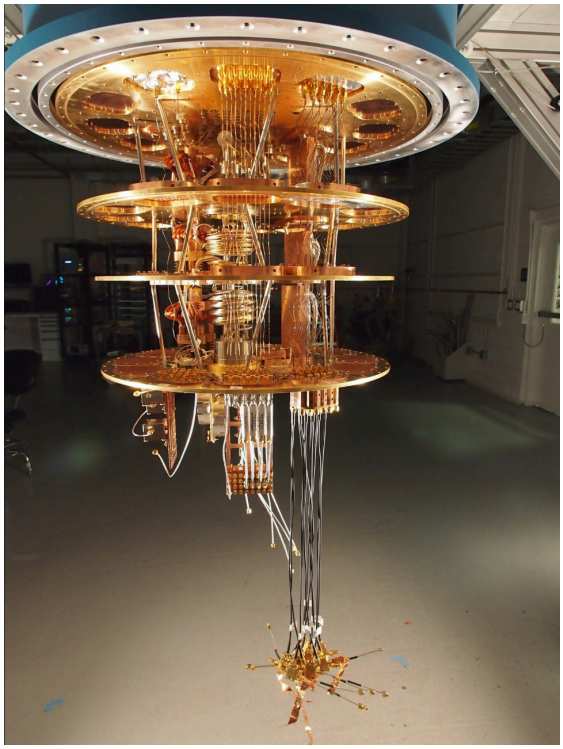


The fact that P vs NP has been unsolved for almost seven decades proves nothing other than it resists solution. It might be solved tomorrow.

If P = NP, yet we continue to build systems based on the assumption P ≠ NP, the consequences will be more serious as the number of systems built upon a false assumption increase over time.

Then there is Quantum Computing. Unless bitcoin mining forces improvements in fault tolerance and accelerates the delivery of quantum computers (which it might), another ten or twenty years may pass before classical computers begin to be replaced by quantum computers. By that time, blockchain systems could be commonplace and used with high confidence. Consider Richard Feynman's observation: "*Much of the reasoning about risk at NASA effectively took the form that if disaster hadn't happened yet, it probably wouldn't happen next time either*".

Once computing steps beyond the nanometre scale and inside the atom, the rules change. We will have entanglement, interference, superposition and decoherence to consider. Most importantly, answers will not be in a binary state. It may become possible to return many, perhaps all possible answers simultaneously. This might be a route to solving NP problems in polynomial time.



There has been speculation that since public key cryptography was discovered and kept secret by British and US national security organisations for several years before it was also discovered by independent researchers (Whitfield Diffie, Martin Hellman, Ron Rivest, Adi Shamir, & Leonard Adleman), a solution to P = NP might already be in secret use. Certainly we cannot expect GCHQ, the NSA, the Chinese Ministry of State Security, or Russian FAPSI to announce they have a mechanism to break public key encryption and spy upon almost any communication. We might assume from the Edward Snowden revelations that the NSA and GCHQ did not have the means in 2013, or maybe that just what they want us to think!

If a solution proving P = NP is published by a researcher as a public notice, the global breakdown of blockchain systems would begin to occur within the time required to implement the solution, perhaps in hours. Contingency planning and implementation for such an event is hard in any case, but almost hopeless without the benefit of time. Only organisations with P vs NP on their risk register would stand a chance.

You may work on blockchain systems in the coming years. There will be many conventional tests needed in blockchain core code and the peripheral systems such as wallets and payment channels. But the scope of your thinking doesn't need to be constrained to those limits.

In Conclusion

Testing provides information as part of risk management. Stakeholders don't like uncertainty but some might be persuaded to face the practical difficulty of predicting the future. If I were to ignore my own advice on mistakes made in risk assessment I would guess there is a 17% chance that P = NP. That looks small, but would you board a plane with a 17% chance of crashing? 17% has the appearance of precision, but it's just an informed guess hiding substantial uncertainty and non-deterministic nature of risk.

No-one knows if or when an answer will be delivered, but time is likely to erode the resistance of P vs NP to a solution.

If you believe in testing ideas and think Testers can provide more advanced and sophisticated advice to risk management, P vs NP is an example of what can be found when a Tester digs deep into unknown areas. To discover more examples, take the red pill and find some yourself!

Good luck to the red pill Testers and Risk Assessors!

References:

1. Scott Berinato: A Visceral Etymology of Risk
2. Vicente Sandoval: The Origins of the Word Risk
3. Risk Society: Ulrich Beck and Anthony Giddens
4. Dr David Hillson: What is 'Risk'? Results from a survey exploring definitions
5. & Top 10 myths of risk.
6. NCSC: A critical appraisal of risk methods and frameworks
7. ISO 31000:2018 Risk Management
8. ISO Guide 73:2009 Risk management – Vocabulary
9. The Orange Book: Management of Risk – Principles and Concepts (October 2004)
10. Andreas Antonopoulos. 2017. Mastering Bitcoin: Unlocking Digital Cryptocurrencies (2 ed.). O'Reilly Media, Sebastopol, CA, USA.
11. Nick Szabo. Shelling Out: The Origins of Money https://nakamotoinstitute.org/shelling-out/
12. A. Back. 2002. Hashcash - A Denial of Service Counter-Measure. http://www. hashcash.org/hashcash.pdf
13. M. Ball, A. Rosen, M. Sabin, and P. Nalini Vasudevan. 2017. Proofs of Useful Work. https://eprint.iacr.org/2017/203.pdf
14. Bitnodes. 2018. Global Bitcoin Nodes Distribution. https://bitnodes.earn.com/
15. Blockchain.info. 2018. Bitcoin Hash Rate. https://blockchain.info/charts/ hash-rate
16. Buybitcoinworldwide.com. 2018. Bitcoin Mining Hardware Comparison. https: //www.buybitcoinworldwide.com/mining/hardware
17. Coin Market Cap. 2018. CryptoCurrency Market Capitalizations. https:// coinmarketcap.com
18. C. Chantrill. 2018. State Spending for Arkansas. https://www. usgovernmentspending.com/year_spending_2018ARbs_19bs2n
19. D. Chaum. 1983. Blind Signatures for Untraceable Payments. In Advances in Cryptology: Proceedings of Crypto 82 (CRYPTO '82). Springer, Boston, MA, Santa Barbara, CA, USA, 199–204. https://doi.org/10.1007/978- 1- 4757- 0602- 4_18
20. W. Cook. 2016. Concorde TSP Solver. http://www.math.uwaterloo.ca/tsp/ concorde.html
21. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. 2009. Introduction to Algorithms, Third Edition (3 ed.). MIT Press, Cambridge, MA, USA.
22. W. Dai. 1998. b-money. http://www.weidai.com/bmoney.txt
23. O. Deutsch, N.R. Schiff, D. Dolev, and M. Schapira. 2018. Preventing (Network) Time Travel with Chronos. In Network and Distributed Systems Security Symposium (Proceedings of NDSS 2018). San Diego, CA, USA. http://dx.doi.org/10.14722/ndss.2018.23231
24. Digiconomist. 2018. Bitcoin Energy Consumption Index. https://digiconomist. net/bitcoin- energy- consumption
25. C. Dwork and M. Naor. 1993. Pricing via Processing or Combatting Junk Mail. In Advances in Cryptology: Proceedings of Crypto 92 (CRYPTO '92). Springer, Berlin,

Heidelberg, Santa Barbara, CA, USA, 139–147. https://doi.org/10.1007/ 3- 540- 48071- 4_10
26. H. Finney. 1993. Digital Cash and Privacy. http://fennetic.net/irc/finney.org/~hal/dig_cash_priv.html
27. Gapcoin. 2014. What is Gapcoin? http://gapcoin.org/index.php
28. M. Garey and D. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP Completeness. W.H. Freeman and Company, New York, NY, USA.
29. M. Jakobsson. 1999. Proofs of Work and Bread Pudding Protocols. In Secure Information Networks. The IFIP, vol 23 (CMS '99). Springer, Boston, MA, Leuven, Belgium, 258–272. https://doi.org/10.1007/978-0-387-35568-9_18
30. R. Karp. 1972. Reducibility among Combinatorial Problems. In IBM Research Symposia Series, Complexity of Computer Computations (CCS '72). Springer, Boston, MA, Yorkton Heights, NY, USA, 85–103. https://doi.org/10.1007/978- 1- 4684- 2001- 2_9
31 S. King. 2013. Primecoin: Cryptocurrency with prime number proof of work. http://primecoin.io/bin/primecoin-paper.pdf
32. A. Loe. 2017. Conquering Generals NP-Hard Proof of Work for Blockchain Construction. https://yadi.sk/d/PG8kvFzP3SnTcs
33. S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
34. Institute of Electrical and Inc Electronics Engineers. 2008. IEEE Standard for Floating-Point Arithmetic. https://doi.org/10.1109/IEEESTD.2008.4610935
35. N. Szabo. 2005. Bit Gold. http://nakamotoinstitute.org/bit- gold/
36. J. Tromp. 2015. Cuckoo Cycle: A Memory Bound Graph-Theoretic Proof-of- Work. In Lecture Notes in Computer Science, vol 8976 (Financial Cryptography and Data Security). Springer, Berlin, Heidelberg, San Juan, Puerto Rico, 49–62. https://doi.org/10.1007/978- 3- 662- 48051- 9_4
37. Litecoin Wikipage. 2015. Litecoin Scrypt Hashing. https://litecoin.info/index.php/Scrypt
38. F. Zhang, I. Eyal, R. Escriva, A. Juels, and R. van Renesse. 2017. REM: Resource Efficient Mining for Blockchains. In 26th USENIX Security Symposium. Springer, Boston, MA, Santa Barbara, CA, USA, 1427–1444. https://eprint.iacr.org/2017/ 179.pdf