

# Scalable DeSecure ECCPoW Blockchains



Heung-No Lee, GIST, South Korea  
Home page: <http://inonet.gist.ac.kr>  
Facebook/ Publication ID: Heung-No Lee



## ● Abstract

In the year 2018, we have witnessed the surge and the fall of crypto-currencies. With the surge, blockchain the new technology behind cryptocurrencies, and its idealistic footprint of advanced thoughts, blockchainism it can be perhaps called, came to enthrall our minds. Thousands of new ambitious projects have been conceived and fast activated with the worldwide frenzy of new funding through initial coin offerings a novel funding mechanism in the blockchain world. Decentralized societies, equal accesses to valuable resources, reducing the cost of middleman, freed individuals from hierarchical organizations, and reducing the spread in inequalities are some of those advanced thoughts. But the fall came; the market value for Bitcoin has collapsed more than 7 times from its peak-value; that of Ethereum has plummeted more than 12 times. These two power houses which have supported those progressive projects are now torn apart. Recent New York Times report reads, “Blockchain: What’s it good for? Absolutely nothing, report finds.” Another one reads, The Blockchain Is a Reminder of the Internet’s Failure. The same utopian promises that bloomed during the Internet’s early days are back. Be afraid.” Should this be the end of our pursue to change and make a better world with blockchains? Obviously not. In this presentation, I would like to talk about the reality of blockchain technology and how distant it is from the ideals. With this assessment, I would like to present some of novel research progresses we made in year 2018 and talk about further research ideas to pursue in year 2019 and beyond.

# Short Bio of Dr. Heung-No Lee

Heung-No Lee graduated from University of California, Los Angeles (UCLA), U.S.A. with Ph.D., M.S., and B.S. degrees all in Electrical Engineering, 1999, 1994 and 1993 respectively. He has written more than 70 international journal publications and a hundred international conferences and workshop papers. He worked at HRL Laboratory, Malibu, California, U.S.A., as Research Staff Member from 1999 to 2002. He worked as Assistant Professor at the University of Pittsburgh, Pittsburgh, Pennsylvania, U.S.A. from 2002 to 2008. He then moved to Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2009 where he is currently tenured full professor. His research lies in the areas of Information Theory, Signal Processing Theory, and Communications Theory, and their application to Communications and Networking systems, Biomedical systems, and Signal Processing systems. Awards he has received recently include Top 50 R&D Achievements of Fundamental Research in 2013 (National Research Foundation), Top 100 National R&D Research Award in 2012 (the Ministry of Science, ICT and Future Planning) and This Month Scientist/Engineer Award (National Research Foundation) in January 2014. He was the Director of Electrical Engineering and Computer Science within GIST College in 2014. Administrative positions he has held at GIST include the Dean of Research and the Director of GIST Research Institute.



# Talk today

- This talk shall focus on the open public blockchains but not on the private blockchains.
- Blockchain Ideals
- Reality
- Future
- Summary

# Bitcoin, What it is?

- A global computer network run by people which **mints coins every 10 minutes**.

## 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.

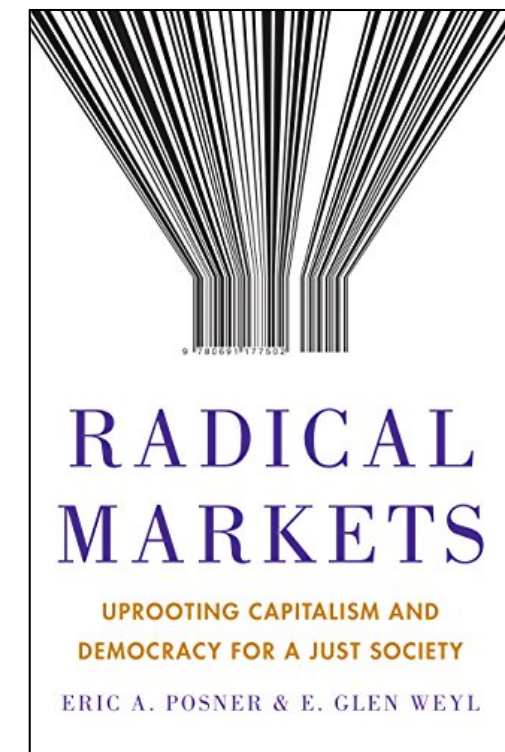
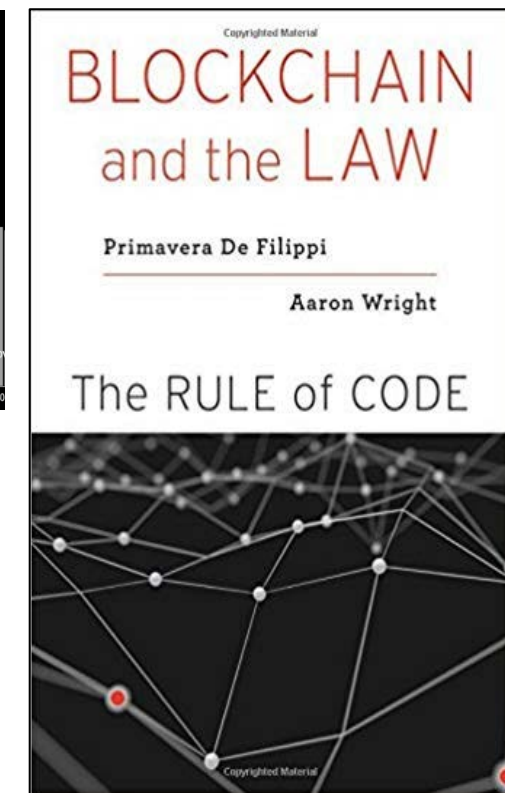
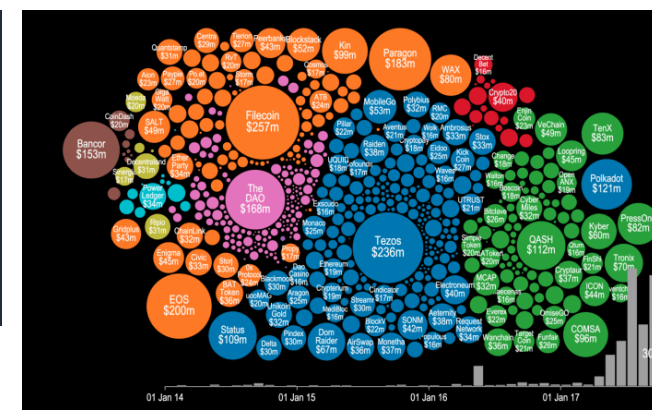


# Bitcoin's Ideals

- Since born in 2009, bitcoin has never been stopped breathing and alive currency system.
- Global digital currency works beyond national boundaries.
- It was the time when trust on the banks and governments were severely degraded.
- Ideals around bitcoin are
  - Decentralization
  - Reforming Wall Street
  - Unbundling big corporations
  - Reducing inequality
  - Sharing economy



# Ethereum's Ideals



- Ethereum network allows not only coin TXs, but also doc files and computer codes.
- A *decentralized* app (Dapp) runs a front end code; a backend code runs in *the ENet*.
  - ✓ Cf) For an app, the backend code is running on *a centralized server*.
- Computer codes can a reflect lawful contract among people.
- A smart contract
  - ✓ Computer codes can be executed and advanced to the next stage each time a contractual term matures.
- Decentralized autonomous organization has its bylaw written in the smart contract.
- The organization spend tokens and make governance decisions w.r.t. smart contracts.
- *Lex Cryptographia!*
- *Uprooting capitalism and democracy for a just society!*

# Blockchain Reality



# Blockchain Core (Program Suite)

## Network of peers

- Node registration, get-address, give-address
- Full nodes, light nodes, wallet nodes

## Wallet for TX generations

- Make private and public keys, addresses, store UTXOs, make TXs, put signatures, announce TXs to the neighbor, check to see if TXs are supported by the blockchain.

## Miners guard the blockchain

- **Data**: Genesis block, regular blocks, one block every 10 min, block-size 1Mbyte
- **Protocol**: consensus, block header, difficulty level adjustment, ...
- **Mining**: Get the longest chain, **validate** it and all transactions within it, get transactions from mempool and form a block, **run SHA repeatedly until you hit a good hash**, put the proof into the block header, and attach the proofed block to the longest chain, and make announcement ASAP.

## Web server interface

- Communication among the nodes, wallets and the miners

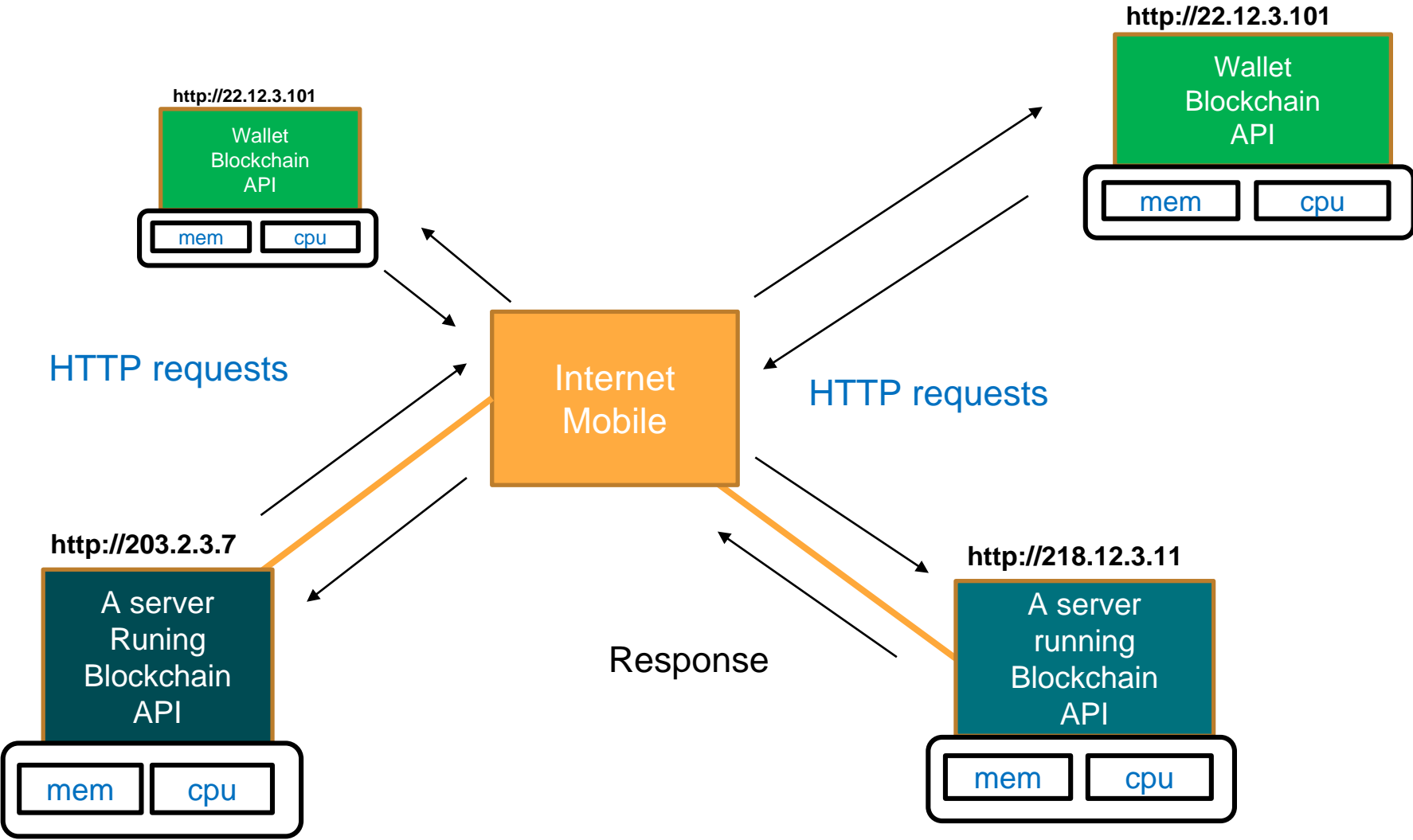
## Program Suite

- C++, Python, Go, Java, Flask, http
- Download and run into your computer, then you have a blockchain server.


# Blockchain after-all is an open computer server network.

- A digital ledger in which the content of TXs is recorded right away as soon as they occur.
- A technology which keeps what's recorded in its original forms.
- What to record in the distributed ledger?
  - Coin transactions (Bitcoin) → **crypto currencies and tokens**
  - Important content → **public record house**
  - Computer code and execution → **code executing computer**
- Blockchain is a distributed server network
  - Computers can run various App's and thus support various services.
- Decentralization: Getting rid of middleman → But we all need to do some work and contribute!
- Incentivization with cryptocurrency
  - Decentralized cooperation, trust by decentralization, born of coin economy!

Anybody who downloads and runs the Blockchain.core can become a member of  
**a blockchain internet**



# Reality

- **Re-centralized**: a few mining sites dominate!
  - **Scalability Trilemma**: if aiming to increase TSP, you shall give up either security or decentralization!
  - Not even just: a few people are predominant in shares of coins.
  - No killer DApps as of yet, why?
  - 98% of ICOs in 2018 turns out to be scams or at least did not deliver what's promised!
  - Crypto-exchanges are full of pump&dump and insider trades!
  - Smart contracts is one and a lawful contract is another!
- 

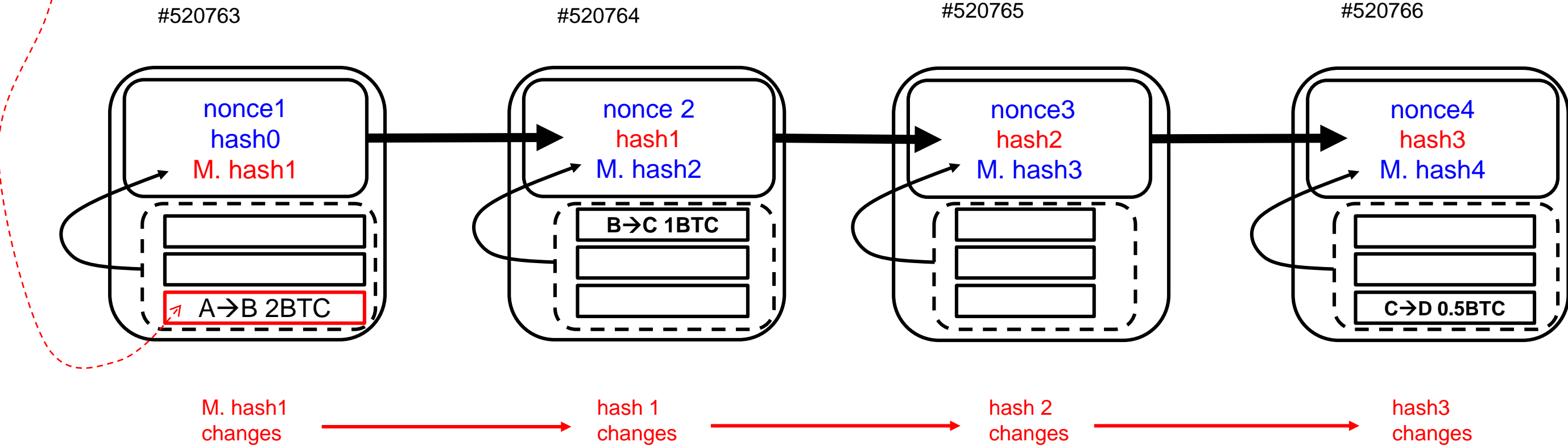
# Future of blockchains

- No more easy funding, at least not only with a white paper!
- White papers are not peer reviewed nor validated are the ideas!
- Investors' expectation shall not be wasted. It is lucky to have many investors still very confident with the potentials of blockchain and cryptocurrencies.
- Blockchains are after all games of gathering autonomous people with a common interest who are willing to give their funds, talents, computing power and time for making the society a better place to live than before.



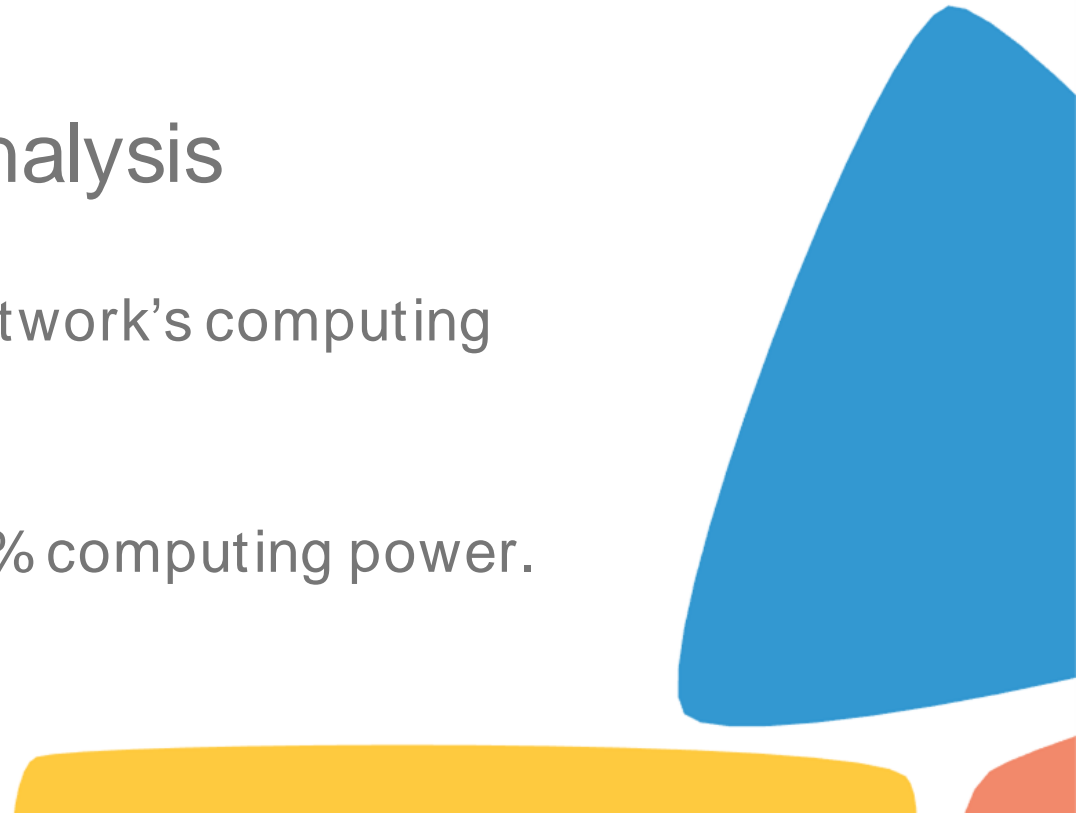
# Blockchain Immutability? Where is it from?

- What is blockchain?  
Series of blocks containing Txns and time-stamp?
- What happens when any alteration is made?
- What if there is no Proof-of-Work (PoW) attached?





# Two novel approaches to resolve the issues of re-centralization and Trilemma!

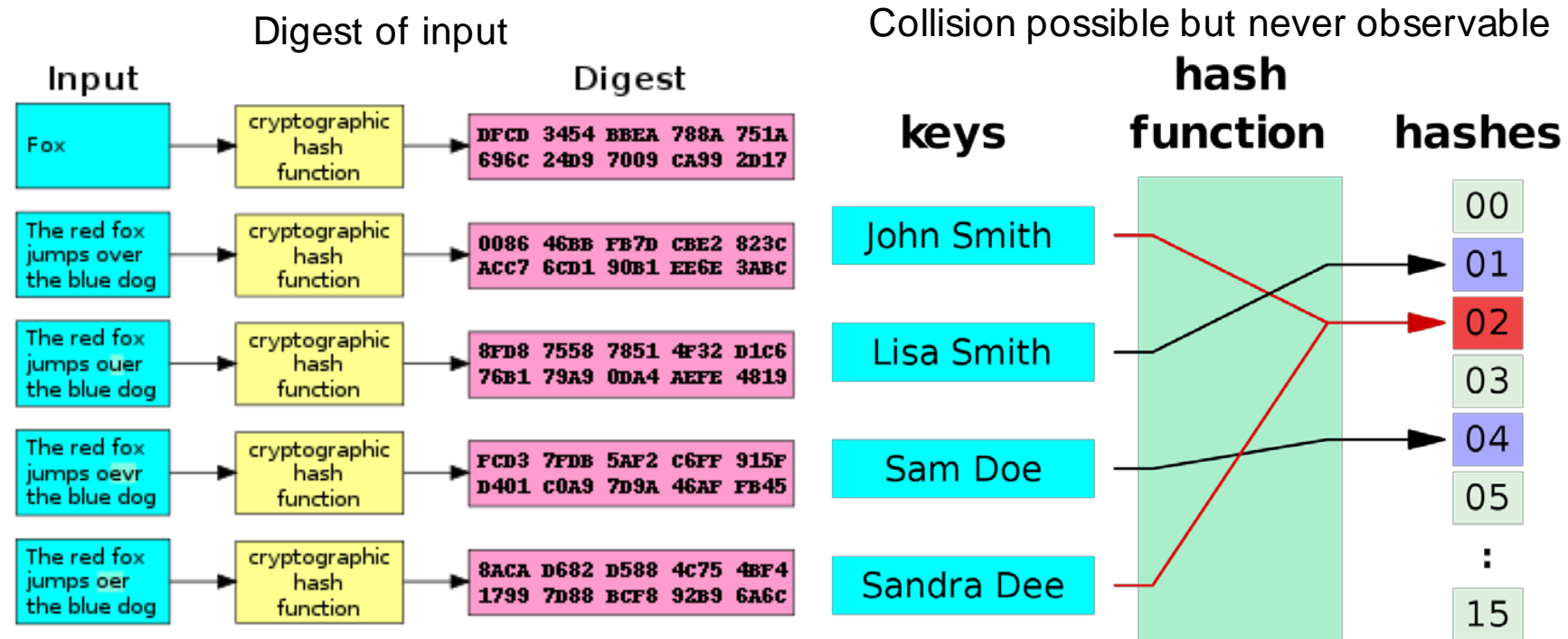
- Error-correction code base proof-of-work
    - This is to have the re-centralized blockchain networks decentralized again!
  - Novel profitable double spending (DS) attack analysis
    - It is well known that one needs to have at least 51% of whole network's computing power for launching a successful DS attack.
    - Our analysis shows that DS attacks are possible even under 50% computing power.
- 

# Properties of Proof-of-Work (PoW)

- The proof of work is used to keep the ledger unalterable. It is the proof that all in the network worked together. For one to redo all the work done, **it shall take the same amount of time spent.**
- Properties of proof-of-work
  - P1: Easy to verify but difficult to prove
  - P2: Robust to detecting block modification attacks
  - P3: Controllable in changing the difficulty level
  - P4: Open to anyone with a CPU

# What is a Hash Function?

- Bitcoin uses SHA256 (NIST, USA)
  - Input to the hash function is a text message or a file.
  - Output of the hash function is 256 bit string.
- Conditions for Good Hash Function
  - (One way) **With any change in input, output is completely different.**
  - (Collision free) Given  $y = H(x)$ , finding  $x_1$  such that  $H(x_1) = y$  shall be almost impossible!
  - (Collision free stronger) Finding an input pair  $x$  and  $x_1$  which leads to  $H(x) = H(x_1)$  shall be almost impossible!



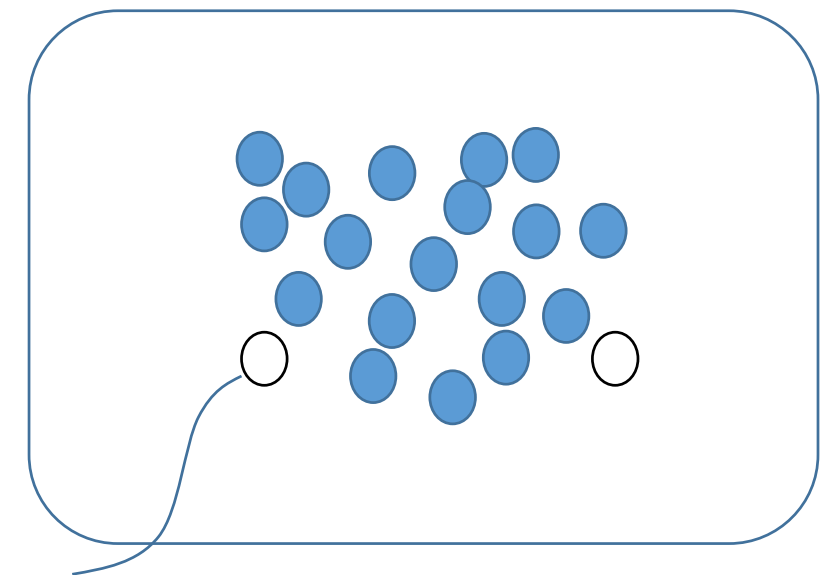
Pictures from google image

# What is Mining? Finding a good block summary!

- Let  $H(*)$  be the Hash Function.
- Function  $F$  takes an input  $x$  and gives output  $y = F(x)$ .
- $F(\text{block}) = \text{block summary}$
- (Proof-of-Work) Finding a nonce which produces a block summary meeting a goodness requirement:  
$$F(\text{block}, \text{nonce}) < \text{a certain value (PoW)}.$$
- Given a block, finding a nonce which satisfies the PoW inequality takes many hash cycles.
- Once *nonce* found, recording it in the block header completes the process of PoW.

There are 20 balls in total and 2 white balls.

What is the probability to select a white ball?

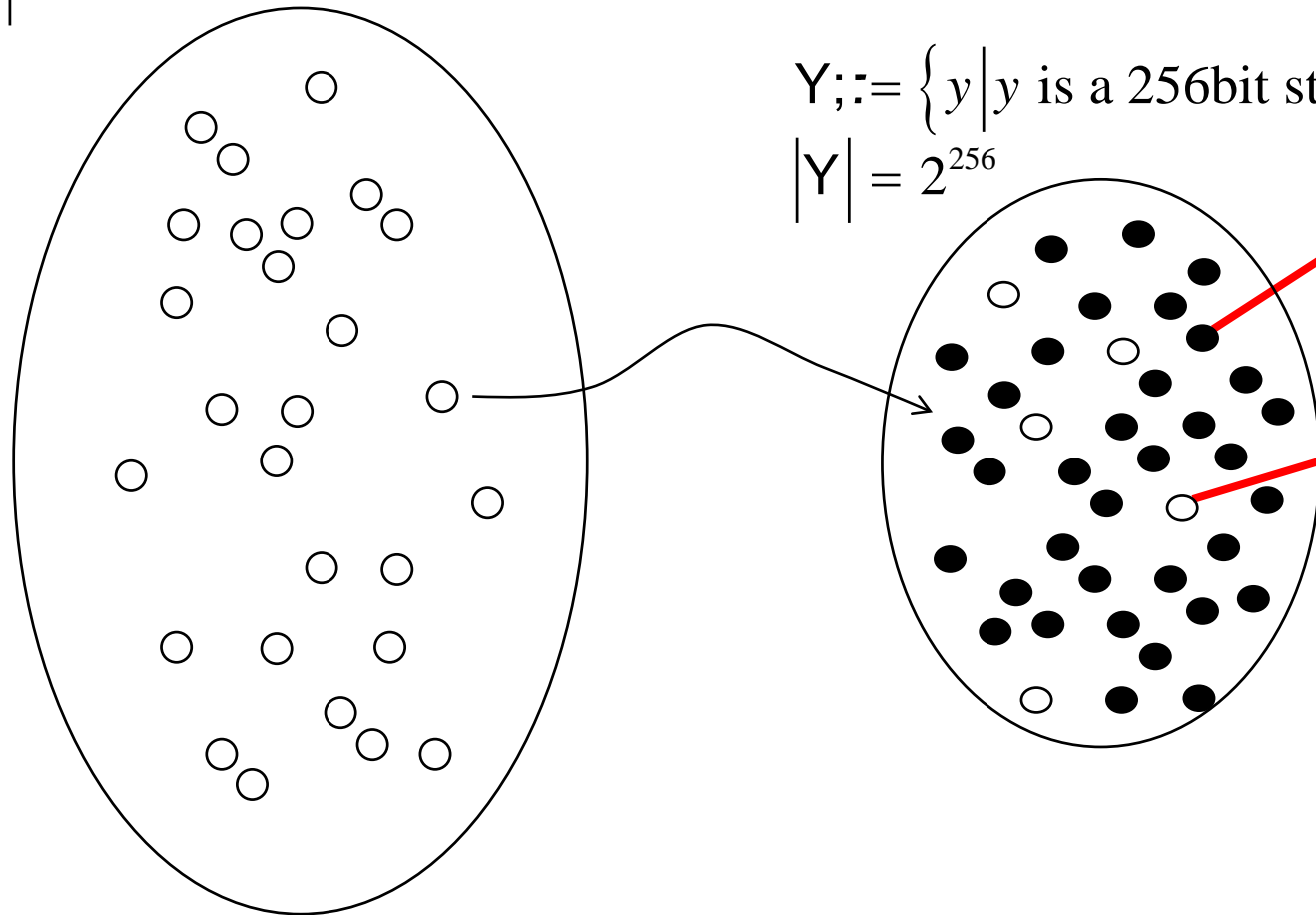


Output of the function

# What is the probability a cpu solves (PoW) in a single cycle, requiring the first four strings be zeros?

$X ::= \{x \mid x \text{ is a message up to 1 Mbyte in size}\}$   
 $|X| = 2^{8,000,000}$

$Y ::= \{y \mid y \text{ is a 256bit string}\}$   
 $|Y| = 2^{256}$



- An ordinary hash value

2d711642b726b04401627ca9fbac32f5c8  
530fb1903cc4db02258717921a4881

- An extraordinary hash value with the first four digits are 0s

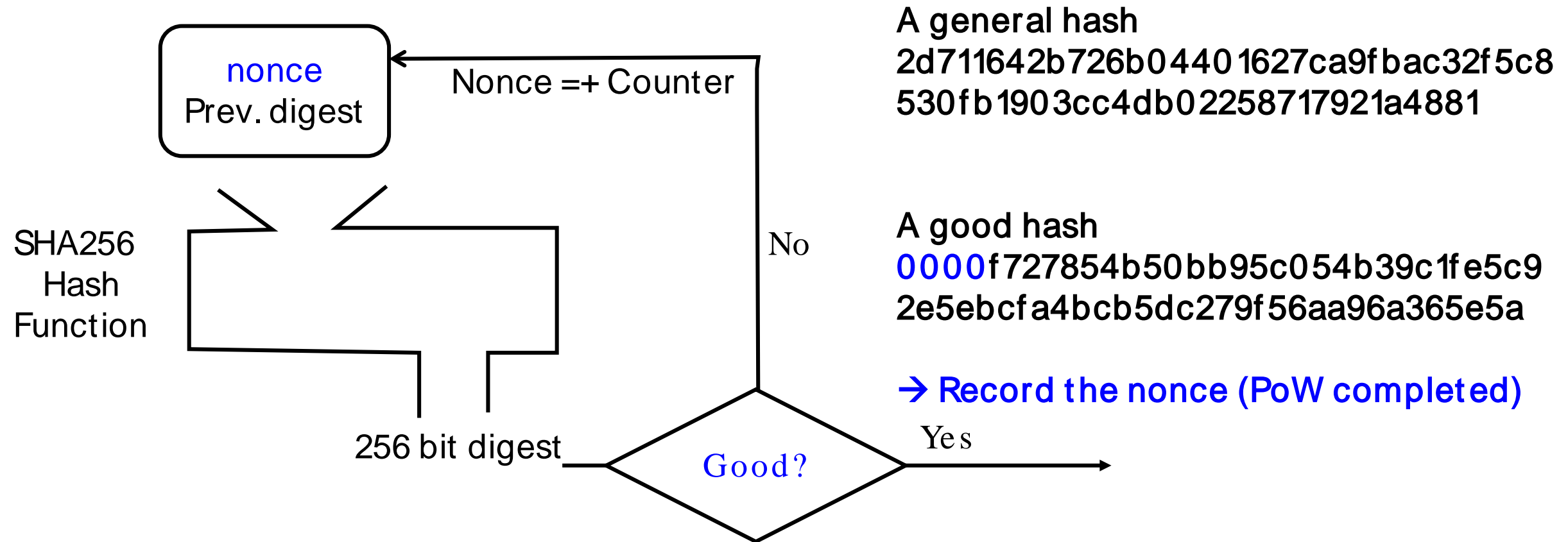
**0000**f727854b50bb95c054b39c1fe5c9  
2e5ebcfa4bcd5dc279f56aa96a365e5a

$c = \text{the set of any hash values}$   
 $= 2^{256}$

$a = \text{the set of wanted hash values}$   
 $= 2^{(256 - 16)} = 2^{240}$

$$P1 = a/c = 2^{-16}$$
$$= 1/2^{16} \sim 1/64000$$

# PoW and mining success



- Changeable difficulty: At least  $d = 16$  leading zero bits.
- Repeat the cycle until a good hash has been found.
- A node which has found a nonce that returns a good hash, records the nonce and completes the PoW.
- This node has the right to mint a given amount of coin as a reward for the work. (Mining coins as reward for work).



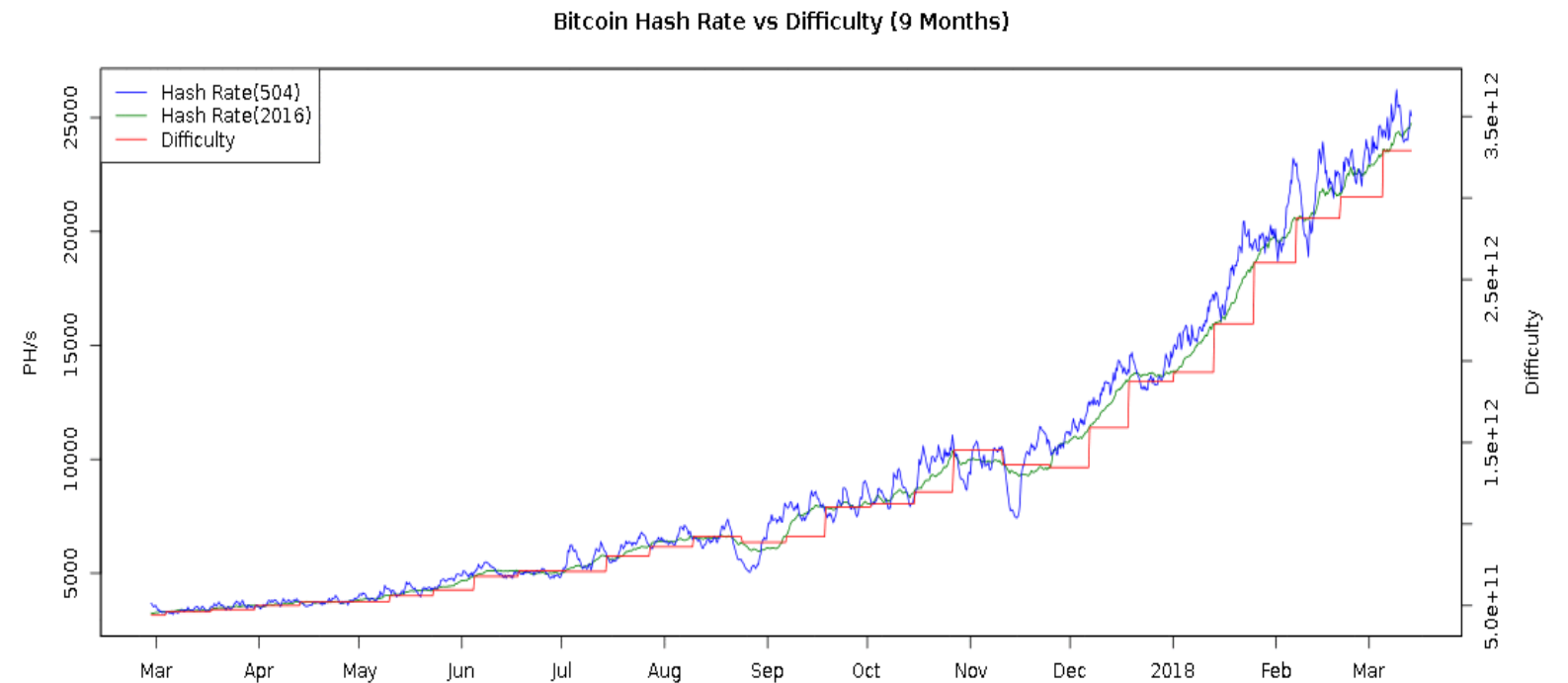
# Bitcoin Difficulty

- Simply put it is the number of leading zeros  $N_z$  to find a good hash.

Today,  $N_z = 74$

Difficulty =  $2^{74} = 18 \text{ e } 21 = 18 \text{ zeta hashes}$

- When there are more miners who joined the mining network, the time to mine a single block gets shortened.
- Then, the difficulty level is increased.
- Then,  $N_z$  is increased to have the average time to mine a single block fixed to 10 min.
- Today, it is about 25 exa hashes/sec.



Source : <https://bitcoinwisdom.com/bitcoin/difficulty>

# Trust enabled by blockchain

- **Immutable recording of transactions and openness** for reference **give confidence** to all the parties who are with free will rationally involved in transactions.
- **What's recorded in blockchain can be trusted** by everyone involved in transactions **for the integrity of its content** even after quite some time has passed since the moment it was generated.

# PoW is fundamental to data immutability of blockchain.

- How long does it take to mine a block alone?  
For one AntMiner machine, it would take 16 years.  
But, with more than 1 million miners working together, it will take 10 minutes on the average to mine a single block.
- The nonce is thus the proof that the network of 1 million miners have worked together to forge a block.
- The content is **scribed** into the blockchain which thus cannot be altered easily.

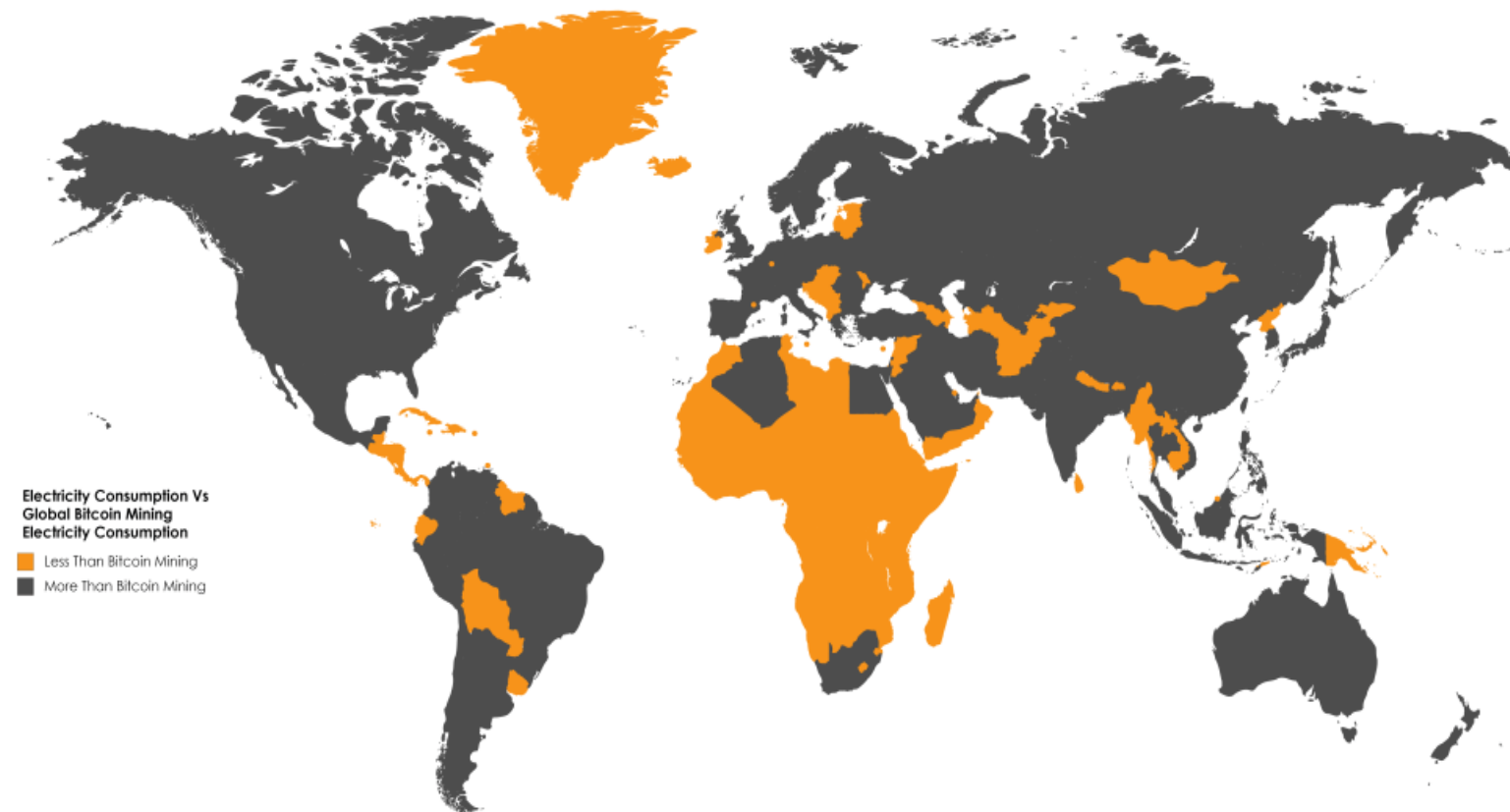
# But the current bitcoin network is **re-centralized**.

- Today, mining mogules investing heavily on state-of-the art ASIC miners appear.
- The bitcoin network is left with only handful of these mogules.
- This shows **the current bitcoin network is centralized**, leading to that the immuntability of the blockchain lies at the had of a few peope.

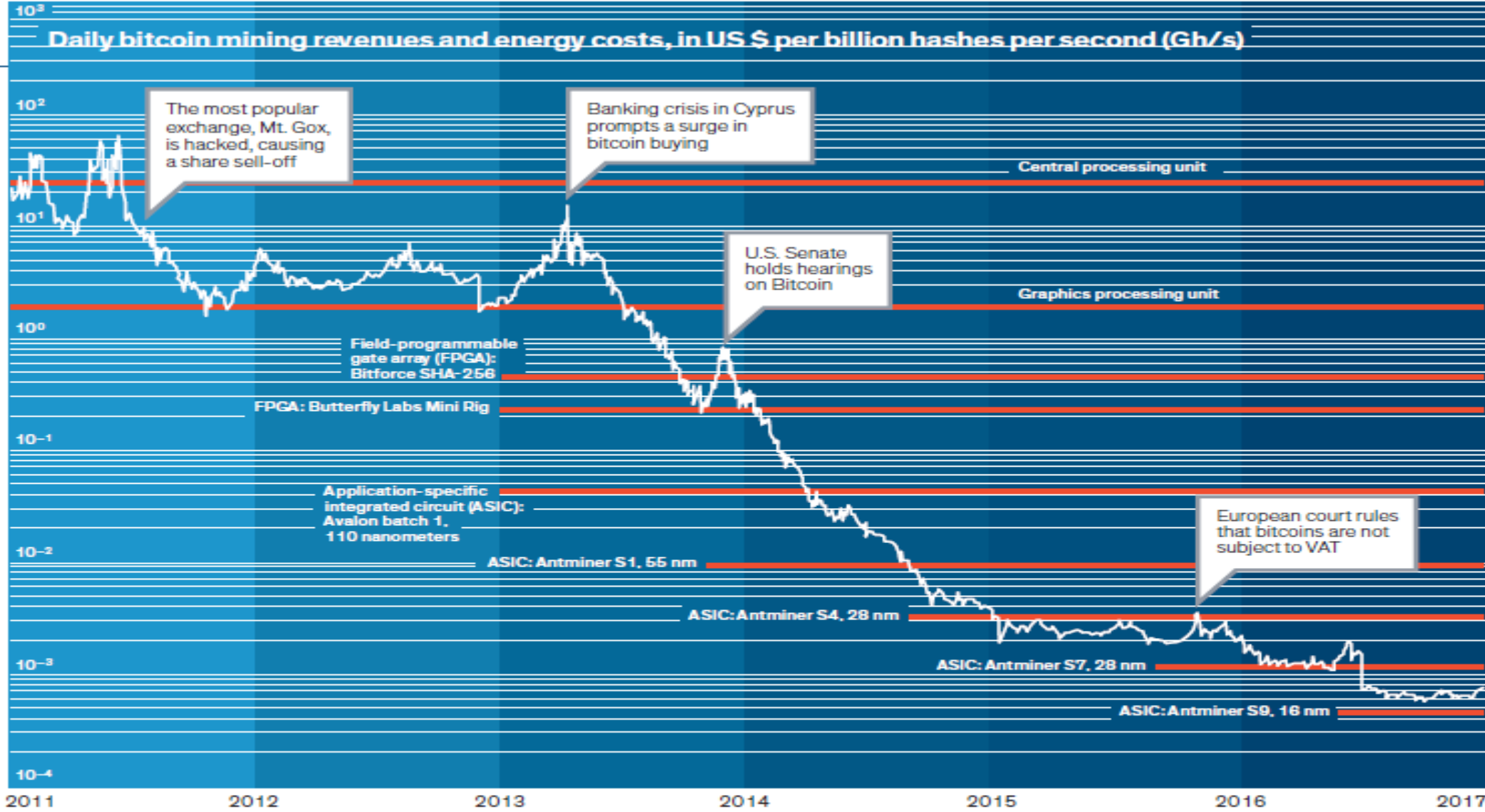


# Energy spending for bitcoin mining exceeds energy consumption of a country

- According to [Bitcoin](#) analysis blog [Digiconomist](#), energy consumed by Bitcoin mining now exceeds what is used by countries like Ireland, Hungary, Oman, and Lebanon. Bitcoin uses about as much power as the entire country of Morocco and slightly less than Bulgaria. If Bitcoin were a country, it would have the 61st highest energy consumption. However, this only covers miners. It does not include any power consumed by Bitcoin-enabled devices like vending machines and ATMs.



# Revenue from Mining vs. Energy Cost

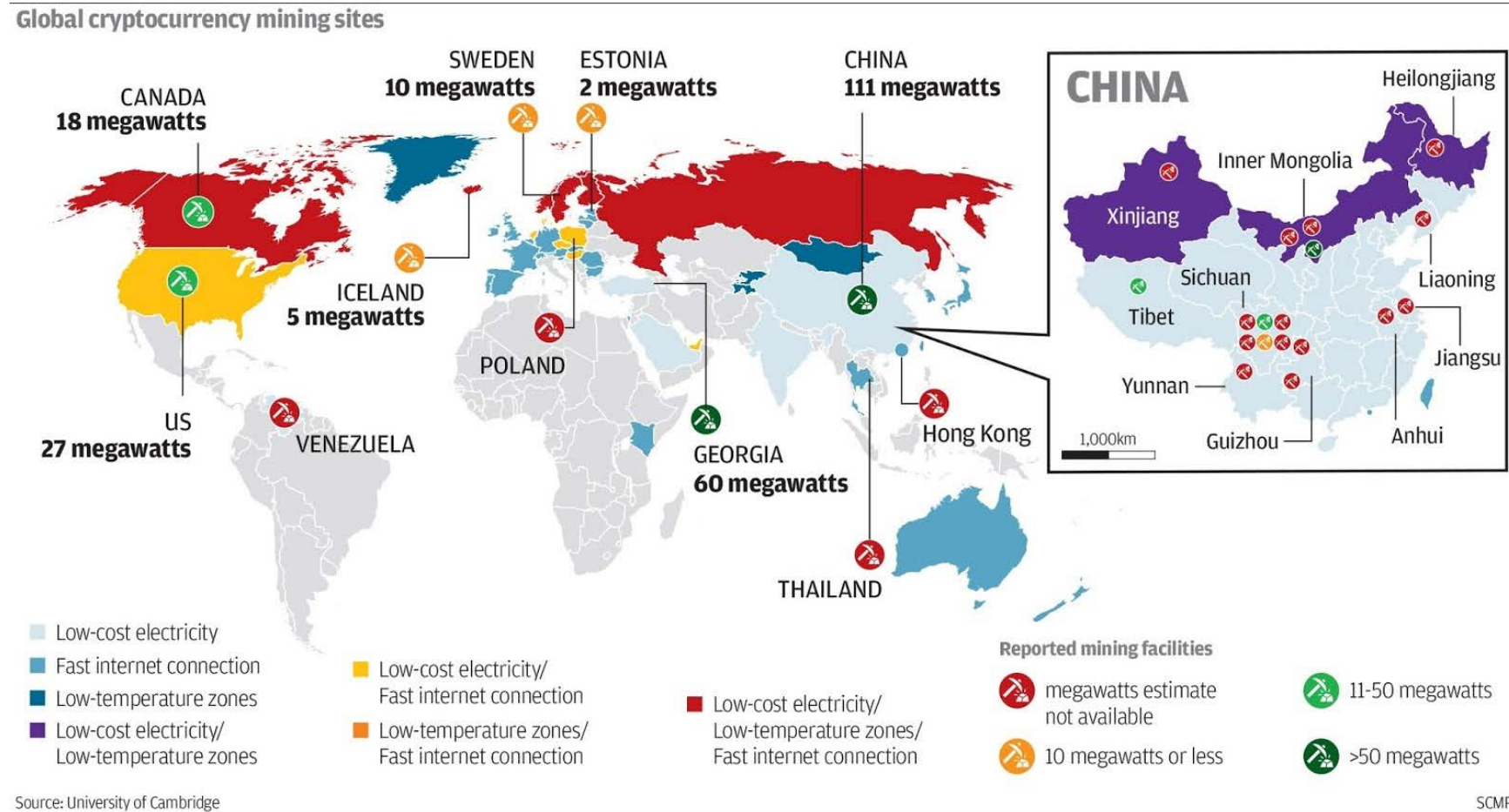


Hardware acceleration is the culprit of high energy cost and re-centralization.

**Sisyphian Slide:** Daily revenues for mining bitcoins [white], in US dollars per unit of computational power, are generally somewhat higher than the daily energy costs [red] of running the computers.










# Bitcoin, re-centralized?



- Provided any one of mining moguls decided to change his position to play attack, for whatever motivation the person might have, the person already has the power to do so.
- Blockchain stays no longer immutable and the public trust established to blockchain has eroded.

# Summary of alternatives to Proof-of-Work

	Pros	Cons	Coins within top 50 rank
<b>PoW</b> (Proof-of-Work)	<ul style="list-style-type: none"> <li>• Strong security</li> <li>- Difficult to produce</li> <li>- Easy to verify</li> </ul>	<ul style="list-style-type: none"> <li>• Extreme computing power</li> <li>• 51% attacks</li> <li>• Transaction speed / Transaction throughput</li> </ul>	 <b>Bitcoin</b>  <b>Ethereum</b>
<b>PoS</b> (Proof-of-Stake)	<ul style="list-style-type: none"> <li>• Energy &amp; hardware efficiency</li> <li>• Much more expensive 51% attacks</li> </ul>	<ul style="list-style-type: none"> <li>• Recentralization</li> <li>• The rich-get-richer</li> <li>• “Noting at stake” problem</li> </ul>	 <b>Qtum</b>  <b>Stratis</b>
<b>DPoS</b> (Delegated PoS)	<ul style="list-style-type: none"> <li>• Scalability and speed</li> <li>• Energy &amp; hardware efficiency</li> <li>• Encouraging good behavior by real-time voting</li> </ul>	<ul style="list-style-type: none"> <li>• Recentralization</li> <li>• DDoS attacks</li> </ul>	 <b>EOS</b>  <b>NEO</b> <small>smart economy</small>
<b>PoA</b> (Proof-of-Activity)	<ul style="list-style-type: none"> <li>• Much more expensive 51% attacks</li> <li>• Decentralization</li> <li>- Validators are randomly selected.</li> </ul>	<ul style="list-style-type: none"> <li>• Recentralization</li> <li>• Extreme computing power</li> <li>• The rich-get-richer</li> </ul>	 <b>decred</b>

# Current PoWs have no ASIC resistance.

- PoWs below are proposed to prevent the advent of the devices.
  1. Ethash algorithm which uses directed acyclic graphs (DAG).
  2. Both Xln and Scripts use multiple SHA functions.
- They were effective in the past, but failed to prevent ASIC devices. This failure is due to no enough variations on crypto puzzles.



# Directed Acyclic Graphs (Ethereum)

- The output of each hash function is mixed with a page fetched from DAG.
- No one knows *a priori* which page has to be fetched from DAG until the output of SHA is provided.
- Limited bandwidth is a bottleneck for developing ASIC devices.
- But, news has been released to advertise ASIC devices for this ethash hashing algorithm.

<https://www.coindesk.com/a-new-line-of-powerful-asic-miners-is-coming-to-ethereum>

<https://cointelegraph.com/news/bitmain-releases-ethash-asic-miners>



**etoro** THE SMARTEST WAY TO BUY AND SELL XRP (by Ripple Labs)  
 INSTANT | INTUITIVE | SECURED  
 Highly volatile investment product. Your capital is at risk. [Join eToro](#)

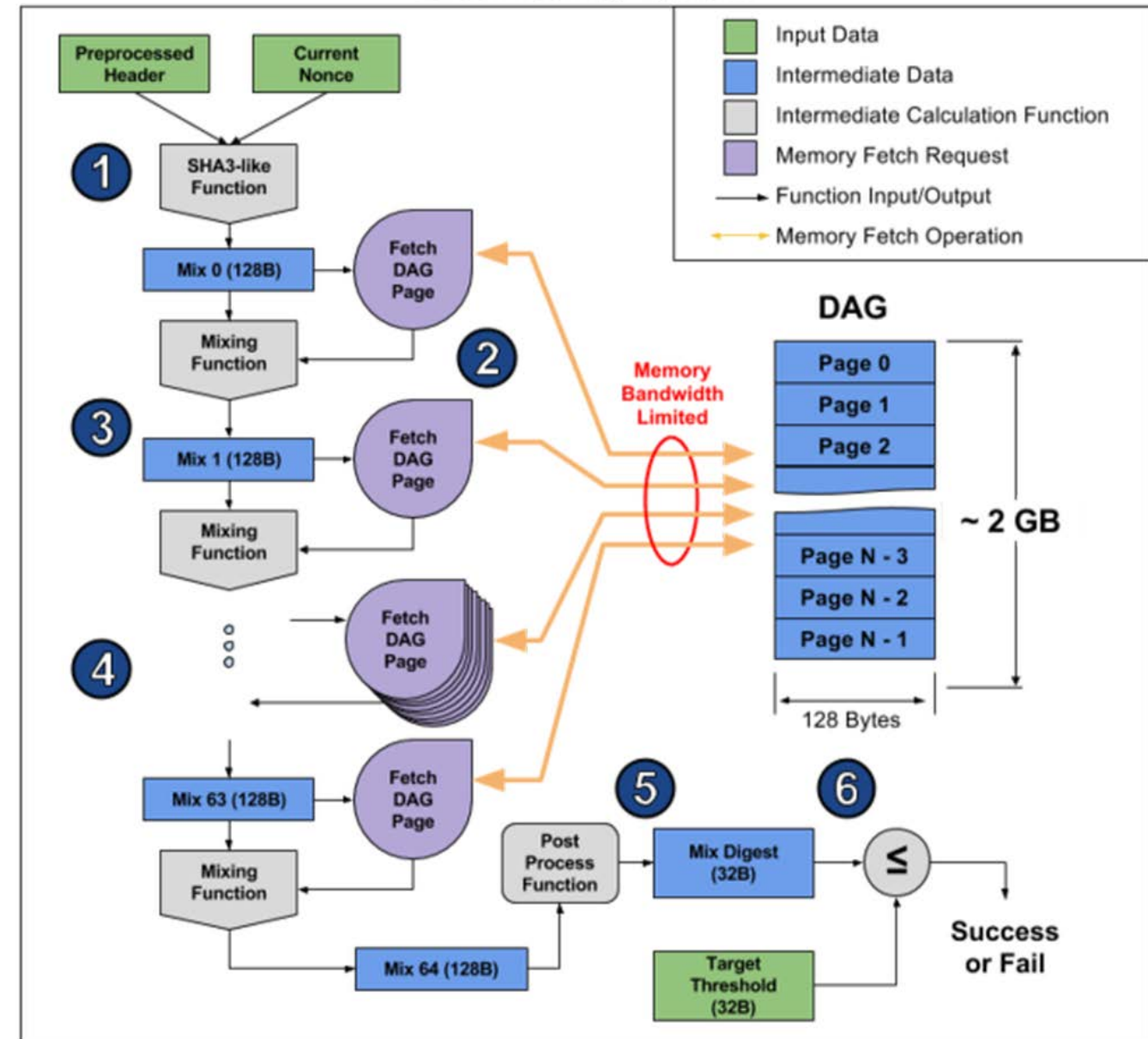
Nikhilesh De  
 Sep 13, 2018 at 16:05 UTC  
 Updated Sep 13, 2018 at 19:15 UTC

A chip designer with extensive experience in developing bitcoin mining devices is turning her sights on the ethereum protocol.

Chen Min, the former chief chip maker at bitcoin mining chip developer Canaan Creative, launched a new venture to build cryptocurrency mining devices called Linzhi. The firm's first project tackles the ethash algorithm used by ethereum and ethereum classic, with a new line of application-specific integrated circuits (ASICs) miners set to be released sometime next year.

Dubbed Project Lavasnow, Linzhi's new ethereum miner claims to use 1/8th the amount of

Ethash Hashing Algorithm



Source : <https://www.slideshare.net/dongsamb/ethash-ethereum-pow-algorithm>

# Multiple SHA functions (Dash)

- Xln uses a fixed sequence of *ln* different hash functions where the output of one becomes the input of the next. But, the order is fixed.
- No way for preventing the advent of ASICs.



The Antminer D3 from Bitmain Read more: [Best Dash Mining Hardware - Antminer D3 Review | 99Bitcoins](#)

X11	X12	X13	X14	X15	X17
blake	blake	blake	blake	blake	blake
bmw	bmw	bmw	bmw	bmw	bmw
groestl	groestl	groestl	groestl	groestl	groestl
jh	jh	jh	jh	jh	jh
keccak	keccak	keccak	keccak	keccak	keccak
skein	skein	skein	skein	skein	skein
luffa	luffa	luffa	luffa	luffa	luffa
cubehash	cubehash	cubehash	cubehash	cubehash	cubehash
shavite	shavite	shavite	shavite	shavite	shavite
simd	simd	simd	simd	simd	simd
echo	echo	echo	echo	echo	echo
	ocean?	hamsi	hamsi	hamsi	hamsi
		fugue	fugue	fugue	fugue
			shabal	shabal	shabal
				whirlpool	whirlpool
					losetlose
					djb2

Source: <https://getpimp.org/what-are-all-these-x11-x13-x15-algorithms-made-of/>



# Scripts (Lbry)

- Lbry sequentially uses a mix of SHA512, SHA256 and RIPEMD.
- ASIC devices for Lbry were already developed.

```
18  uint256 PowHash(const std::vector<unsigned char>& input)
19  {
20      CHash256 h256;
21      CSHA512 h512;
22      CRIPEMD160 h160;
23
24      std::vector<unsigned char> out;
25      out.resize(h512.OUTPUT_SIZE);
26
27      std::vector<unsigned char> out_small;
28      out_small.resize(h160.OUTPUT_SIZE);
29
30      h256.Write(input.data(), input.size());
31      h256.Finalize(&out[0]);
32      h256.Reset();
33
34      h512.Write(out.data(), h256.OUTPUT_SIZE);
35      h512.Finalize(&out[0]);
36
37
38      h160.Write(out.data(), h512.OUTPUT_SIZE / 2);
39      h160.Finalize(&out_small[0]);
40      h160.Reset();
41
42      h256.Write(out_small.data(), h160.OUTPUT_SIZE);
43
44      h160.Write(out.data() + h512.OUTPUT_SIZE / 2, h512.OUTPUT_SIZE / 2);
45      h160.Finalize(&out_small[0]);
46
47      out.resize(h256.OUTPUT_SIZE);
48      h256.Write(out_small.data(), h160.OUTPUT_SIZE);
49      h256.Finalize(&out[0]);
50
51      uint256 result(out);
52      return result;
53 }
```

Home / Uncategorized / baikal Miner Giant-B



## Baikal Miner Giant-B

★★★★☆ (3 customer reviews)

**IN STOCK 48 hours working time to send**

Algorithm Hash power  
blake256R14 160GH/s 410watt  
blake256R8 160GH/s 260watt  
blake2B 80GH/s 300watt  
Lbry 40GH/s 400watt  
Pascal 40GH/s 210watt

SKU: Giant-B Category: Uncategorized Tags: BAIKAL B, BAIKAL GIANT B, baikal miner, GIANT-B MINER

\$800.0

496 in stock

1

ADD TO CART

Source: <https://github.com/lbryio/lbrycrd/blob/master/src/hash.cpp>

# Needs new time-variant PoW

- ASICs become viable if the crypto puzzle is fixed.
- To make ASICs non-viable, our solution is to make many crypto puzzles and the issuance of puzzle **time-variant!**
- We call our solution **ECCPoW** because it is a result of **combination of Error Correction Coding (ECC) with PoW.**



# Item to consider a new PoW!

- A new puzzle generation system is capable of varying puzzles from block to block with the following properties:

P1: Easy to verify but difficult to prove

P2: Robust to detect block modification attacks

P3: Controllable in changing the difficulty level

P4: Open to anyone with a CPU

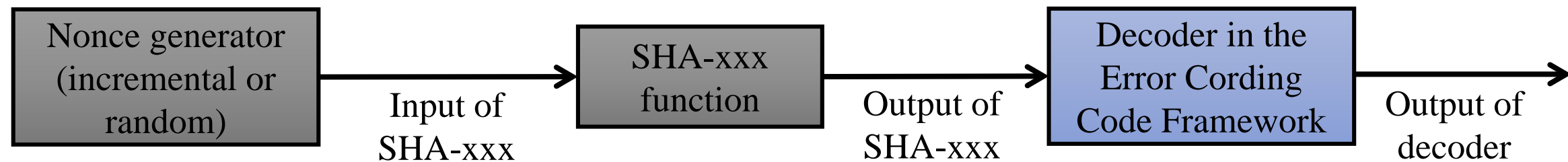
P5: Unfixed and changeable from block to block

- The re-centralized problem can be resolved thanks to P5.



# A proposed Error Correction Codes PoW (ECCPoW)

- There are many one-way functions in Inverse Problems such as [Error Correction Codes](#), Sparse-Signal Recovery, Space-Time Coding, Sphere-Decoding, Digital Communications Receiver algorithms.
- In these problems, encoding is easy but [decoding is time-consuming!](#)
- We combine a Error Correcting Code framework with SHA-xxx.



- The decision of mining success is made with the output of the above decoder.

# Impact of innovation

- Bitcoin system is most stable, secure, and immutable coin systems, which has run successfully so far during the past 10 years. (Similar argument holds true for Ethereum as well)
- We can create an unlimited number of Bitcoin (or Ethereum) systems using the proposition.
- Thousands of Bitcoin systems can resolve the scalability issues such as  
Transaction speed (TXs/ sec)
- This can be done without sacrificing the other advantages of Bitcoin system such as immutability and decentralization.
- The invention is the key part in any open public blockchain.
- [Any public blockchain can take the same set of benefits](#) equipped with the proposed ECCPoW.

# The proposed invention

- There are many inverse problems which can be used in cryptocurrency mining problems.
- In the context of Error Correction Coding, there is an encoder-decoder pair separated by a channel.
  - Encoding is easily made with relatively little computation.
  - Decoding is typically time-consuming with more computation.
- There are various ways to control the difficulty level.
- For example, the difficulty level of decoding can be varied by the size of the block code, the rate of the code, and the constraints on the input.

# Block code

- A block code  $C(N, \text{Rate}, \mathbf{G}, \mathbf{F}, \text{ENC}, \text{DEC}, \text{GF}(q))$  is well defined as a collection of codewords. When,  $q = 2$ , it is a binary system.
- $N$  is the dimension of the code (e.g.  $N = 512$ )
- $\text{Rate} = (N - M) / N$  is the rate of the code, where  $M < N$ .
- For example, with  $N = 1024$  and  $M = 256$ ,  $\text{Rate} = 3/4$ .
  
- $\mathbf{G}$  is the Generator matrix with dimension  $N \times (N - M)$ .
- $\mathbf{F}$  is the Check matrix with dimension,  $M \times N$ .
- $\mathbf{G}$  and  $\mathbf{F}$  are orthogonal to each other, i.e.,  $\mathbf{FG} = \mathbf{0}$ .
  
- A message vector  $\mathbf{m}$  is an  $(N - M) \times 1$  vector.
- A codeword  $\mathbf{c}$ , an  $N \times 1$  vector, is an element of the code and can be generated by multiplying a message vector  $\mathbf{m}$  to the Generator matrix  $\mathbf{G}$ , i.e.,  $\mathbf{c} = \mathbf{Gm}$ .
  
- Galois Field of size  $q$ ,  $\text{GF}(q)$ , is used for addition and multiplication operations and storage of numbers in the system.

# Block code, encoder and decoder

- ENC implies the encoder function, i.e., ENC takes the message vector  $\mathbf{m}$  as the input and produces a codeword vector corresponding to it, e.g.  $\mathbf{c} = \text{ENC}(\mathbf{G}, \mathbf{m})$ .
- DEC implies the decoding function; DEC takes an arbitrary vector  $\mathbf{e}$  and returns a closest codeword  $\hat{\mathbf{c}}$ , i.e.,  $\hat{\mathbf{c}} = \text{DEC}(\mathbf{F}, \mathbf{e})$ .

$$\mathbf{s} = \mathbf{F} \mathbf{e}$$

$\mathbf{s} \in GF(q)^{M \times 1}$   
 $\mathbf{F} \in GF(q)^{M \times N}$   
 $\mathbf{e} \in GF(q)^{N \times 1}$

$$M < N$$

Encoder : Given  $\mathbf{e}$ , find  $\mathbf{s} = \text{Enc}(\mathbf{e}, \mathbf{G})$

Decoder : Given  $\mathbf{s}$ , find  $\hat{\mathbf{c}} = \text{Dec}(\mathbf{s}, \mathbf{F})$

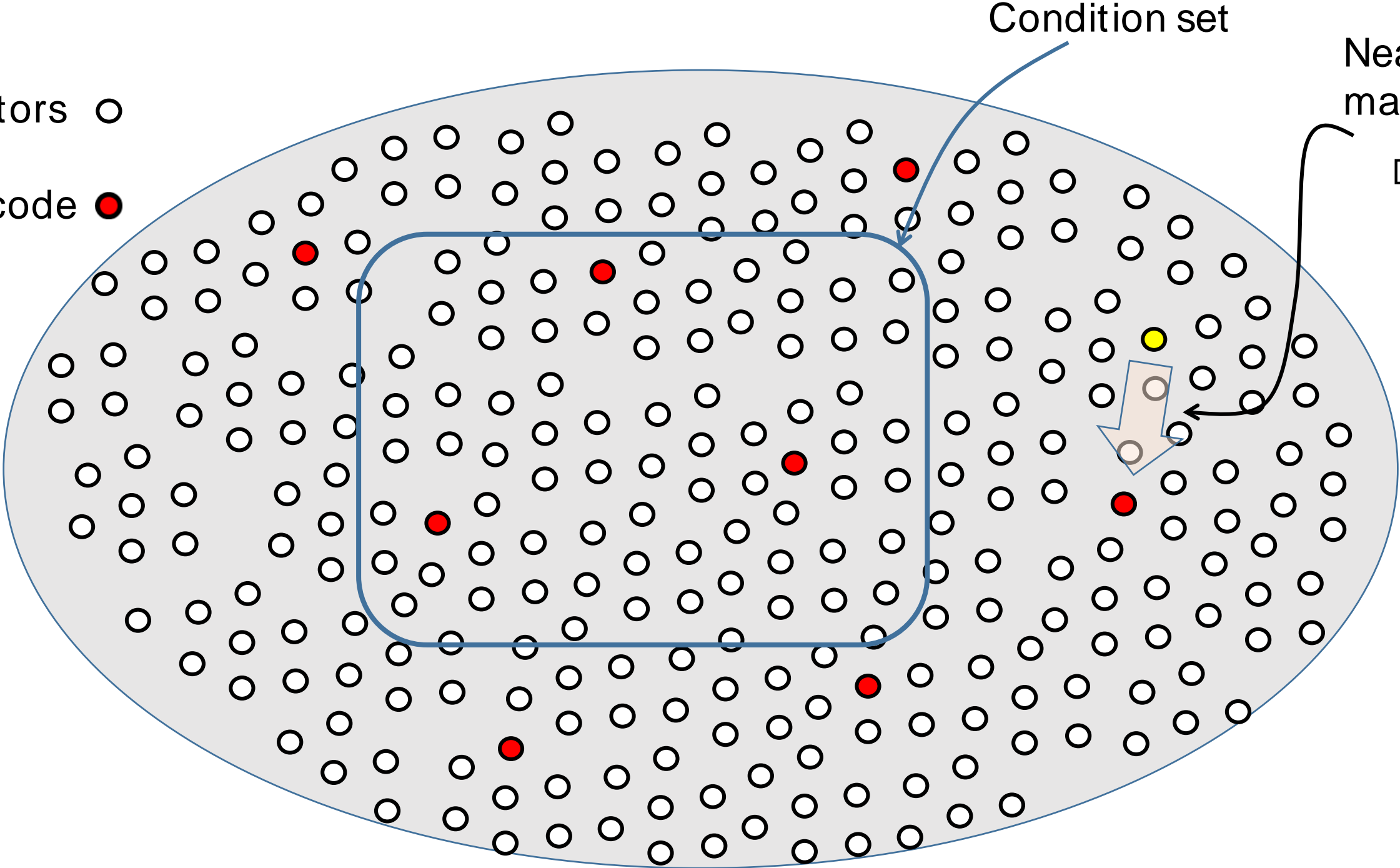
# Decoder

- DEC is to find a codeword  $\hat{\mathbf{c}}$  most close to the input word  $\mathbf{e}$ .
- For the concept of distance, the Hamming distance can be used.  
For example,  $DH(\mathbf{e}, \hat{\mathbf{c}}) = \|\mathbf{e} - \hat{\mathbf{c}}\|_0$  is the number of non-zero values in the  $(\mathbf{e} - \hat{\mathbf{c}})$  vector.
- There are many ways to find  $\hat{\mathbf{c}}$  satisfying  $\mathbf{F}\hat{\mathbf{c}} = \mathbf{0}$ .
- We propose to use [the message passing graph decoder](#) for its excellency in accuracy and superiority in decoding speed.  
This is **to prevent a cheating attack** in which a smart miner comes up with a new decoder algorithm of his own developed and outpaces the regular miners using the designated decoder. If this is allowed, a hidden advantage goes to the smart miner.

# Geometrical Explanations

$2^{256}$  vectors ○

Rate  $\frac{1}{4}$  code ●  
 $= 2^{64}$



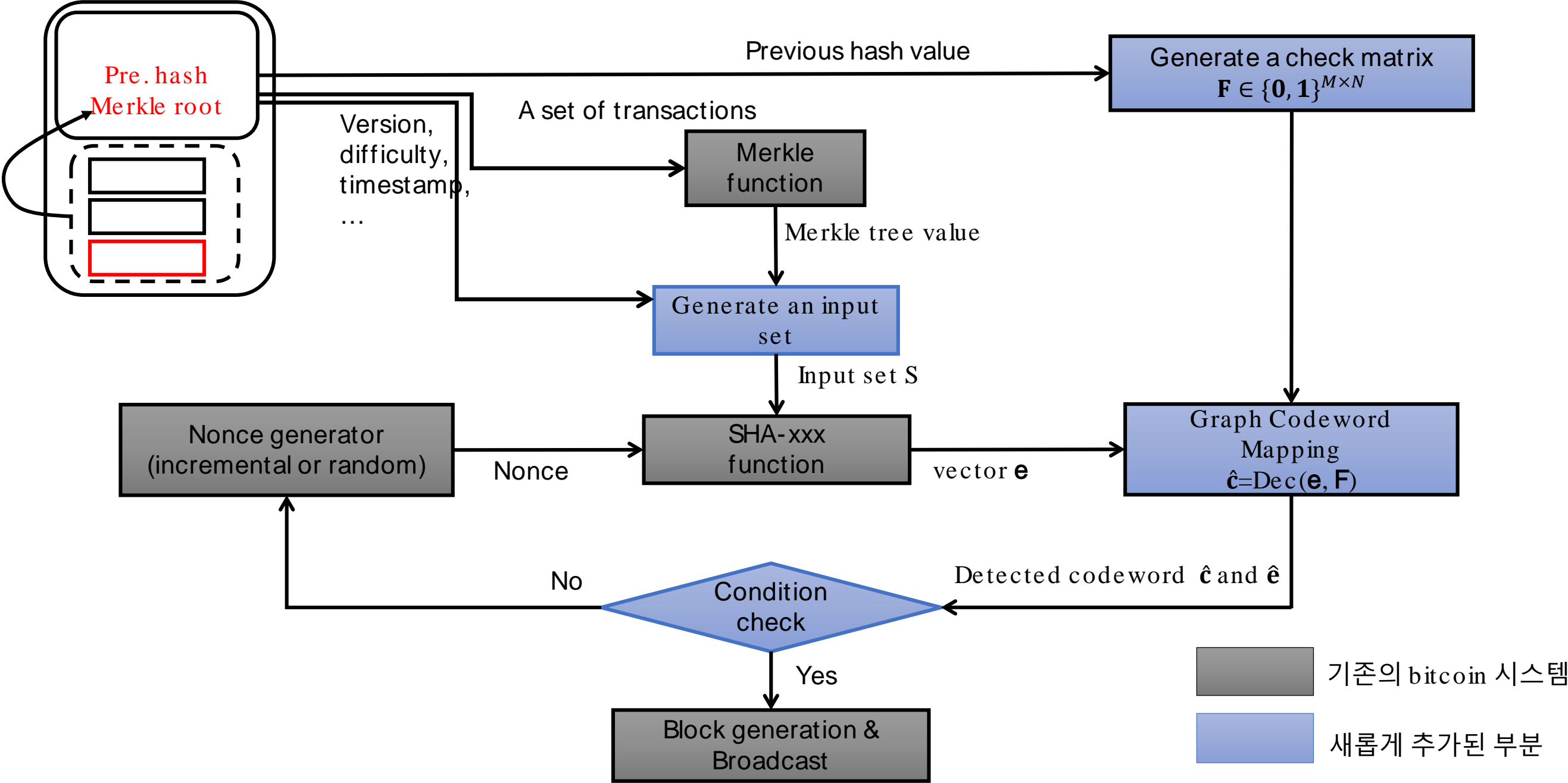
Condition set

Nearest codeword mapping of  $\mathbf{e}$  to  $\hat{\mathbf{c}}$ .

$$\text{DEC}(\text{yellow circle}) = \text{red circle}$$

$\hat{\mathbf{c}} = \mathbf{e} - \mathbf{c}$   
is the sparse error pattern.

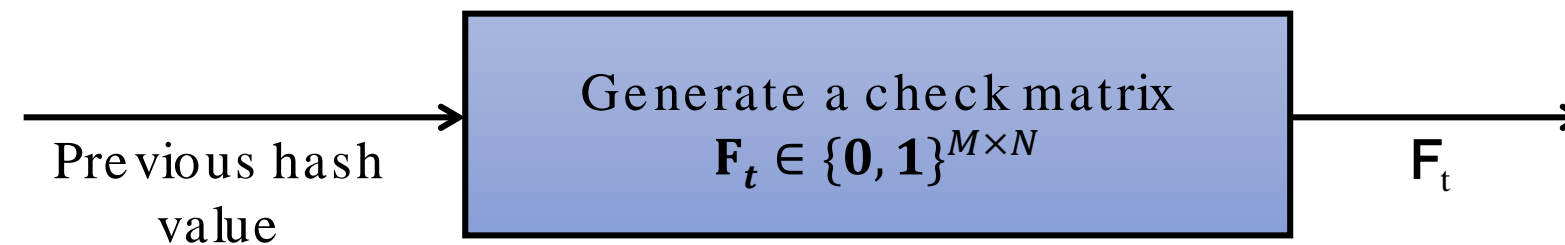
# Diagram of ECCPoW





# Generate a Check Matrix

- Parameter set  $\mathcal{S}_t = \{h_{t-1}, \text{code parameters}\}$ ;
- $\text{GenCheckMatrix}(\mathcal{S}_t) = \mathbf{F}_t$
- Generate a check matrix  $\mathbf{F}_t$  w.r.t. previous hash  $h_{t-1}$ .
- Takes the previous hash  $h_{t-1}$  as the input to this routine.
- That is,  $\mathbf{F}_t$  changes from block to block.



# Pseudo Code of the Decoder

- Input:
- ✓ Hard decision of a priori LLR:  $L_a^t = \mathbf{e}[t]$
- Iteration: repeat until converge
- Update variable-to-check node messages for  $t = 1, 2, \dots, N$  and  $\forall l \in Q1(t)$ :

$$L^{t \rightarrow l} = \left[ \sum_{l' \in Q1(t) \setminus l} (L_a^t \oplus L^{l' \rightarrow t}) / (j-1) \right]$$

- Update check-to-variable node messages for  $l = 1, 2, \dots, M$  and  $\forall t \in Q2(l)$ :

$$L^{l \rightarrow t} = \oplus \sum_{t' \in Q2(l) \setminus t} L^{t' \rightarrow l}$$

- Output
- ✓ Hard decision of a posteriori LLR:

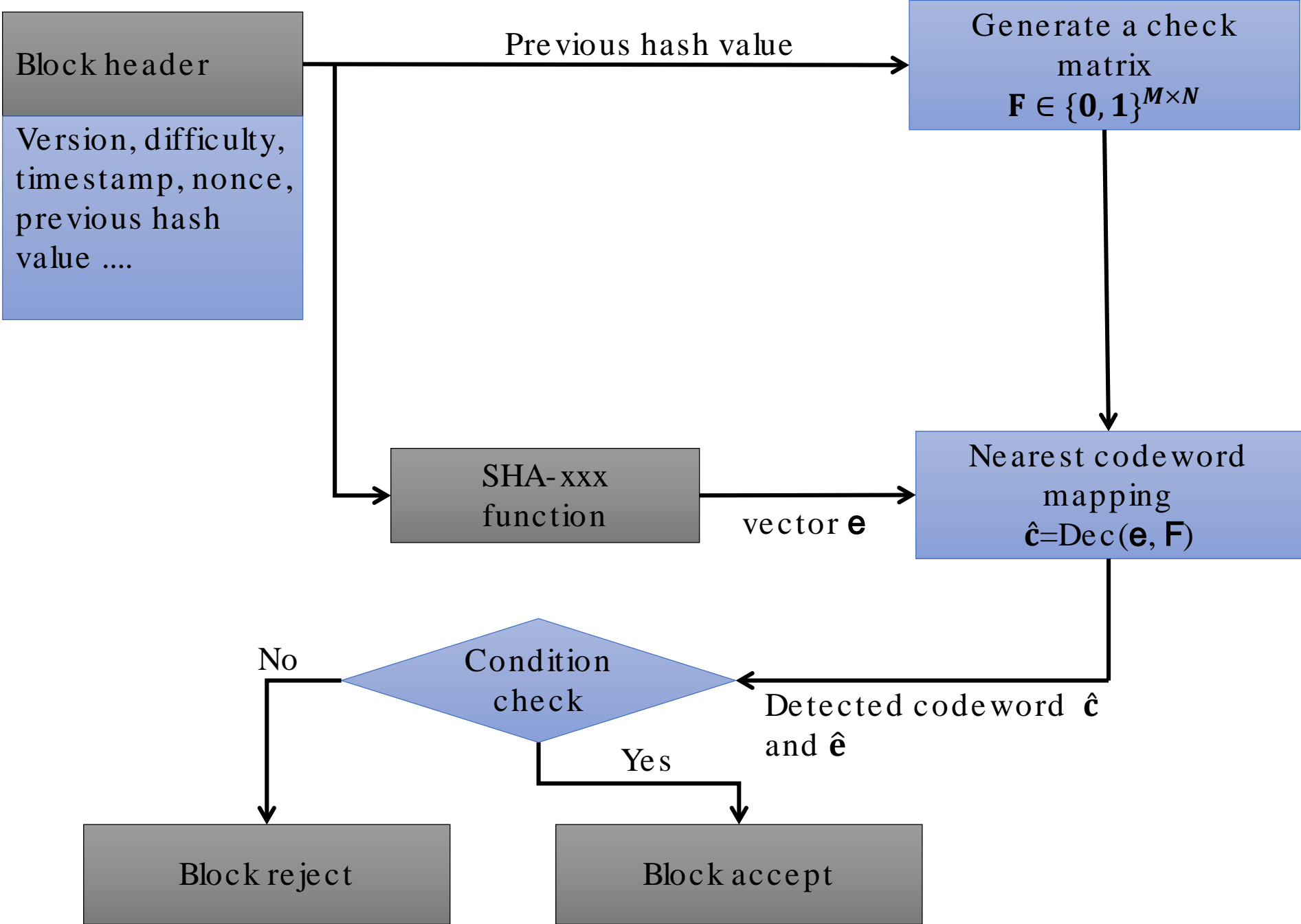
$$L^{t \rightarrow l} = L_a^t \oplus \left[ \sum_{l' \in Q1(t)} L^{l' \rightarrow t} / j \right] \square \hat{\mathbf{c}}[t]$$

# Implemented code in C

```
CMminer.c
]기타 파일 (전역 범위) NCMminer(double * BPSK_AWGN_Codeword)
91 }
92 //Bit to Check Node Messages --> LRqt1
93 for(i = 0; i < ITERATIONS; i++) {
94     for(k = 0; k < COLUMNS; k++) {
95         for(l = 0; l < COLUMN_WEIGHT; l++) {
96             temp3 = LRft[k];
97             for (m = 0; m < COLUMN_WEIGHT; m++) {
98                 if (m != l) {
99                     temp3 += LRrt1[k][Row_In_Column[m][k]]; //LRqt1[k][Row_In_Column[l][k]] = infinity_test(LRqt1[k][Row_In_Column[l][k]]);
100                 }
101             }
102             LRqt1[k][Row_In_Column[l][k]] = (short) temp3;
103         }
104     }
105
106     fprintf(out, "\n\n\nLRqt1 iteration %i\n", i);
107     for(k=0; k < COLUMNS; k++) {
108         fprintf(out, "\n");
109         for(m=0; m < ROWS; m++)
110             fprintf(out, "%i ", LRqt1[k][m]);
111     }
112
113     //Check to Bit Node Messages --> LRrt1
114     for(k = 0; k < ROWS; k++){
115         for(l = 0; l < ROW_WEIGHT; l++){
116             temp3 = 0.0;
117             sign=1;
118             for( m =0; m < ROW_WEIGHT; m++){
119                 if( m != l){
120                     temp3 = temp3 + func_f( fabs( LRqt1[Column_In_Row[m][k]][k] ) );
121                     if(LRqt1[Column_In_Row[m][k]][k] > 0.0)
122                         temp_sign = 1;
123                     else
124                         temp_sign = -1;
125                     sign=sign+temp_sign;
126                 }
127             }
128             magnitude = func_f(temp3);
129             LRrt1[ Column_In_Row[l][k] ][k] = (short) sign*magnitude;
130         }
131     }
132     fprintf(out, "\n\n\nLRrt1 iteration %i\n", i);
133     for(k=0; k < COLUMNS; k++){
134         fprintf(out, "\n");
135         for(m=0; m < ROWS; m++)
136             fprintf(out, "%i ", LRrt1[k][m]);
137     }
138     //Last iteration get LR (pi)
139     for(m = 0; m < COLUMNS; m++){
140         LRpt[m] = LRft[m];

```

# Diagram of New Verifiers



# New Functions in ECCPoW

- New functions

1. `int **H = GenCheckMatrix(int n, int wc, int wr, int seed);`
2. `bool DEC(int **H, int *e, int n, int wc, int wr, int *c);`
3. `void Dec_Difficulty(int &n, int &wc, int &wr, int level);`

- These functions are the key parts of the proposed solution.

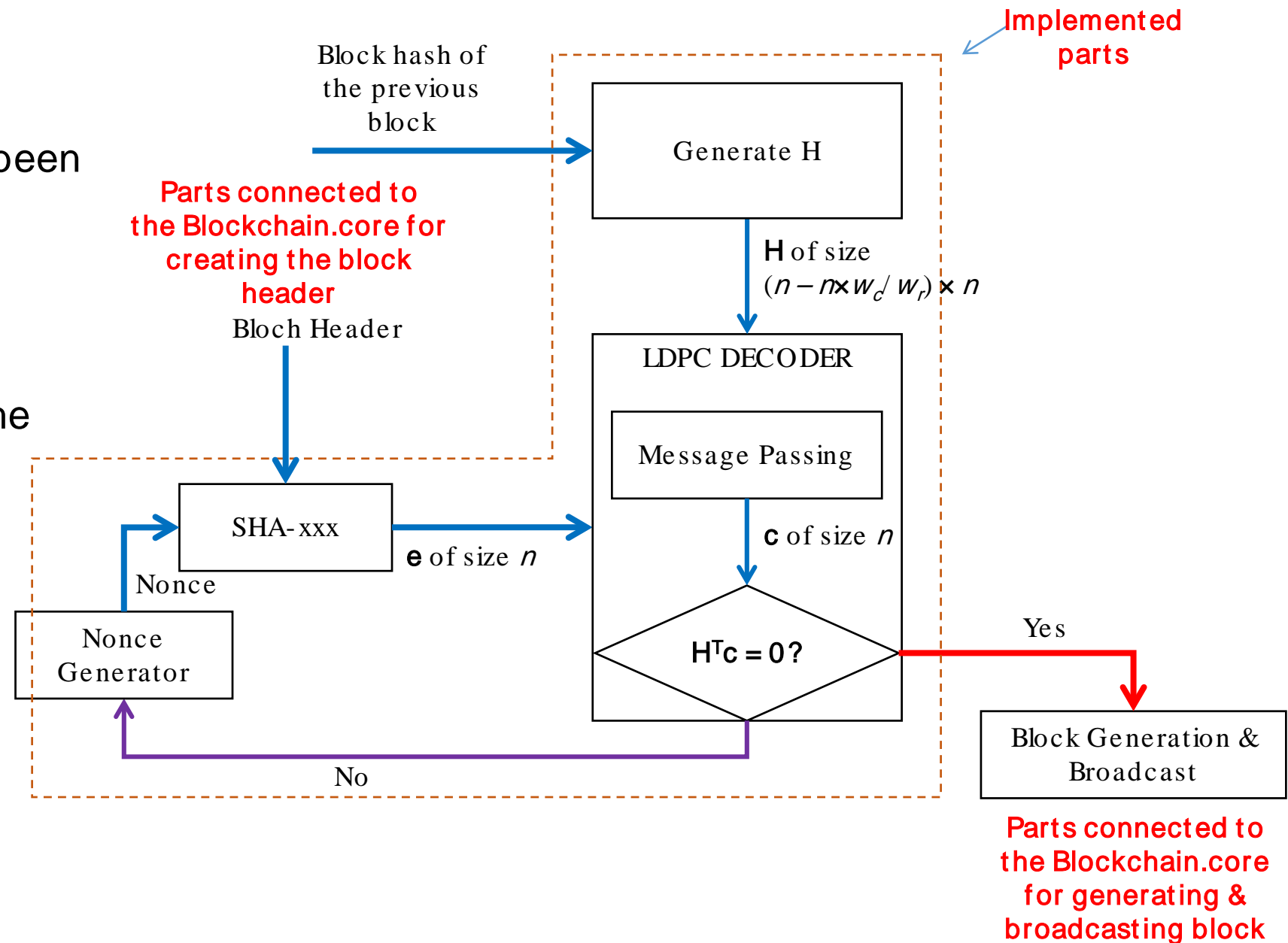
1. They are implemented in C++.
2. They are used to implement a mining routine

- An example of mining

1. generate block header with zero nonce.
2. `Dec_Difficulty(&n,&wc,&wr,difficulty)`
3. `Seed = f(phv)`
4. `H = GenCheckMatrix(n, wc, wr, seed)`
5. `nonce = nonce + 1`
6. `e = SHA256(version, time, difficulty, nonce, mtv)`
7. `flag = DEC(H,e,n,wc,wr)`
8. If `flag == 0`; go to step 4
9. Update `chv` and nonce.
10. Generate block and broadcast.

# ECCPoW Hardfork

- New ECCPoW  
A new structure of the block header has been introduced and, three new functions are also have been introduced.
- We aim to link these functions to existing the blockchain. For example, mining function, chain validation function, consensus function and so on.



# Impact of ECCPoW 1: It is easier to start a new blockchain network.

- A large blockchain network is stable and not easy to disrupt.
- Today there are mining equipment renting sites.
- A new borne blockchain network needs to grow, but newbies are much more vulnerable to 51% attacks.
- New blockchain networks with ECCPoW do not suffer from such problems since there are no mining equipment available for ECCPoW.

# Impact of ECCPoW 2:

## One can make multiple blockchain networks

- It is easy to make a new blockchain with ECCPoW.
- Suppose hardforking a Bitcoin and an Ethereum with ECCPoW.
- Let us call them BitECC and EhterECC protocols.
- Make the first blockchain network by running EtherECC over a network (Pusan coin)
- Make the second blockchain network by running BitECC over other network (Gwangju coin)
- Make the third blockchain network by running EtherECC over another network (Seoul coin)
- Make the fourth blockchain network by running BitECC over yet another network (Korea coin)
- Each cryptocurrency is independent with its own genesis block and random starting seed and can be adjusted sufficiently strong for its regional requirement in the sense of scalability, security and decentralization.
- These blockchains are inter-connected at the local, regional, and national, transnational level.



# Impact of ECCPoW 3: Resolving the Scalability Trilemma

- Trilemma by V. Buterin is well known: Only up to **two out of the three** virtues such as Scalability, Decentralization and Security **can be achieved simultaneously**.
- With ECC, each blockchain is already very strong in decentralization.
- Each ECC blockchain is flexible enough to provide various settings of transaction speeds and security levels.
  - ◆ Campus ECC blockchain networks can be set to work very fast allowing up to 100s of thousands of TXs per second since the delay of the underlying communications network is very small.
  - ◆ Regional ECC blockchain networks can be set to work fast, i.e. allowing up to 10s of thousands of TXs per sec.
  - ◆ National ECC blockchain networks can be set sufficiently fast for covering inter-regional transactions.
  - ◆ Transnational ECC blockchain networks shall be set to work slow due to large delays.
- All these blockchains started up with its own seed and decentralized levels are mutually independent and each one can be set to work at the required level of security and speed to serve its purpose.
- All these ECC blockchains can be inter-connected via *distributed value-exchange* networks.
- The connected ECC blockchains can be named the ECC Blockchain International.
- *ECC Blockchain International* as a whole can resolve the Scalability Trilemma.

# Impact of ECCPoW 4:

## It is safe to use a time-proven blockchain protocol.

- Bitcoin protocol has withstood the tough test of time.
- Thus, the networking part and the wallet part are robust enough.
  
- PoW is problem. Yes.
- But it is not the problem of PoW.
- It is the **fixedness** of the PoW puzzle.
  
- ECCPoW puzzles can be made to vary over time.
  
- The problematic consensus part with a fixed PoW can be replaced with the new ECC PoW consensus.

## Impact of ECCPoW 5:

The complexity of ECCPoW puzzles can be set to grow very large; thus the cost for hardware acceleration is boundless.

- ECCPoW is a computer algorithm!
- Thus it is not impossible to find a hardware acceleration solution for it.
- ECCPoW puzzle can be represented as a randomly connected bipartite graph.
- In order to parallelize the algorithm, more memory and computation resource need to be allocated.
- The size of ECCPoW puzzle can grow very large.
- As the size of the puzzle grows, the more needed is the memory and computation resource.
- With ECCPoW puzzles, therefore, one can easily deter the emergence of hardware acceleration solution.
- Deterrence to hardware acceleration offers a blockchain network with small power consumption requirement.

# Development Schedule

- Open research platform
  - Source codes github uploaded
  - Open development
- 2019 plan
  - ECCPoW 0.5 Version
  - Ethereum and Bitcoin Hardforks with ECCPoW 0.5v
  - Develop them into Ethereum ECCPoW 1.0v and Bitcoin ECCPow 1.0v
- 2020 plan
  - Network growth at least by 10,000 nodes worldwide
  - Co-working with Bitcoin and Ethereum communities

# Concluding Remarks

- PoW is fundamental for blockchains immutability.
- You put PoW to a block, you get the benefit of data immutability.
- Recentralization issue is problem due to fixeness of PoW puzzles, not due to PoW itself.
- Trilemma by V. Buterin is well known. Only up to two out of the three virtues such as Security, Decentralization and Security can be achieved simultaneously.
- Flexible puzzles enabled by ECCPoW can resolve the recentralization problem; PoW has shown to be the most secure.
- Multiple layers of ECCPoW blockchains can operate simultaneously resolving the issues of scalability and thus breaking the trilemma.
- ECCPoW blockchains can play a crucial role in ushering in the ideals of blockchains and advance our society to the next level!

- Thank you!

- Q&A

- We are looking for people to join us.