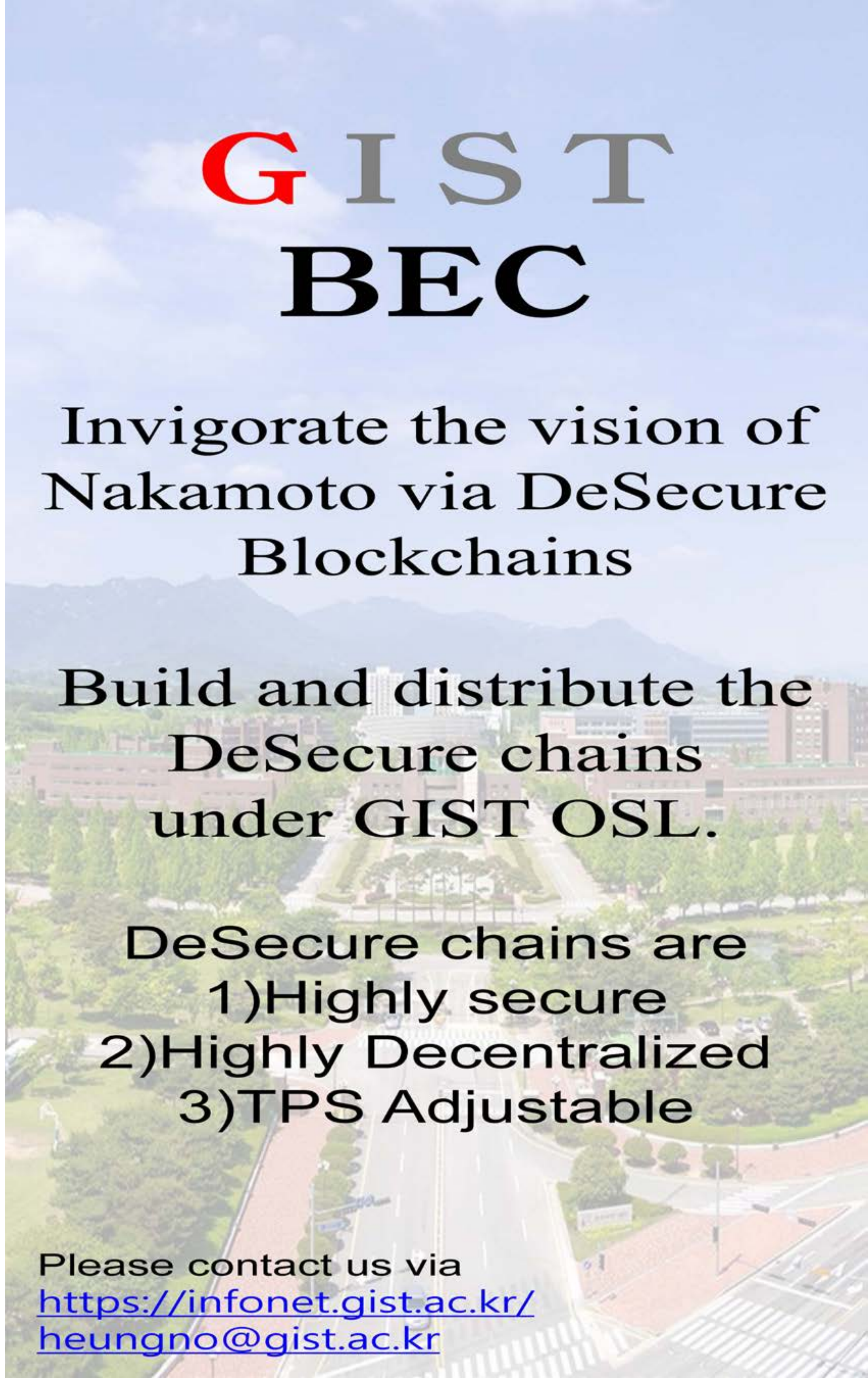
Scalable DeSecure ECCPoW Blockchains

Heung-No Lee, GIST, South Korea
Home page: http://infonet.gist.ac.kr
Facebook/Publication ID: Heung-No Lee
E-mail: heungno@gist.ac.kr

Abstract

GIST Blockchain-Economy Center (BEC, Director Heung-No Lee) aims to introduce the Decentralized Secure(DeSecure) blockchains it has been developing since 2018. They aim to resolve the re-centralization problem of today's mining market. One of the key ideas is to have the proof-of-work (PoW) puzzle time-varying from block-to-block, using the error-correction-codes (ECC). Two new blockchains based on Bitcoin and Ethereum are to be developed using new consensus algorithm based on this new ECC-PoW. Time-varying puzzles make it very difficult to develop an ASIC mining chips. As the result, with the size of network growing, the difficulty level needs not be growing as well. As such, energy spent for mining can be controlled. The proposed ECC-PoW mechanism is to be explained in details. In addition, our plan to hardfork Bitcoin and Ethereum, by replacing the SHA based PoW with the proposed ECC-PoW, and by developing two new DeSecure blockchains, i.e. BTC-ECC and ETH-ECC, is discussed. The two DeSecure blockchains will be openly shared under an open source license at Github. We address how DeSecure blockchains can be used to resolving the issue of scalability. Our schedule to release the cores (C++ and Go) and technical meet-ups will be addressed.



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 270 journal and conference publications. In the past, he worked at HRL Laboratory, Malibu, California, U.S.A., as Research Staff Member and as Assistant Professor at the University of Pittsburgh, Pittsburgh, Pennsylvania, U.S.A. He is currently a full tenured professor at Gwangju Institute of Science and Technology (GIST), Republic of Korea.

His research lies in the areas of Information Theory, Signal Processing 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.

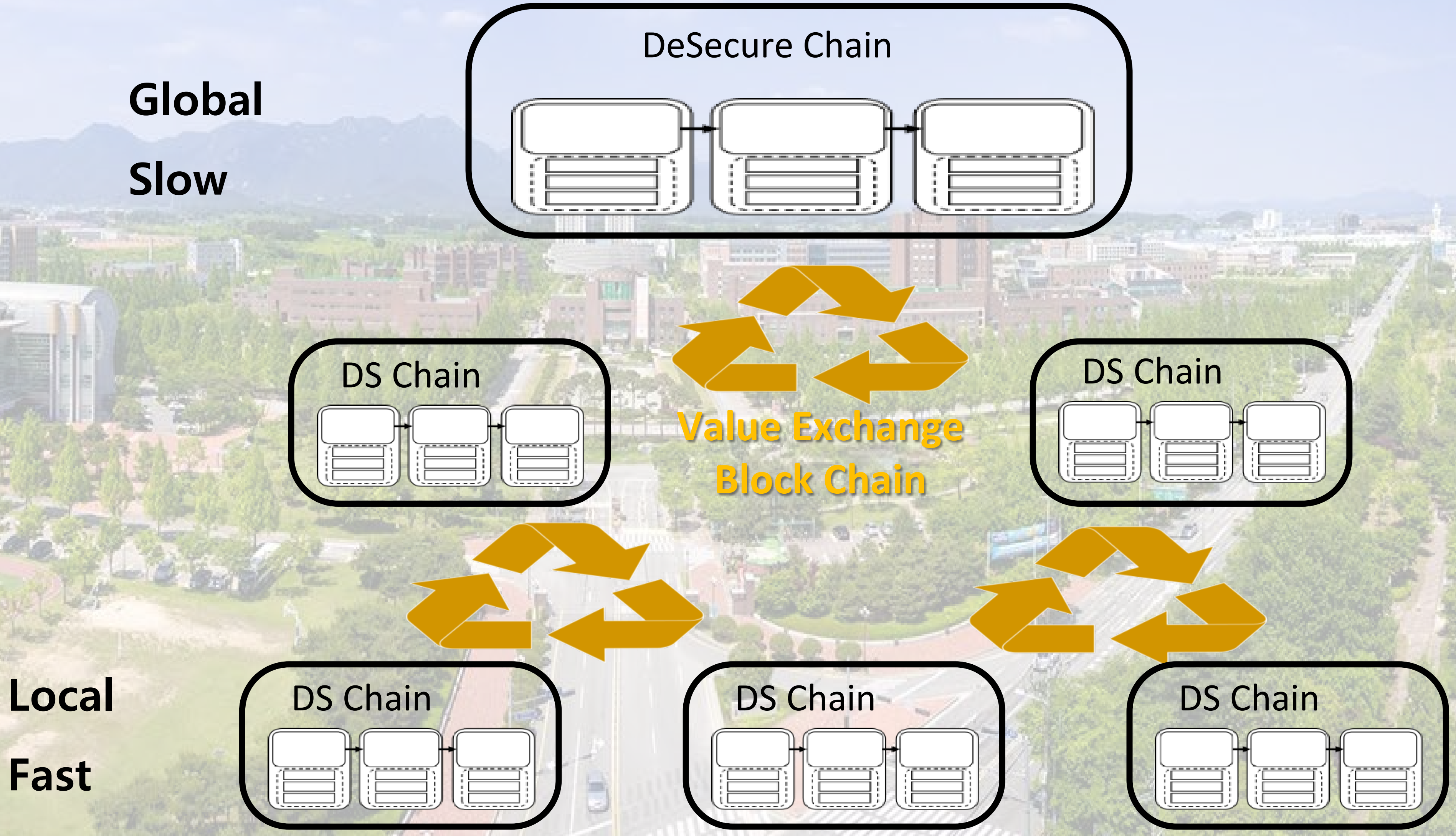
He was the Director of Electrical Engineering and Computer Science within GIST College in 2014. Administrative positions he has held at GIST include the Dean of Research and the Director of GIST Research Institute.

Talk today

- DeSecure Blockchains
- Error Correction Codes based PoW
- Safe Transactions enabled by novel DS analysis
- Release Plan

Gwangju Institute of Science and Technology
Blockchain Economy Center

DeSecure chains are
1) **Highly secure**
2) **Highly Decentralized**
3) **TPS Adjustable**



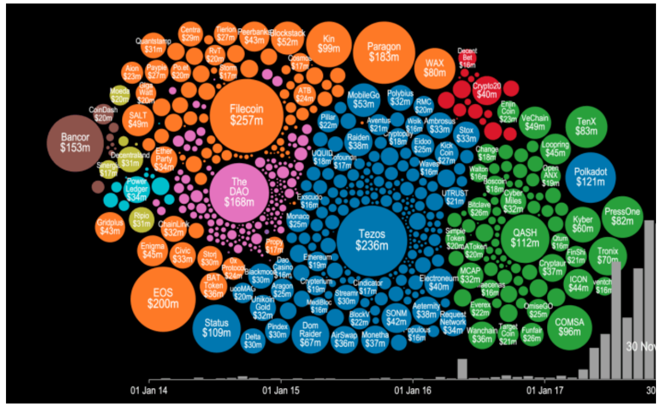
The aim is
to build and distribute the DeSecure chains under GIST OSL.

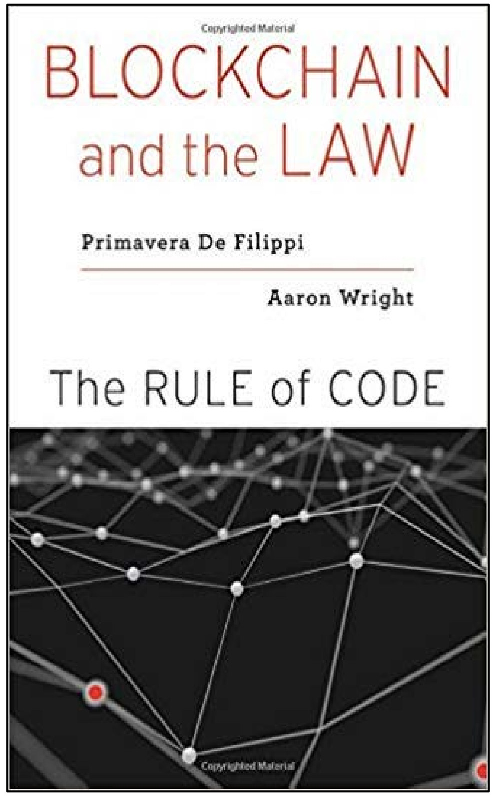
Please contact us via
https://infonet.gist.ac.kr/
heungno@gist.ac.kr

Bitcoin’s Ideals

- Since birth in 2009, Bitcoin has never stopped breathing and alive currency system.
- It is a global digital currency which 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
  Reduction of inequality

Ethereum's Ideals





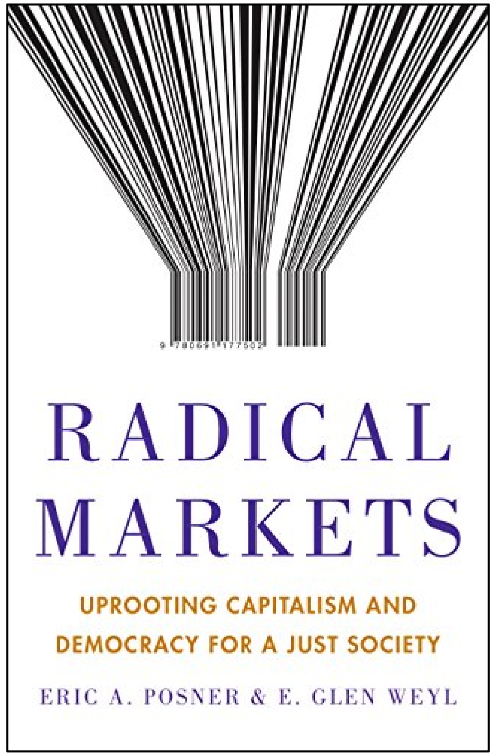
- Ethereum network allows not only coin TXs, but also doc files and computer codes.

- A *decentralized* app (Dapp) runs a front end code; a backend code runs in *the Eth Net*.
  - ✓ cf) For an ordinary app, the backend code is running on *a centralized server*.

- Smart contracts
  - ✓ A computer code can be executed and advanced to the next stage each time a contractual term matures.

- Decentralized autonomous organization has its bylaw written in smart contracts.
  - ✓ The organization spends tokens and makes governance decisions w.r.t. smart contracts.

- *Lex Cryptographia!*
- *Uprooting capitalism and democracy for a just society!*
- *Sharing Economy!*



Novel DeSecure Blockchains

- BTC and ETH are great BUT they are
  - Re-Centralized
  - Scalability Issue
  - Said to be too slow and small

- We aim to approach these two issues with DeSecure blockchains
  - Anti-ASIC ECC PoW
  - Ecosystem of DeSecure blockchains

- DeSecure blockchains uses novel Error-Correction Code PoW.

- We aim to provide two DeSecure blockchains, ETH-ECC and BTC-ECC.

They Have Sought Alternatives to SHA-PoW, BUT

	Pros	Cons	Coins within top 50 rank
PoW (Proof-of-Work)	• Strong security - Difficult to produce - Easy to verify	• Extreme computing power • 51% attacks • Transaction speed / Transaction throughput	Bitcoin, Ethereum
PoS (Proof-of-Stake)	• Energy & hardware efficiency • Much more expensive 51% attacks	• Recentralization • The rich-get-richer • "Noting at stake" problem	Qtum, Stratis
DPoS (Delegated PoS)	• Scalability and speed • Energy & hardware efficiency • Encouraging good behavior by real-time voting	• Recentralization • DDoS attacks	EOS, NEO smart economy
PoA (Proof-of-Activity)	• Much more expensive 51% attacks • Decentralization - Validators are randomly selected.	• Recentralization • Extreme computing power • The rich-get-richer	decred

Comparison to Existing Scalability Solutions

DeSecure Blockchain aims to resolve the re-centralization problem without sacrificing the securedness and decentralization!

Type	DeSecure	Bitcoin		Ethereum	
Name	Multi-level, multiple chains	Seg-Wit	Lightening Network	Plasma	Sharding
How	Many ECCPoW based chains can talk to each other via value-exchange service	Realize by modifying a block data structure	Allow off-chain transactions and record the end result of these transactions into the main blockchain	Allow transactions in child chains, TX records end up at the main chain are limited.	Divide BC DB with multiple shards
Pro	Many different services and levels of chains can co-work.	Easy to realize	Faster transactions Small TX fees	Faster transactions Small TX fees	Faster transaction
Con	No single chain solution Requires an ecosystem	Small improvement	The content of off-chain transactions lost	Some TX content lost Only full node can run this	Increased SW complexity

We aim to Replacing SHA-PoW with ECC-PoW!

Blockchain Core Program

Three key parts

1. Web server interface networking of peers

- Node registration, get-address, give-address
- Full node or light node
- Communication among the wallets and the miners

2. Wallet for TX generations

- Make private and public keys, address, store UTXOs, make TX, put signature, announce it to the neighbor, check to see if the TX is supported by the blockchain.

3. Consensus Mechanism

Consensus Engine

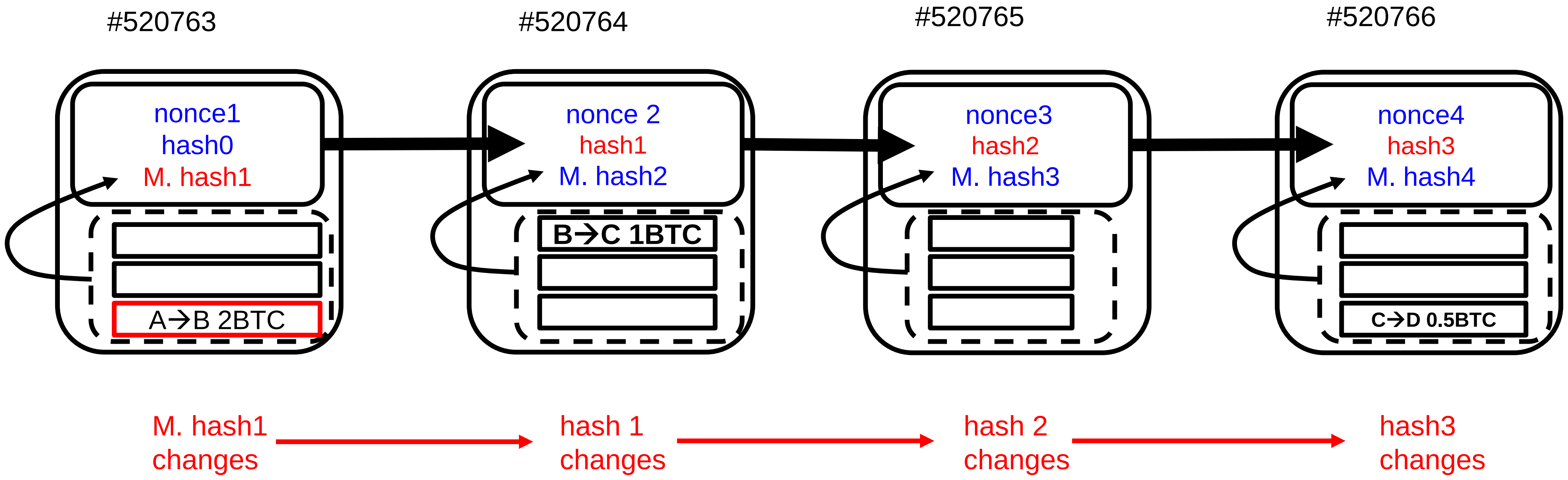
- Data: Genesis block + regular blocks, one block every 10 min, block-size 1Mbyte
- Protocol: consensus, block header, difficulty level adjustment, …
- Mining: Get the longest chain, validate it and all transactions within it, get transactions from mempool and form a block, run SHA repeatedly until you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

Program Suite

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

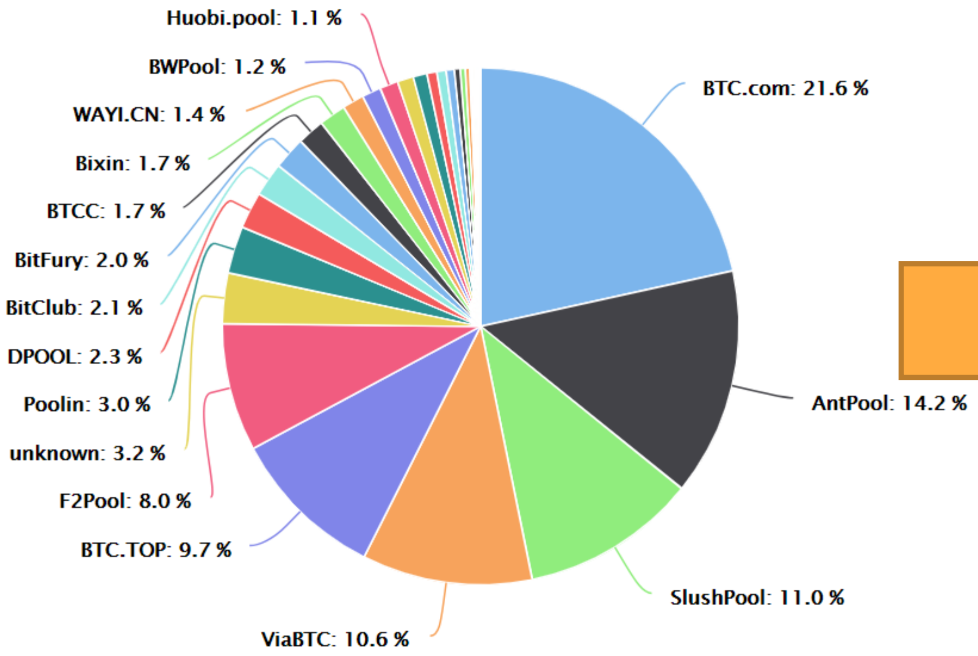
Pow is fundamental to OPEN blockchains!

- What happens when any alteration is made?
- Proof-of-Work (PoW)
- Immutability and openness allow transactions.
  - A → B 2 BTC
  - B → C 1BTC
  - C → D .5 BTC

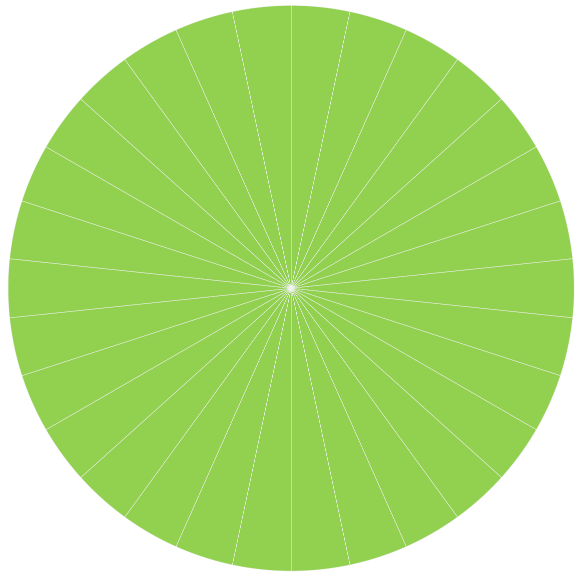


ECC-PoW aims to resolve Recentralization Issue.

- ASIC → Mining Moguls → Discourage Average Miners
- Prone to Collusion, Censorship



Recentralized



Decentralized again

1. ASIC resistant
2. Vulnerability to DS attacks reduced

There are items to consider for a new PoW!

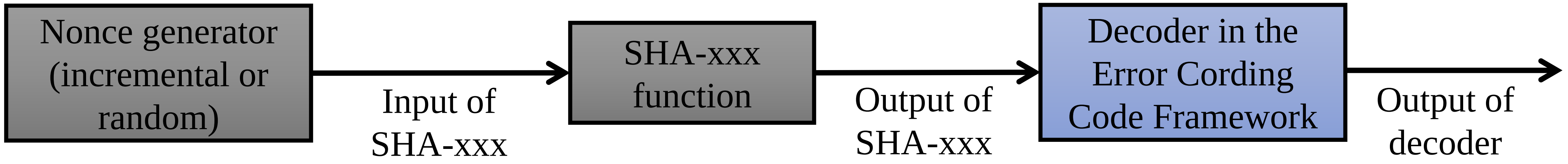
- A new puzzle generation system is capable of varying puzzles from block to block with the following properties:

  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.

Novel 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.

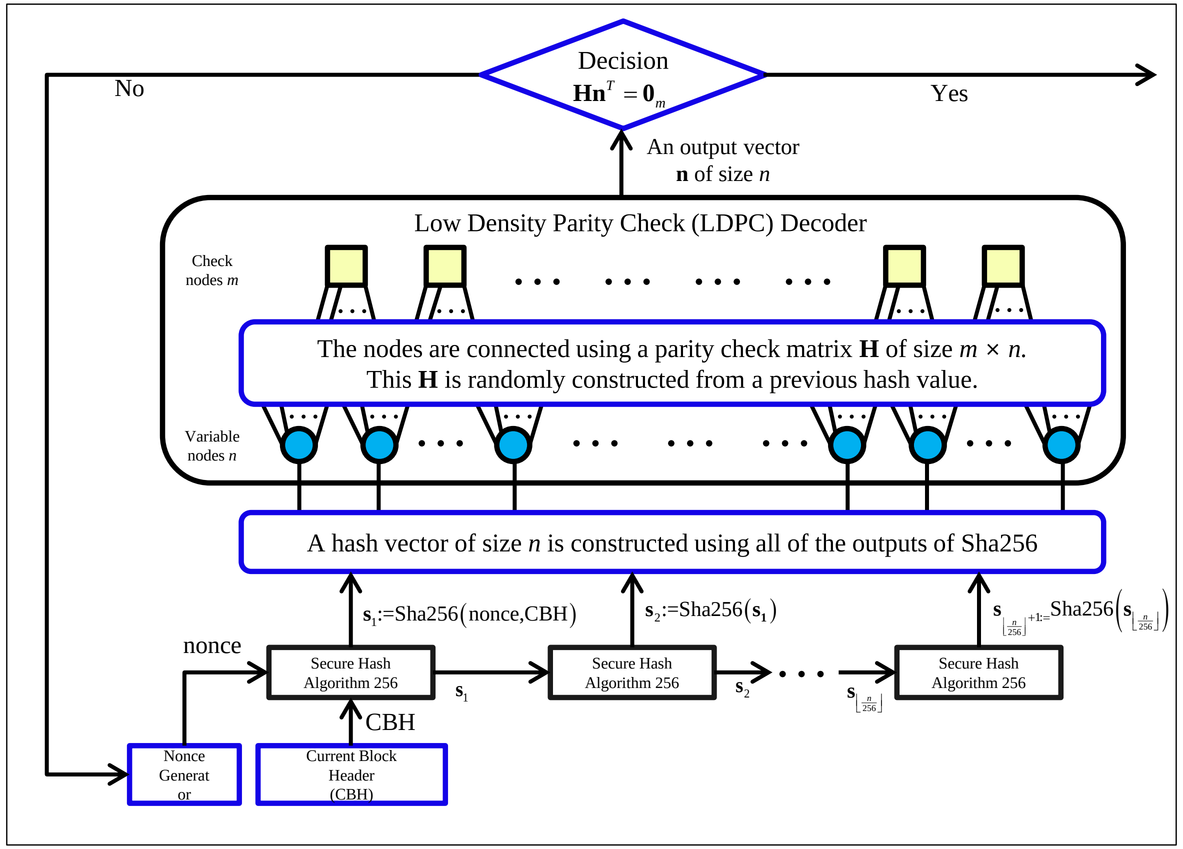


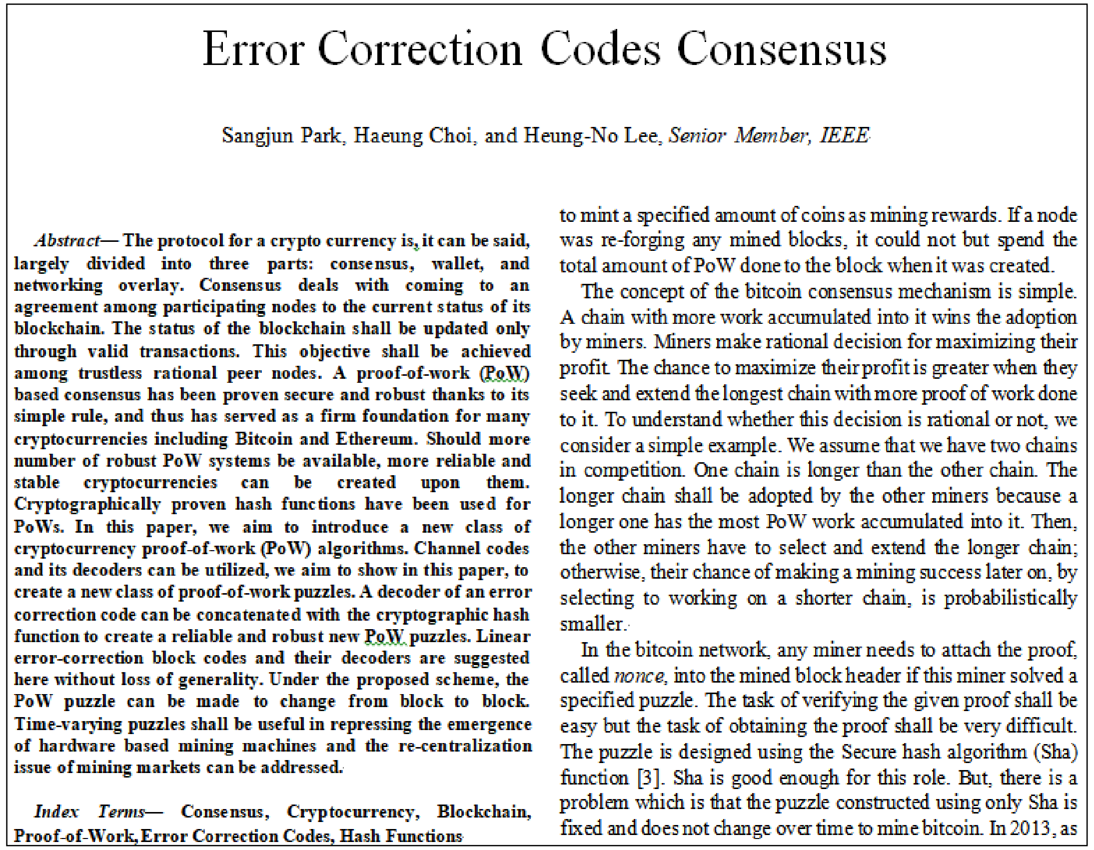
- The decision of mining success is made with the output of the above decoder.

Novel ECCPoW Consensus mechanism, how!

▣ ECCPoW Engine

- Compound code of SHA and LDPC decoder.
- Variable size of Parity Check Matrix (PCM) → Amt of resource (mem, comp) varies.
- PCM is varied by the hash of the previous block.





Error Correction Codes Consensus

Sangjun Park, Haeung Choi, and Heung-No Lee, *Senior Member, IEEE*

Abstract— The protocol for a crypto currency is, it can be said, largely divided into three parts: consensus, wallet, and networking overlay. Consensus deals with coming to an agreement among participating nodes to the current status of its blockchain. The status of the blockchain shall be updated only through valid transactions. This objective shall be achieved among trustless rational peer nodes. A proof-of-work (PoW) based consensus has been proven secure and robust thanks to its simple rule, and thus has served as a firm foundation for many cryptocurrencies including Bitcoin and Ethereum. Should more number of robust PoW systems be available, more reliable and stable cryptocurrencies can be created upon them. Cryptographically proven hash functions have been used for PoWs. In this paper, we aim to introduce a new class of cryptocurrency proof-of-work (PoW) algorithms. Channel codes and its decoders can be utilized, we aim to show in this paper, to create a new class of proof-of-work puzzles. A decoder of an error correction code can be concatenated with the cryptographic hash function to create a reliable and robust new PoW puzzles. Linear error-correction block codes and their decoders are suggested here without loss of generality. Under the proposed scheme, the PoW puzzle can be made to change from block to block. Time-varying puzzles shall be useful in repressing the emergence of hardware based mining machines and the re-centralization issue of mining markets can be addressed.

Index Terms— Consensus, Cryptocurrency, Blockchain, Proof-of-Work, Error Correction Codes, Hash Functions

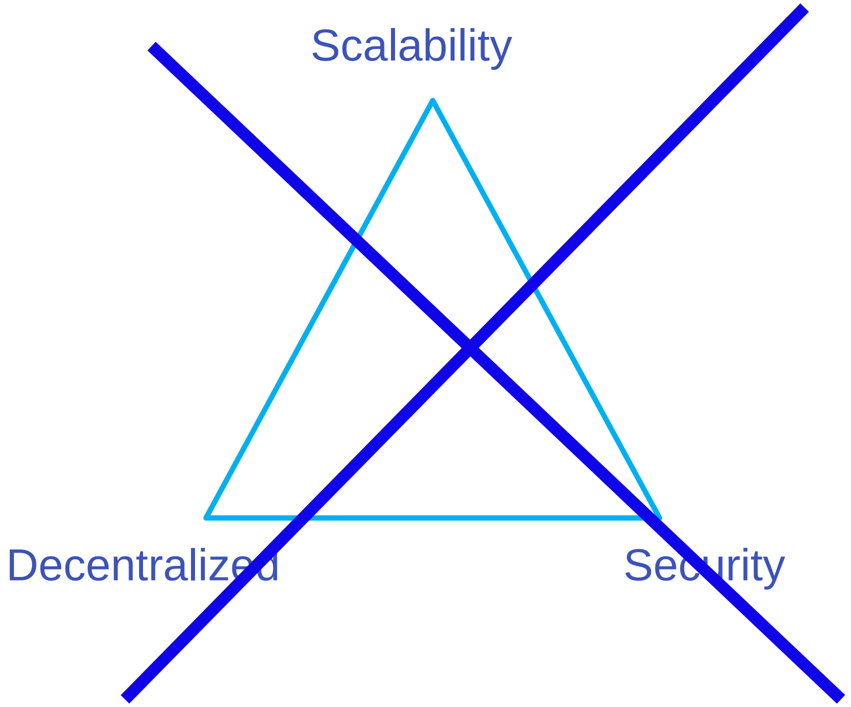
to mint a specified amount of coins as mining rewards. If a node was re-forging any mined blocks, it could not but spend the total amount of PoW done to the block when it was created.

The concept of the bitcoin consensus mechanism is simple. A chain with more work accumulated into it wins the adoption by miners. Miners make rational decision for maximizing their profit. The chance to maximize their profit is greater when they seek and extend the longest chain with more proof of work done to it. To understand whether this decision is rational or not, we consider a simple example. We assume that we have two chains in competition. One chain is longer than the other chain. The longer chain shall be adopted by the other miners because a longer one has the most PoW work accumulated into it. Then, the other miners have to select and extend the longer chain; otherwise, their chance of making a mining success later on, by selecting to working on a shorter chain, is probabilistically smaller.

In the bitcoin network, any miner needs to attach the proof, called *nonce*, into the mined block header if this miner solved a specified puzzle. The task of verifying the given proof shall be easy but the task of obtaining the proof shall be very difficult. The puzzle is designed using the Secure hash algorithm (Sha) function [3]. Sha is good enough for this role. But, there is a problem which is that the puzzle constructed using only Sha is fixed and does not change over time to mine bitcoin. In 2013, as

※ 국제 학술지 IEEE trans. Information Forensics and Security에 제출예정

Blockchain Trilemma?



“

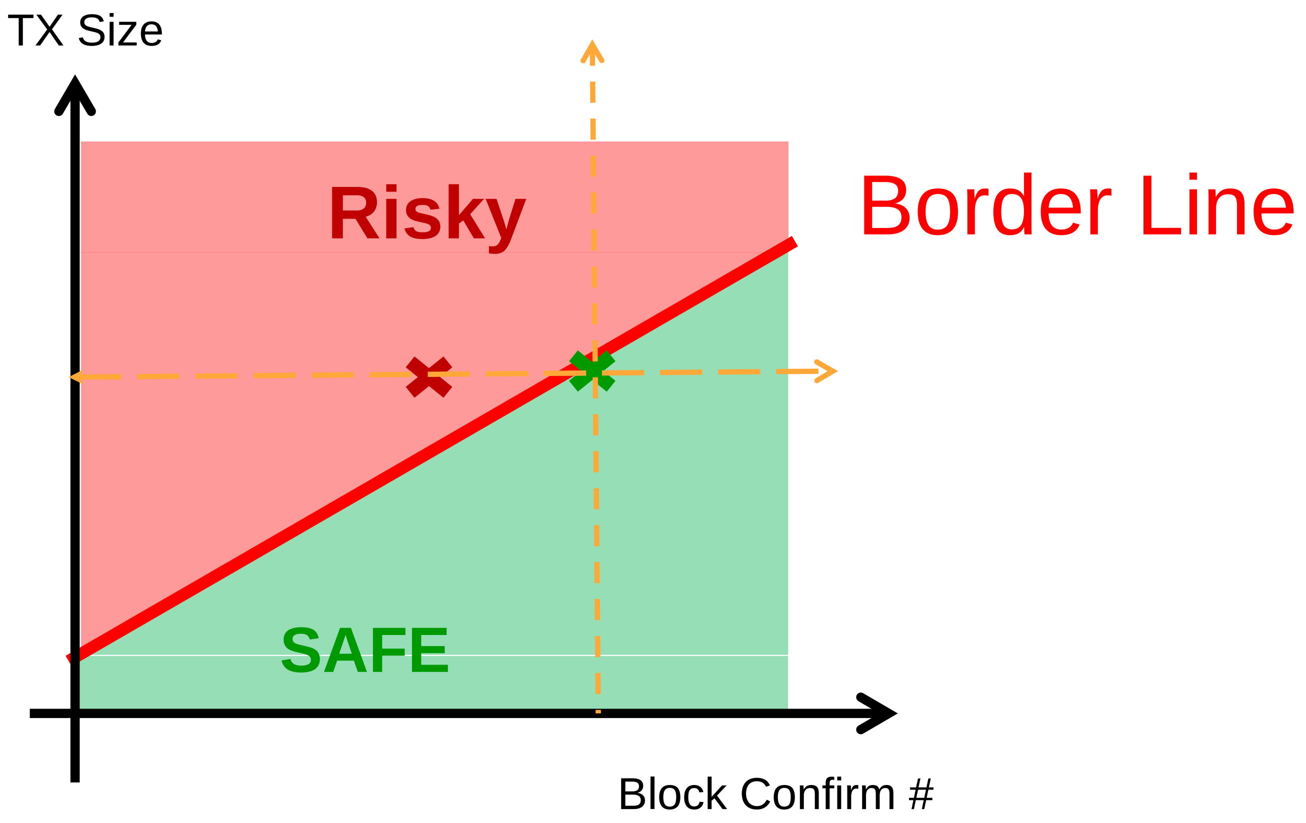
blockchain systems can only at most have two of the following three properties

- Vitalik Buterin, Sharding FAQ
https://github.com/ethereum/wiki/wiki/Sharding-FAQ

”

- Wrong approach!
- Not in a single blockchain, can it be achieved!
- ***We shall promote many decentralized secure (DeSecure) blockchains and approach the scalability problem!***

Profitable DS Risk Analysis



<Risk Analysis Algorithm>

Profitable Double-Spending Attacks

Jehyuk Jang and Heung-No Lee, *Senior Member, IEEE*

***Abstract*—Our aim in this paper is to investigate the profitability of double-spending (DS) attacks that manipulate a priori mined transaction in a blockchain. Up to date, it was understood that the requirement for successful DS attacks is to occupy a higher proportion of computing power than a target network's proportion; i.e., to occupy more than 51% proportion of computing power. On the contrary, we show that DS attacks using less than 50% proportion of computing power can also be vulnerable. Namely, DS attacks using any proportion of computing power can occur as long as the chance to making a good profit is there; i.e., revenue of an attack is greater than the cost of launching it. We have novel probability theory based derivations for calculating time finite attack probability. This can be used to size up the resource needed to calculate the revenue and the cost. The results enable us to derive sufficient and necessary conditions on the value of a target transaction which make DS attacks for any proportion of computing power profitable. They can also be used to assess the risk of one's transaction by checking whether or not the transaction value satisfies the conditions for profitable DS attacks. Two examples are provided in which we evaluate the attack resources and the conditions for profitable DS attacks given 35% proportion of computing power against *Syscoin* and *BitcoinCash* networks, and quantitatively shown how vulnerable they are.**

***Index Terms*— Blockchain, Bitcoin, Double-Spending Attack, Profit, Gambler's Ruin Theorem, Poisson Counting Process.**

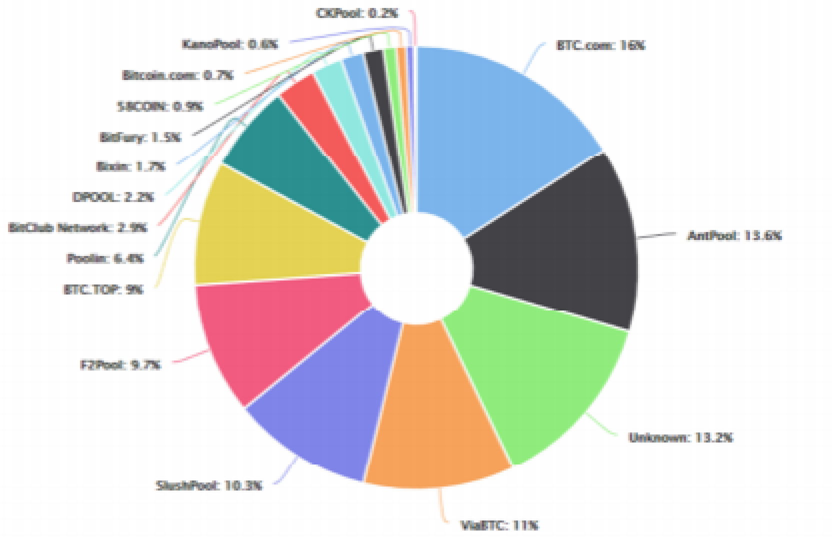


Fig. 1. Computation power distribution among the largest mining pools provided by *blockchain.com* (date accessed: 22 Oct. 2018).

peers in a network to share a common chain. If a full node succeeds in generating a new block, he/she has the latest version of the chain. All of the nodes in the network continuously communicate with each other to share the latest chain. If a node suffers from a conflict between two or more different chains, the consensus rule provides a rule that a single chain is selected. Satoshi Nakamoto suggested the *longest chain consensus* for *Bitcoin* protocol which conserves the longest chain among the conflictions [1]. There are also

※ Jahyuk will present tomorrow afternoon!

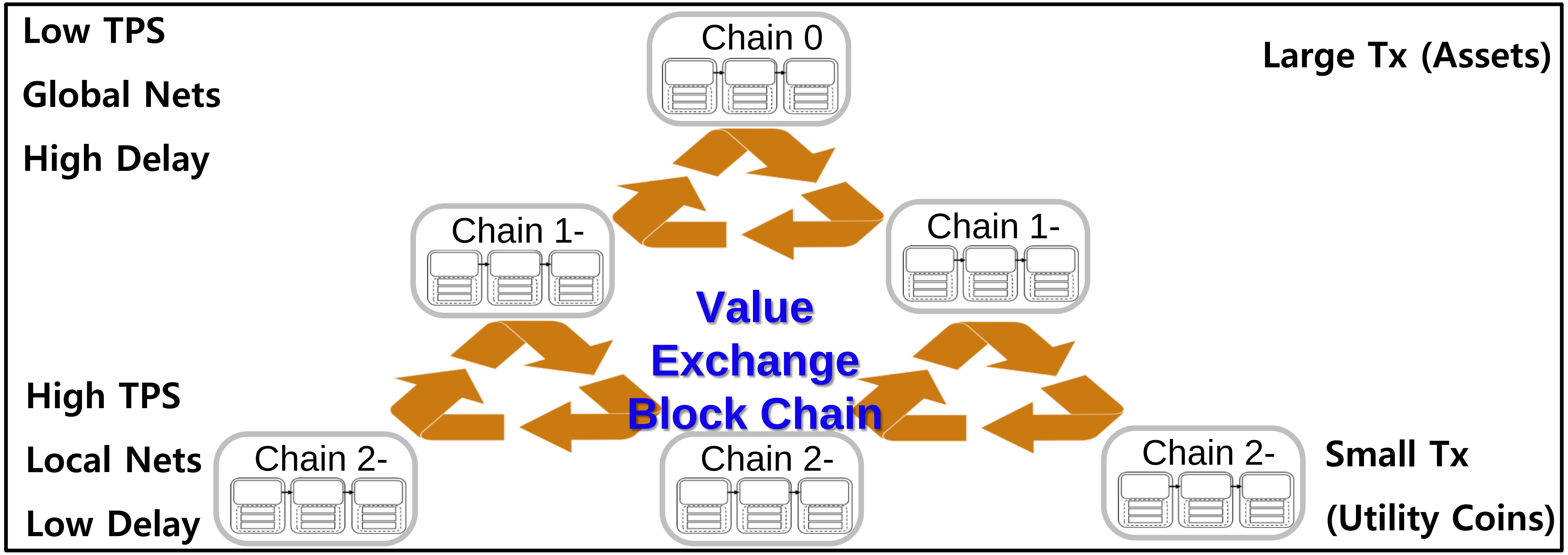
※ 국제 학술지 IEEE trans. Information Forensics and Security에 제출됨

https://arxiv.org/ftp/arxiv/papers/1903/1903.01711.pdf

Provision of DeSecure chains, use ecosystem to solve Scalability issue!

▣ Global chains → national chains → local chains

▣ One chain is designed to hold only up to 20 DApps



<Multi-level DeSecure chains>

Block code

- A block code C(N, Rate, **G**, **F**, ENC, DEC, 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. $N = 512$)
- Rate = $(N - M)/N$ is the rate of the code, where M < N.
- For example, with $N = 1024$ and $M = 256$, Rate = 3/4.

- **G** is the Generator matrix with dimension $N \times (N - M)$.
- **F** is the Check matrix with dimension, $M \times N$.
- **G** and **F** are orthogonal to each other, i.e., **FG** = **0**.

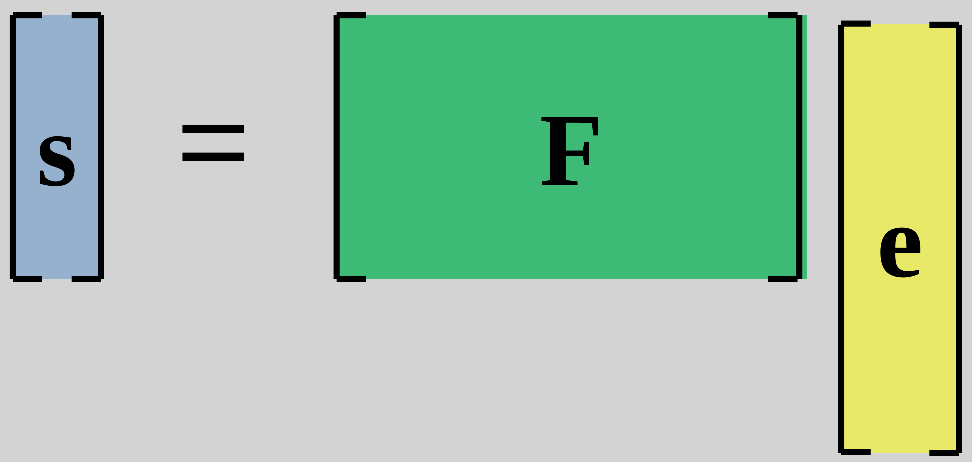
- A message vector m is an $(N - M) \times 1$ vector.
- A codeword **c**, an N x 1 vector, is an element of the code and can be generated by multiplying a message vector m to the Generator matrix **G**, i.e., **c** = **Gm**.

- Galois Field of size q, 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 **m** as the input and produces a codeword vector corresponding to it, e.g. **c** = ENC(**G**, **m**).

- DEC implies the decoding function; DEC takes an arbitrary vector e and returns a closest codeword **c**ˆ, i.e., **c**ˆ= DEC(**F**, **e**).



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

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

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

$$M < N$$

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

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

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 $\mathbf{F}\hat{\mathbf{c}} = \mathbf{0}$.

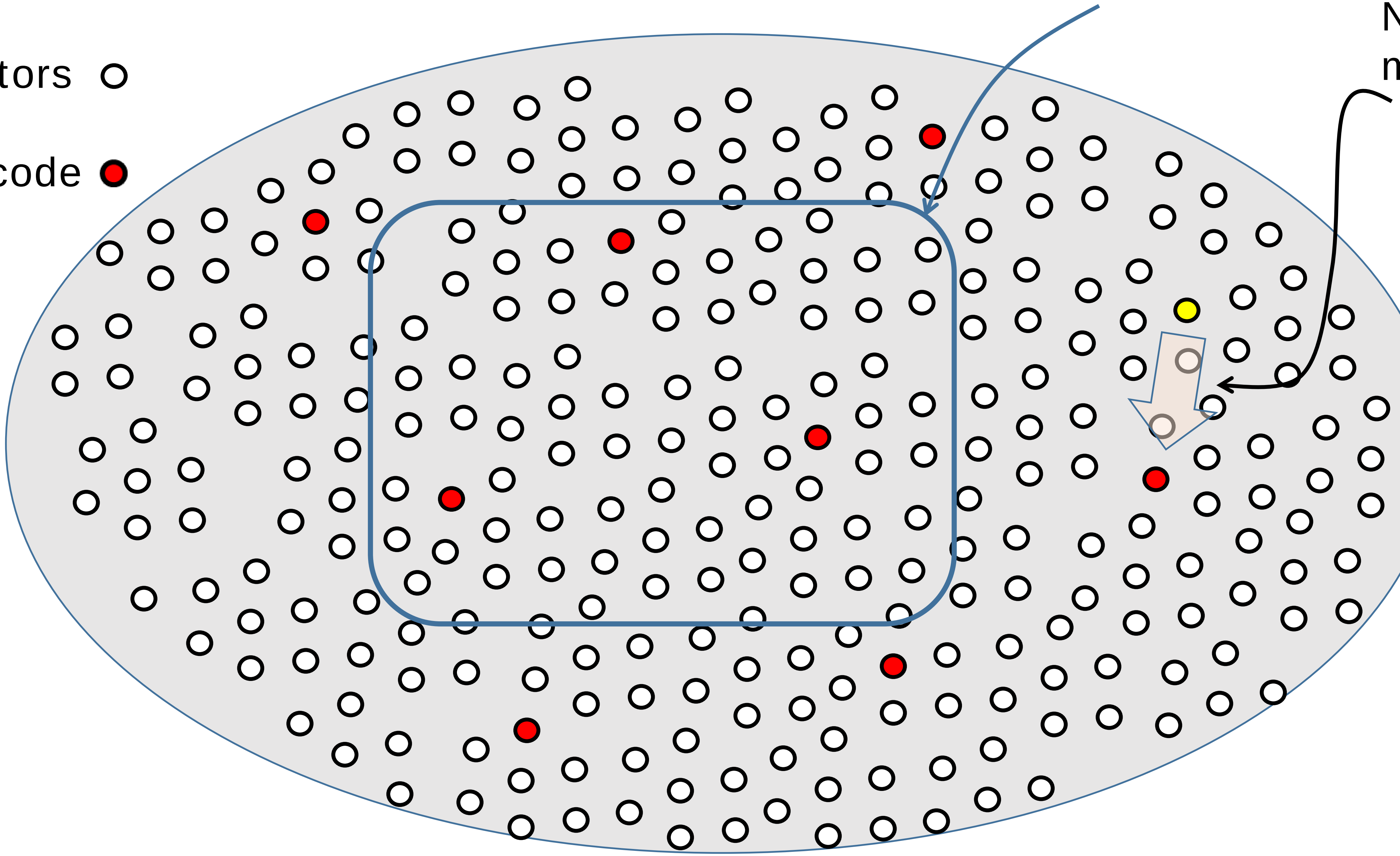
- We propose to use the message passing graph decoder for its excellency in accuracy and superiority in decoding speed.
  This is to prevent a cheating attack in which a smart miner comes up with a new decoder algorithm of his own developed and outpaces the regular miners using the designated decoder. If this is allowed, a hidden advantage goes to the smart miner.

Geometrical Explanation

2^{256} vectors ○

Rate ¼ code ● $= 2^{64}$

Condition set

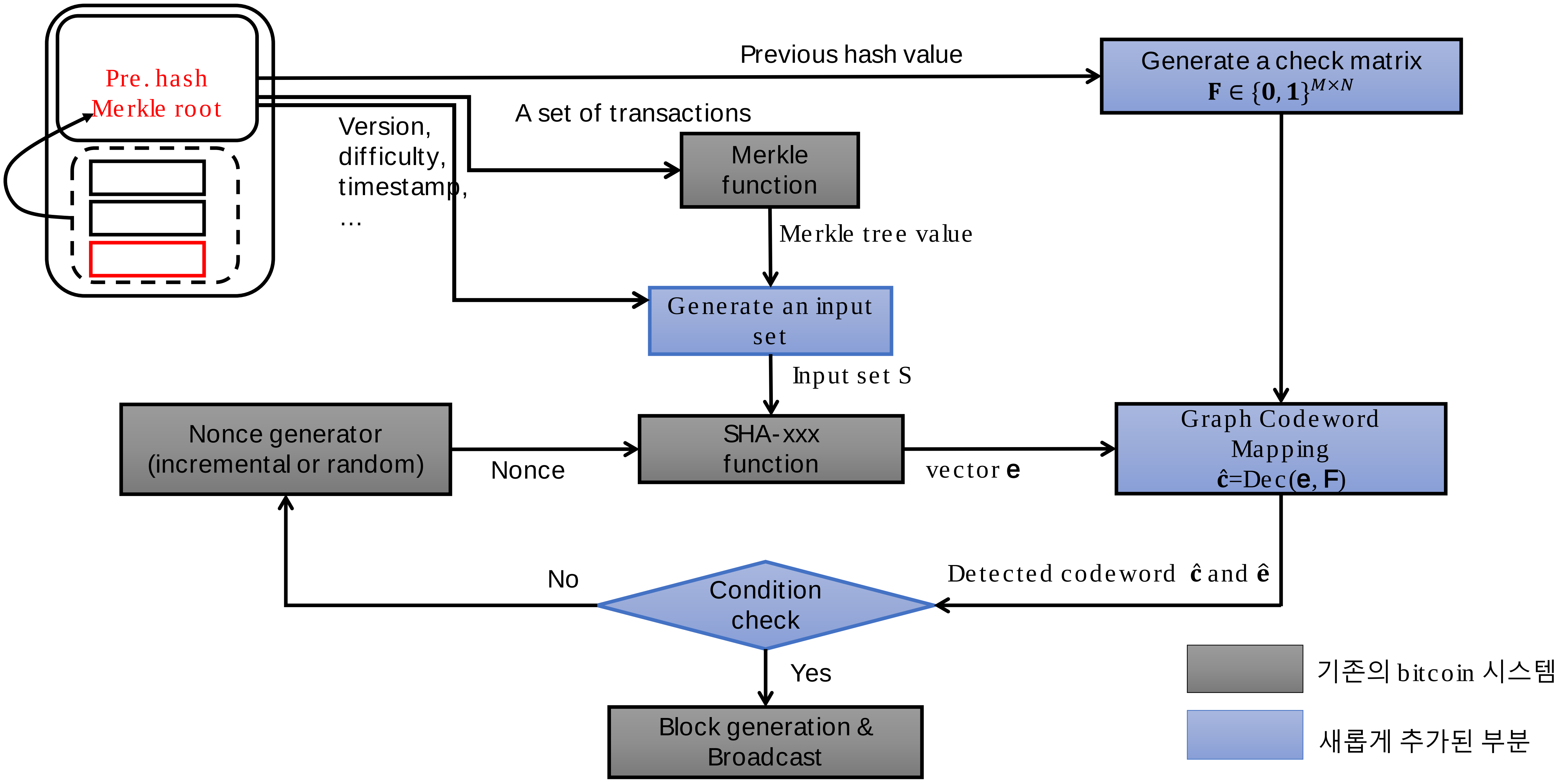


Nearest codeword mapping of $\mathbf{e}$ to $\hat{\mathbf{c}}$.

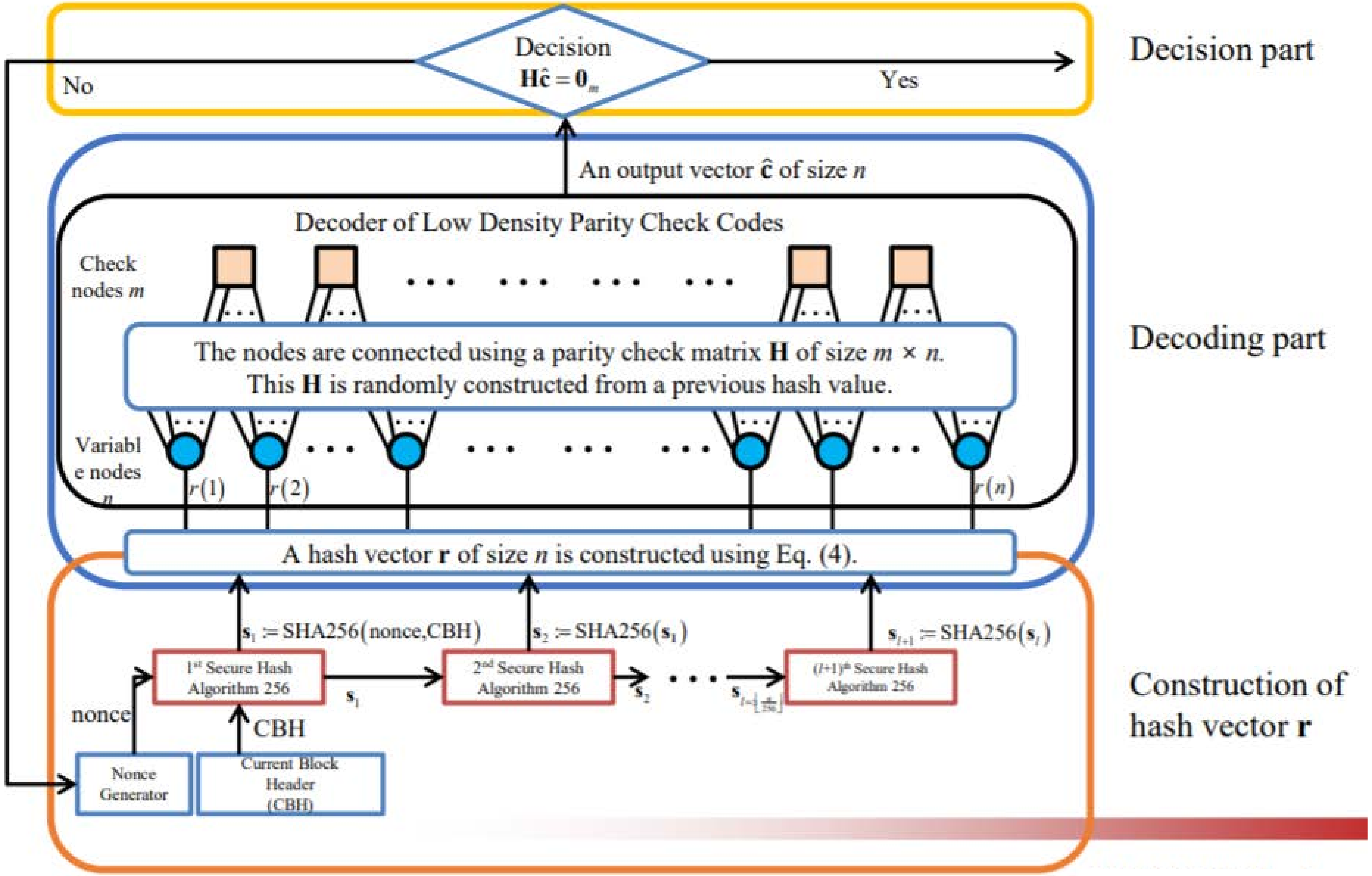
DEC(○) = ●

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

Diagram of ECCPoW

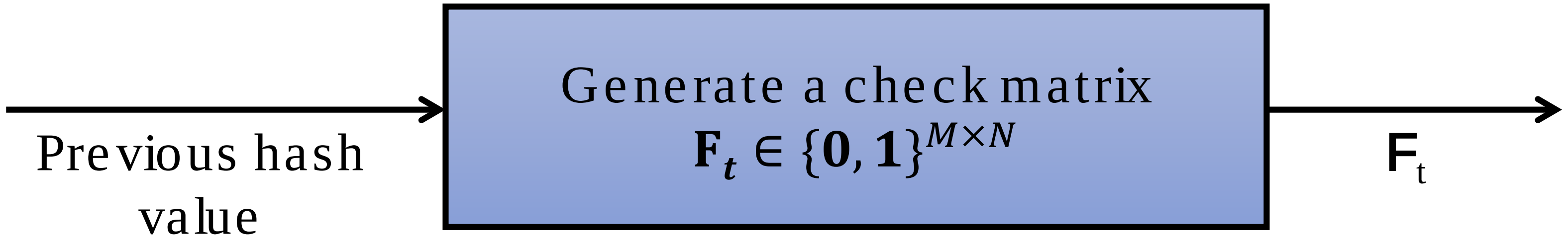


Big Picture of ECCPoW



Generate a Check Matrix

- Parameter set $S_t = \{h_{t-1}, \textit{code parameters}\}$;
- GenCheckMatrix(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 converse
- Update variable-to-check node messages for $t = 1, 2, .., N$ and $\forall l \in Q1(t)$:

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

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

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

- Output

✓ Hard decision of a posteriori LLR:

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

Implemented code in C

CMminer.c

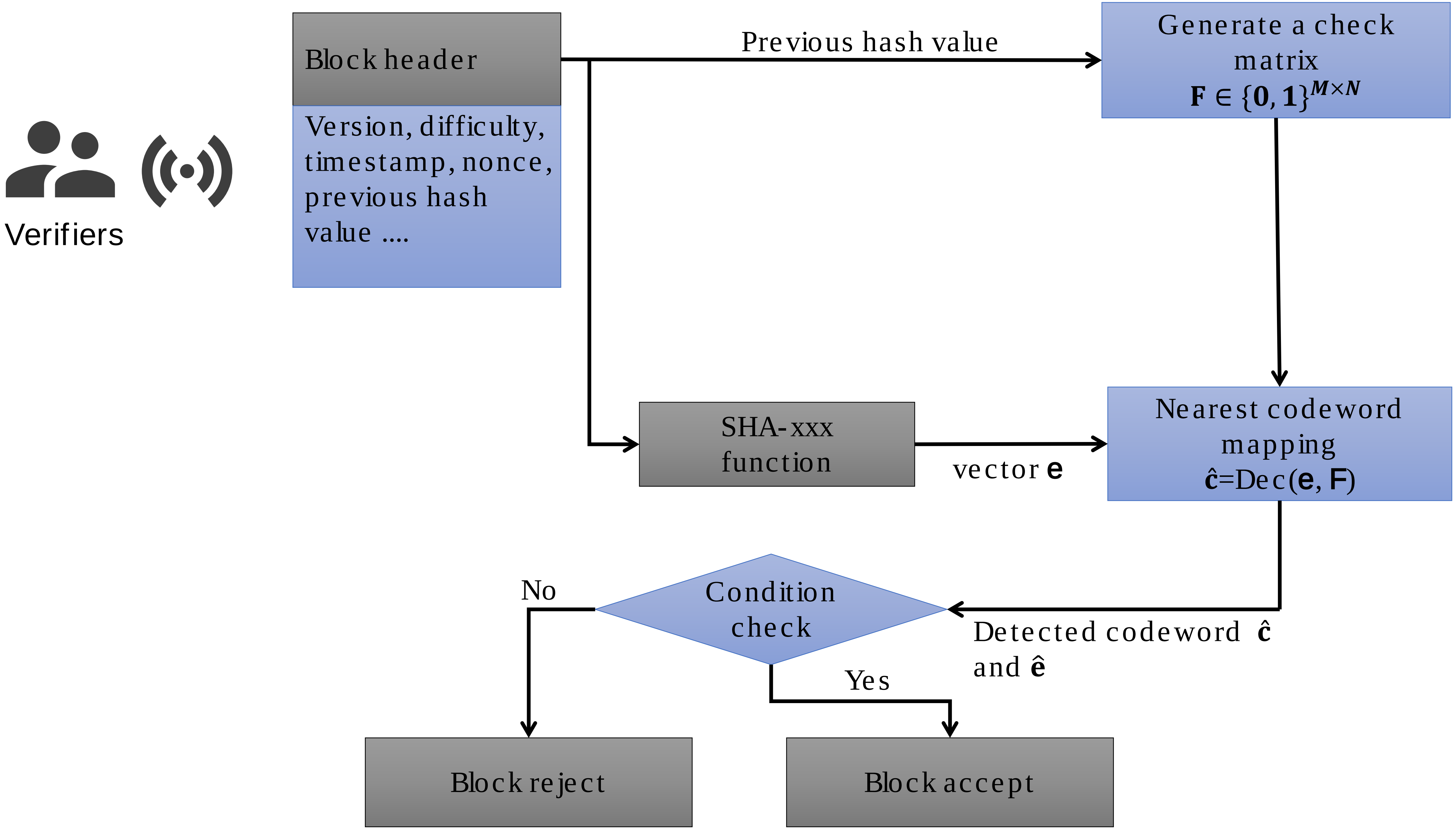
기타 파일 | (전역 범위) | NCMminer(double * BPSK_AWGN_Codeword)

```
            }
        //Bit to Check Node Messages --> LRqtl
        for(i = 0; i < ITERATIONS; i++) {
            for(k = 0; k < COLUMNS; k++) {
                for(l = 0; l < COLUMN_WEIGHT; l++) {
                    temp3 = LRft[k];
                    for (m = 0; m < COLUMN_WEIGHT; m++) {
                        if (m != l) {
                            temp3 += LRrtl[k][Row_In_Column[m][k]];//LRqtl[ k][Row_In_Column[l][k] ] =  infinity_test(LRqtl[ k][Row_In_Column[l][k] ]);
                        }
                    }
                    LRqtl[ k][Row_In_Column[l][k] ] =  (short) temp3;
                }
            }

            fprintf(out,"₩n₩n₩nLRqtl iteration %i₩n",i);
            for(k=0; k <COLUMNS; k++) {
                fprintf(out,"₩n");
                for(m=0;m<ROWS;m++)
                    fprintf(out,"%i ",LRqtl[k][m]);
            }

            //Check to Bit Node Messages --> LRrtl
            for(k = 0; k < ROWS; k++){
                for(l = 0; l < ROW_WEIGHT; l++){
                    temp3 = 0.0;
                    sign=1;
                    for( m =0; m < ROW_WEIGHT; m++){
                        if( m != l){
                            temp3 = temp3 + func_f( fabs( LRqtl[Column_In_Row[m][k]][k] ) );
                            if(LRqtl[Column_In_Row[m][k]][k] > 0.0)
                                temp_sign = 1;
                            else
                                temp_sign = -1;
                            sign=sign*temp_sign;
                        }
                    }
                    magnitude =  func_f(temp3);
                    LRrtl[ Column_In_Row[l][k]] [k] =   (short) sign*magnitude;
                }
            }
            fprintf(out,"₩n₩n₩nLRrtl iteration %i₩n",i);
            for(k=0; k <COLUMNS; k++){
                fprintf(out,"₩n");
                for(m=0;m<ROWS;m++)
                    fprintf(out,"%i ",LRrtl[k][m]);
            }
            //Last iteration get LR (pi)
            for(m =0; m < COLUMNS; m++){
                LRpt[m] = LRft[m];
```

Diagram of New Verifiers



New Functions in ECCPoW

- New functions
  1. int **H = GenCheckMatrix(int n, int w_c, int w_r, int seed);
  2. bool DEC(int **H, int *e, int n, int w_c, int w_r, 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,

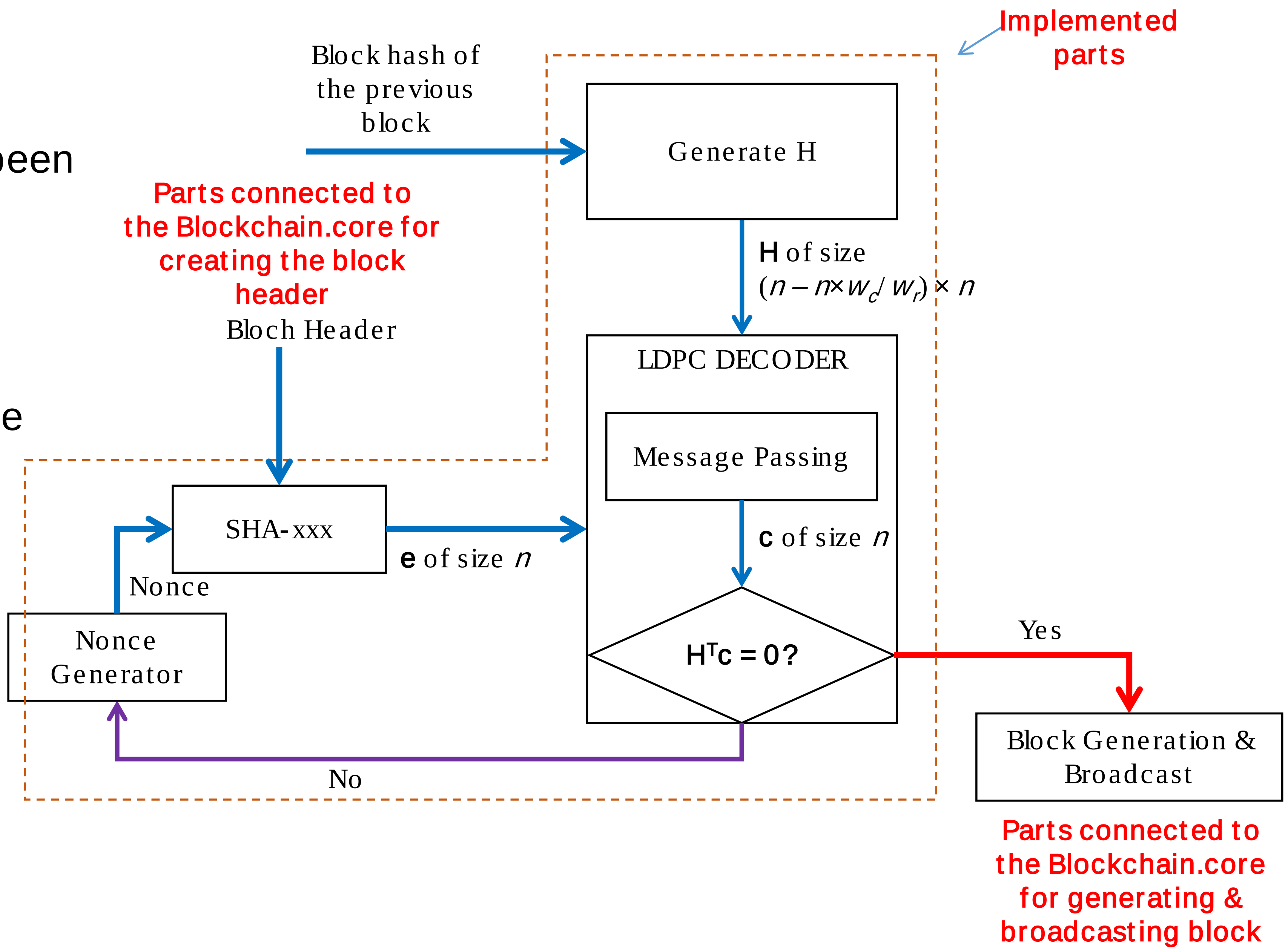
  three new functions are also have been introduced.

- We aim to link these functions to existing the blockchain. For example,

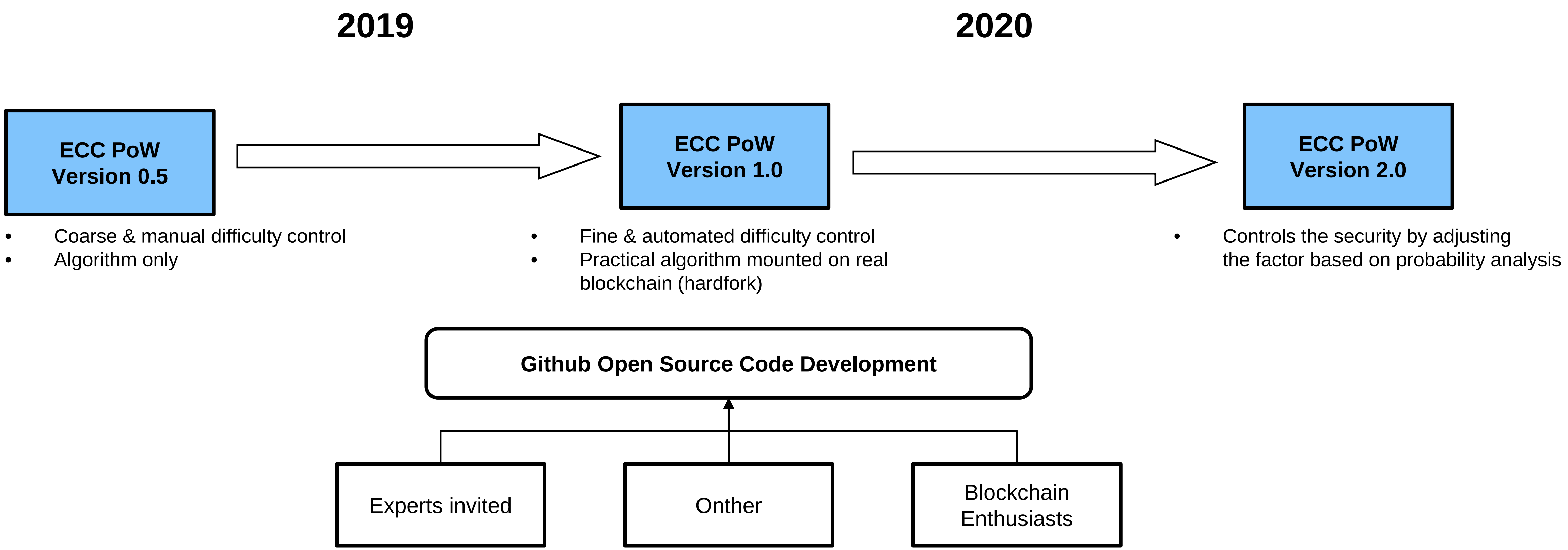
  mining function,

  chain validation function,

  consensus function and so on.



DeSecure Blockchain Release Plan



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.

- DeSecure 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 BTC-ECC and ETH-ECC protocols.

- Make the first blockchain network by running ETH-ECC over a network (Pusan coin)
- Make the second blockchain network by running BTC-ECC over other network (Gwangju coin)
- Make the third blockchain network by running ETH-ECC over another network (Seoul coin)
- Make the fourth blockchain network by running BTH-ECC 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 Scalibility, Decentralization and Security **can be achieved simultaneously**.

- With ECC, each blockchain is already very strong in decentralization.
- Each EEC blockchain is flexible enough to fit into 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 DeSecure chains 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.
- These DeSecure chains 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 serve to 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. We seek to get two Security and Decentralization.
  - Flexible puzzles enabled by ECCPoW can resolve the recentralization problem;
  - PoW has shown to be the most secure.
- Scalability is left to the ecosystem of DeSecure blockchains.
  - 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!

Selected References of GIST Blockchain Economy Center

- [Lee1] JH Jang and Heung-No Lee, "Profitable Double Spending Attacks," March 5th, 2019 submitted to IEEE Trans. Information Forensics and Securities, downloadable from https://arxiv.org/abs/1903.01711.
- [Lee2] 장재혁, 이흥노, "50%미만 이중 지불 공격", OSIA S&TR Journal, Vol. 32, No. 1, Mar. 2019. (pdf) (GIST 연구원 GRI 사업, 과학기술응용연구단 실용화사업)
- [Lee3] 정현준, 이흥노, "암호화폐 투자와 규제 현황", 한국정보과학회, 정보과학회지, 제 36권, 제 12호, pp. 49-56, Dec, 2018. (pdf)
- [Lee4] 박상준, 김형성, 이흥노, "Introduction to Error-Correction Codes Proof of Work," 블록체인경제 특집호, 대한전자공학회지, June 2019.
- [Lee5] Sangjun Park, HS Kim, Heung-No Lee, "Time-Variant Proof-of-Work Using Error-Correction Codes," to be submitted to IEEE Trans. Information Forensics and Securities.
- [Lee6] Mohamed Yaseen.J, Giljun Jung and Heung-No Lee."Decentralized Framework for Medical Images Based on Blockchain and Inter Planetary File System", The 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society(EMBC 2019), Berlin, Germany, Jul. 23-27, 2019.
- [Lee7] Please visit INFONET home page https://infonet.gist.ac.kr/?page_id=14 for more references.

- Thank you!

Heung-No Lee, GIST, South Korea
Home page: http://infonet.gist.ac.kr
Facebook/Publication ID: Heung-No Lee
E-mail: heungno@gist.ac.kr

- Q&A

- We are actively looking for blockchain students to join us.
- Send me an e-mail!