Blockchain Technology

Authors:

Nikhil Vanjani (14429)
4th Year B. Tech. CSE
IIT Kanpur

Abhishek Verma (14026)
4th Year B. Tech. CSE
IIT Kanpur

UGP Supervisor:

Dr. Arnab Bhattacharya
Associate Professor,
Computer Science and Engineering
Indian Institute of Technology, Kanpur
Kanpur, India

Abstract

- *There has been a buzz about blockchains in the recent years after the breakthrough of the Bitcoin blockchain in 2008. Subsequently various blockchains have been emerging.*
- *However, to say that blockchains are ready to take over the world would be nothing less than an exaggeration. There are a lot of fundamental issues yet unresolved including scalability, privacy, forks, regulators, network bootstrapping, and evolution.*
- *In this project, we studied about issues related to scalability, privacy, and forks. In particular, we focused on various consensus protocols and applying zero-knowledge proofs in blockchains.*
- *We will discuss about Algorand, a recently proposed algorithm based on Byzantine agreements.*
- *We will also discuss about zk-SNARKS (zero-knowledge succinct non-interactive argument of knowledge) which is the backbone of Zcash.*

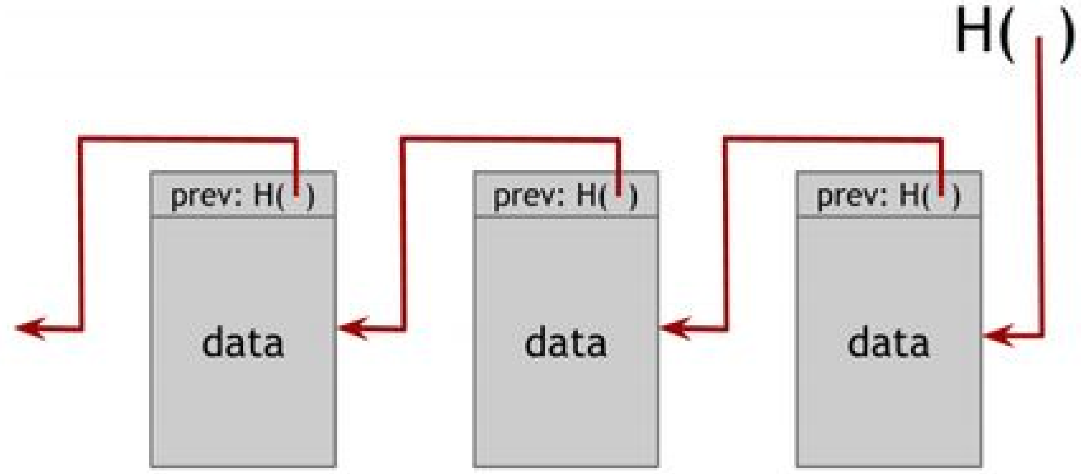
Outline

- Background
- Indian Government's initiatives towards adapting blockchains
- Consensus Protocols
- Algorand
- Zcash : a zero-knowledge blockchain

Background

- History
- Use Cases
- Major Challenges

- The idea dates back to 1970s when Ralph Merkle designed the Merkle Tree data structure
- Blockchains are immutable, append only, public ledgers



Source : Coursera: Bitcoins and Cryptocurrencies

- 1982: David Chaum's paper "*Blind Signatures for Untraceable Payments*" was the first major breakthrough in this direction
- 1998: PayPal
- 2008: Bitcoin -- the most popular blockchain based digital currency till date !

Background

- History
- Use Cases
- Major Challenges

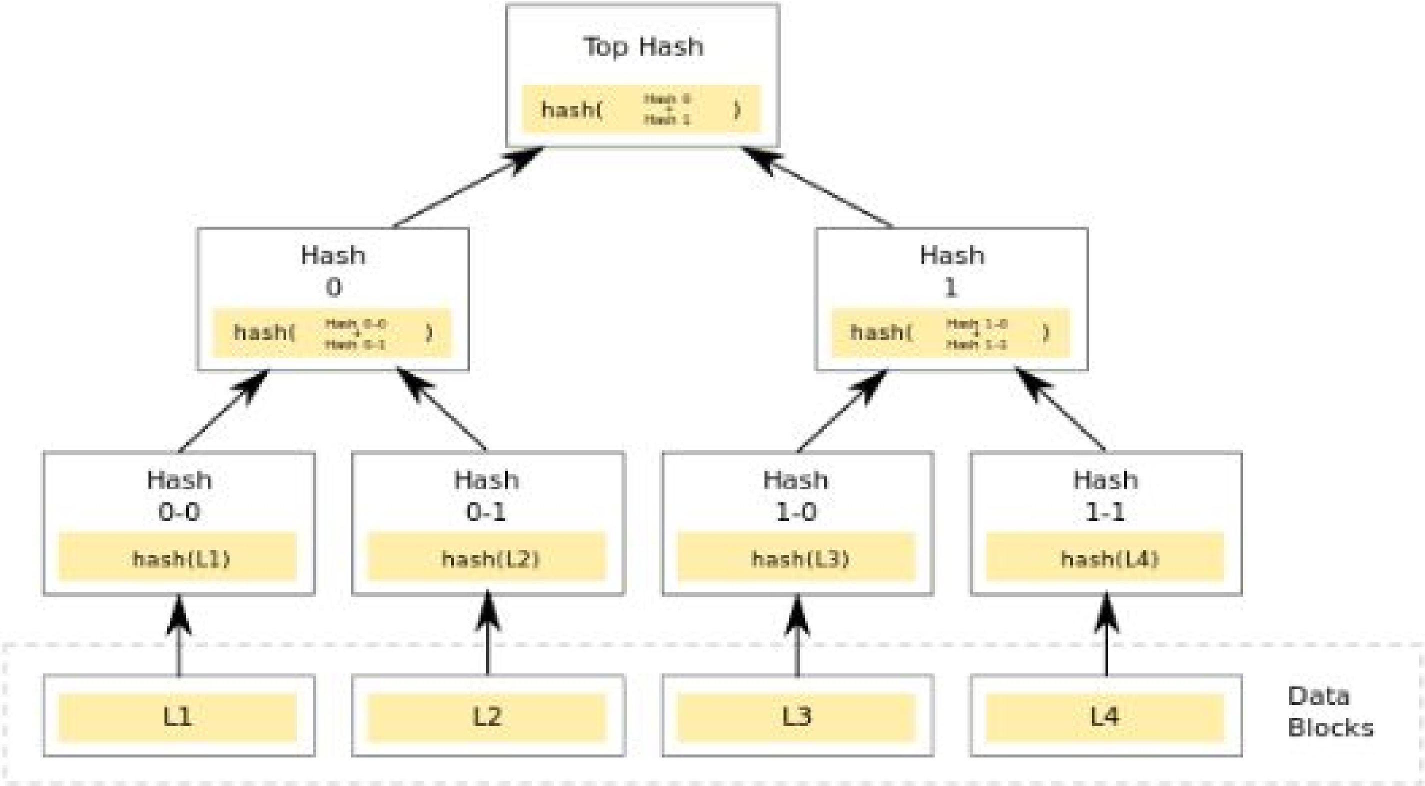
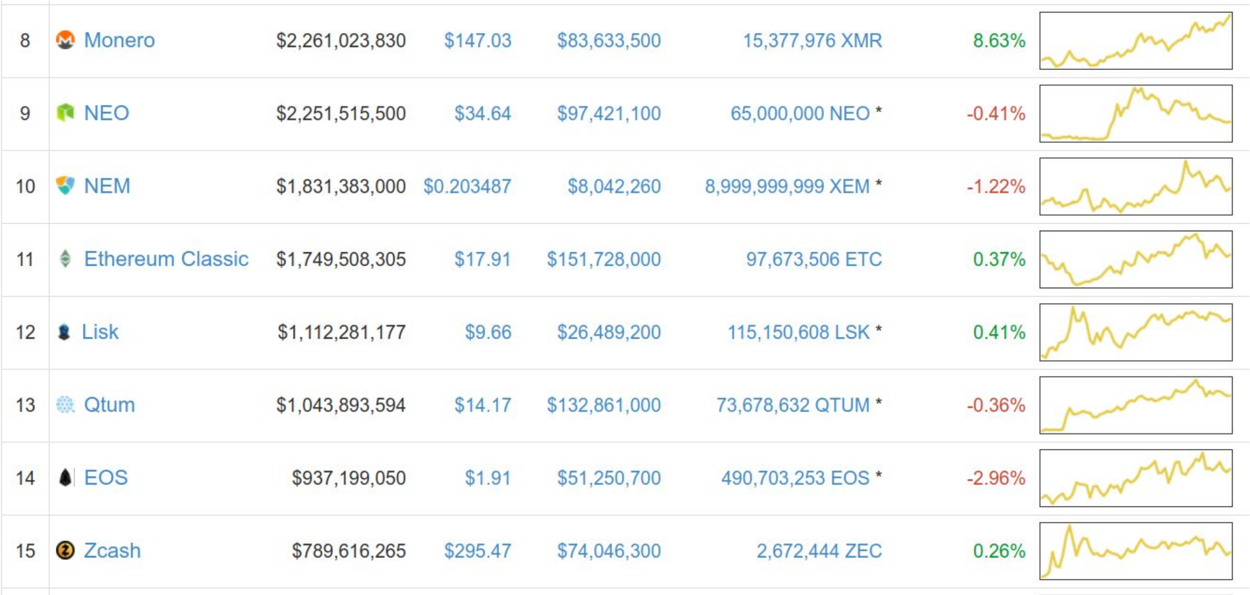


Figure : Merkle Tree
Source : By Azaghal - Own work, CC0,
https://commons.wikimedia.org/w/index.php?curid=18157888

Background

- History
- Use Cases
- Types of Blockchains
- Major Challenges

- ▷ But
  - ○ In today's digital age, popularity depends a lot on marketing the product !
- ▷ Do we really care about popularity ? No !
- ▷ No one ever marketed Bitcoin, yet it is most popular today
- ▷ Another classic example : Zcash vs Monero
  - ○ Both provide privacy, Zcash's is based on ZKP while more recently, Monero shifted to Ring Signatures after Zcash exposed it !



8	Monero	$2,261,023,830	$147.03	$83,633,500	15,377,976 XMR	8.63%
9	NEO	$2,251,515,500	$34.64	$97,421,100	65,000,000 NEO *	-0.41%
10	NEM	$1,831,383,000	$0.203487	$8,042,260	8,999,999,999 XEM *	-1.22%
11	Ethereum Classic	$1,749,508,305	$17.91	$151,728,000	97,673,506 ETC	0.37%
12	Lisk	$1,112,281,177	$9.66	$26,489,200	115,150,608 LSK *	0.41%
13	Qtum	$1,043,893,594	$14.17	$132,861,000	73,678,632 QTUM *	-0.36%
14	EOS	$937,199,050	$1.91	$51,250,700	490,703,253 EOS *	-2.96%
15	Zcash	$789,616,265	$295.47	$74,046,300	2,672,444 ZEC	0.26%

Background

- History
- Use Cases
- Types of Blockchains
- Major Challenges

- ▷ Bitcoin Currency + Bitcoin Blockchain
  - ○ Bitcoin
- ▷ Bitcoin currency + non-bitcoin blockchain
  - ○ Sidechains like Blockstream, Truthcoin
- ▷ Non-bitcoin currency + bitcoin blockchain:
  - ○ Factom
  - ○ Namecoin
- ▷ Non-bitcoin currency + non-bitcoin blockchain:
  - ○ Ethereum
- ▷ Non-blockchain consensus:
  - ○ MaidSafe
- ▷ Blockchain-neutral smart services:
  - ○ PeerNova

Background

- History
- Use Cases
- Types of Blockchains
- Major Challenges

	Public Blockchain	Private Blockchain	Permissioned/ Consortium Blockchain
Read / write	anyone	controlled	controlled
regulator	no	Yes - single	Yes - consortium
Decentralized ?	yes	no	partially
Trusted ?	no	yes	low
Scalability ?	no	yes	yes
Privacy ?	no	yes	yes

Background

- History
- Use Cases
- Types of Blockchains
- Major Challenges

- ▷ Scalability
  - ○ Speed
  - ○ Cost
  - ○ Space
- ▷ Network Bootstrapping and Evolution
- ▷ Forks and Regulators
- ▷ Privacy

- **Our study focused on Scalability, Forks and Privacy**

Indian Government Initiatives

- BankChain [1] : community of banks for exploring, building and implementing blockchain solutions.
  - Formed in February 2017 with State Bank of India being the first member, BankChain now has 27 members from India and the Middle East
  - Projects
    - Secure documents
    - Peer-to-peer payments
    - Asset / Charge registry
    - Syndication of loans
    - Blockchain Security Controls
    - Know Your Customer
    - Blockchain Libraries
    - Virtual currencies
    - Cross border payments
    - Trade finance
- Institute for Development and Research in Banking Technology -- RBI's research Wing
  - Whitepaper - "*Applications of Blockchain Technology to Banking and Financial Sector in India*" [2]

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

- ▷ Digital Currencies since 1980's but not decentralized ! Why ?
  - ○ Need a way to arrive at Consensus
  - ○ No good algorithms existed !
- ▷ Proof of Work was proposed in 1990s
  - ○ Use cases: mitigating spam emails, DoS attacks
- ▷ Bitcoin : consensus protocol - PoW ... so what's new ?
  - ○ It combined PoW with a monetary incentive model
  - ○ Introduced Miners to arrive at consensus
- ▷ Mining : CPUs -> GPUs -> FPGAs-> ASICs -> clusters of ASICs
  - ○ Economic issues :costs a lot to set up mining facility
  - ○ Environmental issues : wastes lot of electrical power
- ▷ Recent Research in Consensus Protocols
  - ○ Altering PoW to circumvent its shortcomings
  - ○ Exploring alternatives to PoW, keeping incentive models in place -- eg: PoS, Proof of Space
  - ○ Exploring alternatives to PoW, altering incentive models as well -- eg: Algorand

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

▷ PoW was introduced by Dwork and Naor[3] to mitigate spam emails. In general, the idea is each task is accompanied by a value which is moderate hard to compute but easy to verify.

▷ Most common example is to find a value=x st. Hash(task || x) starts with t zero bits.

▷ This makes it a random process with success probability 1/(2^t) and requires on an average 2^t attempts to get a satisfactory value x.

▷ In bitcoin blockchain, new blocks are validated by miners. In order for a block to be accepted by network participants, miners must complete a **proof of work** on all the data in the block

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

- Each block contains the hash of the preceding block which means each block has a chain of blocks that together contain a large amount of work.
- Changing a block(which can only be done by creating a new block containing the same predecessor) requires regenerating all successors and redoing the work they contain.
  - This protects the blockchain from tampering.
  - The number of successors is relevant when qualifying the validity of a block: at least 6 successors are believed sufficient to consider a block valid.
- Problems
  - PoW consumes lot of electricity and computational power.
  - Hardware costs are high.
  - Mining centralization is always a risk.
  - Miners need some incentive to work honestly.

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

- PoS is a newer type of consensus algorithm by which a blockchain network aims to achieve distributed consensus.
- In PoS based cryptocurrencies, miner for next block is chosen by random selection where probability of being chosen is proportional to wealth.
- But, selection by account balance would result in centralization. To overcome this, we use several other selection methods:
    - Randomized block selection: Nxt and Blackcoin use randomization to predict next generator. They use a formula that looks for the lowest hash value in combination with the size of stake.
    - Coin age based selection: Peercoin’s PoS system combines randomization with concept of coin age(product of number of coins and number of days the coins have been held). The nodes with largest coin age compete for the next block and the generator selection amongst them is decided randomly. However, once a stake of coins has been used to sign a block, they must start over with zero coin age.

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

▷ Advantages

- Energy consumption is less as compared to PoW.
- No latency: PoS avoids the computational overhead of PoW and therefore allows reducing transaction confirmation time.
- Here, the motivation of a generator(which owns a large amount of coins) is to “guard” the coins.

▷ Shortcomings

- “Nothing at stake” problem: Block generators have nothing to lose by voting for multiple blockchain histories, which in turn prevents the consensus from ever resolving. Ethereum tries to solve this problem by using “Slasher protocol” which punishes the cheater who forges on top of more than one blockchain branch.
    - This also has issues. Attacker may split their credit among several users, so even when they are caught, they will be penalized less

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

- ▷ PoW: basic idea - dedicate non-trivial computational work for serving every request
- ▷ PoSp[4]: basic idea - dedicate non-trivial disk space for serving every request
- ▷ Motivation: often users have free disk space ! PoSp protocol will be free of cost
- ▷ Abstract Protocol:
  - ○ Communication between Prover P, Verifier V in 2 phases
  - ○ After Initiation Phase:
    - ■ P stores data D of size S
    - ■ V stores small piece of information
  - ○ Verification Phase
    - ■ V initiates some query
    - ■ P executes query and responds which V accepts/rejects
  - ○ Requirement
    - ■ V is highly efficient in both phases
    - ■ P is highly efficient in executing queries

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

▷ Actual Scheme:

- ○ Based on hard to pebble problems
- ○ Initiation Phase:
  - ■ V send description of a hash function to P
  - ■ P labels nodes of a hard to pebble graph using this function. Labelling is along the lines of creating a Merkle Tree
  - ■ V also computes Merkle Tree and sends Merkle Hash to P
- ○ Verification Phase:
  - ■ V asks P to tell label values of some random nodes

Consensus Protocols

- Detour
- Proof of Work (PoW)
- Proof of Stake (PoS)
- Proof of Space (PoSp)
- Byzantine Agreements

▷ Byzantine fault is any fault presenting different symptoms to different observers.

▷ The objective of Byzantine Fault Tolerance (BFT) is to be able to defend against Byzantine failures, in which components of a system fail with symptoms that prevent some components of the system from reaching agreement among themselves.

▷ BFT can be achieved if the loyal (non-faulty) generals have a unanimous agreement on their strategy.

▷ Honey Badger [5] demonstrated how BFT can be used to build a cryptocurrency-
  ○ Designates a set of servers to be in charge of reaching consensus on the set of approved transactions
  ○ Bad !! not decentralized !

▷ Hybrid consensus [6,7,8] refines the approach of using the Nakamoto consensus to periodically select a group of participants (eg: every day), and runs a Byzantine agreement between selected participants to confirm transactions until new servers are selected
  ○ Bad !! because it is PoW based, forks are still possible !

Algorand[9]

- **Big Picture**
- Assumptions
- Procedure
- Tentative Consensus
- Advantages
- BA* protocol
- Problems and Solutions

▷ Algorand is a new cryptocurrency that has properties:

- No Latency : confirms transactions with latency on the order of a minute while scaling to many users.
- No Forks : ensures that users never have divergent views of confirmed transactions, even if some of the users are malicious and the network is temporarily partitioned.
- uses Byzantine Agreement protocol (called BA*) to reach consensus among users on the next set of transactions.
- Security: Algorand uses a novel mechanism based on Verifiable Random Functions (VRFs) that allow users to privately check whether they are selected to participate in the BA.

Algorand

- Big Picture
- **Assumptions**
- Procedure
- Tentative Consensus
- Advantages
- BA* protocol
- Problems and Solutions

▷ Most honest users (e.g., 95%) can send messages that will be received by most other honest users (e.g., 95%) within a known time bound.

- ○ An adversary cannot manipulate the network at large scale.
- ○ If an adversary does gain full control of the network, safety is still guaranteed so long as the period of adversary control is bounded (e.g., at most one day or one week)
- ○ Following this period the network is strongly synchronous again for a reasonably long time (a few hours)

Algorand

- Big Picture
- Assumptions
- **Procedure**
- Tentative Consensus
- Advantages
- BA* protocol
- Problems and Solutions

▷ There is no distinction between miners and users, all Algorand users are created equally and communicate through a gossip protocol.

▷ Messages are signed using the private key of the sender, and peer selection is weighted based on how much money a peer has.

▷ The gossip protocol is used by users to submit new transactions, and every user collects a block of pending transactions that they hear about.

▷ In each round a set of users are chosen at random (in a fully decentralised way using cryptographic sortition) to propose a block.

Algorand

- Big Picture
- Assumptions
- **Procedure**
- Tentative Consensus
- Advantages
- BA* protocol
- Problems and Solutions

▷ Cryptographic sortition process provides lottery winners with a priority (comparable between users) and a proof of that priority — priority is used to determine which of several proposed blocks everyone should adopt.

▷ Selected users propose their block together with their priority and proof using the gossip protocol.

▷ To reach consensus on a proposed block, Algorand uses the BA★ agreement protocol.

Algorand

- Big Picture
- Assumptions
- Procedure
- **Tentative Consensus**
- Advantages
- BA* protocol
- Problems and Solutions

▷ Tentative consensus can be produced in one of two cases:

  ○ When the network is strongly synchronous, an adversary may with small probability be able to cause BA★ to reach tentative consensus on a block. In a few rounds Algorand will reach final consensus on a successor, with overwhelming probability, and thus confirm earlier transactions.

  ○ If the network was weakly synchronous (e.g., controlled by an adversary) then BA★ can reach tentative consensus on two different blocks, creating a fork. Algorand automatically repairs these by periodically running BA★ to reach consensus on which *fork* to use going forward.

Algorand

- Big Picture
- Assumptions
- Procedure
- Tentative Consensus
- **Advantages**
- BA* protocol
- Problems and Solutions

▷ Unlike many proof-of-stake schemes though, malicious leaders cannot create forks in the network.

▷ The weights are only there to ensure an attacker cannot amplify their power by using pseudonyms — so long as an attacker controls less than 1/3 of the monetary value Algorand can guarantee that the probability for forks is negligible.

Algorand

- Big Picture
- Assumptions
- Procedure
- Tentative Consensus
- Advantages
- **BA* protocol**
- Problems and Solutions

▷ The inputs to BA★ are a *context* capturing the current state of the ledger (sortition seed, user weights, and the last agreed-upon block), the *round number*, and a new *proposed block* from the highest priority block proposer.

▷ As its output, if the highest-priority block proposer was honest, BA★ reaches final consensus, otherwise it may declare tentative consensus.

Algorand

- Big Picture
- Assumptions
- Procedure
- Tentative Consensus
- Advantages
- BA* protocol
- **Problems and Solutions**

- Problems:
  - Sybil attacks
  - Scalability
  - Resilience to DoS attacks
- Solutions:
  - Weighted Users: It solves the problem of Sybil Attacks
  - Consensus by Committee: It solves the scalability issues. But having a committee implies targeted attacks against committee members
  - Cryptographic Sortition : prevents adversary from targeting committee members. It selects committee members in a private and non interactive way, but adversary may target a member after he sends message in BA*.
  - Participant Replacement : mitigates above attack by requiring each member to speak only once. This mitigates DOS attacks as well.

Zcash[10]

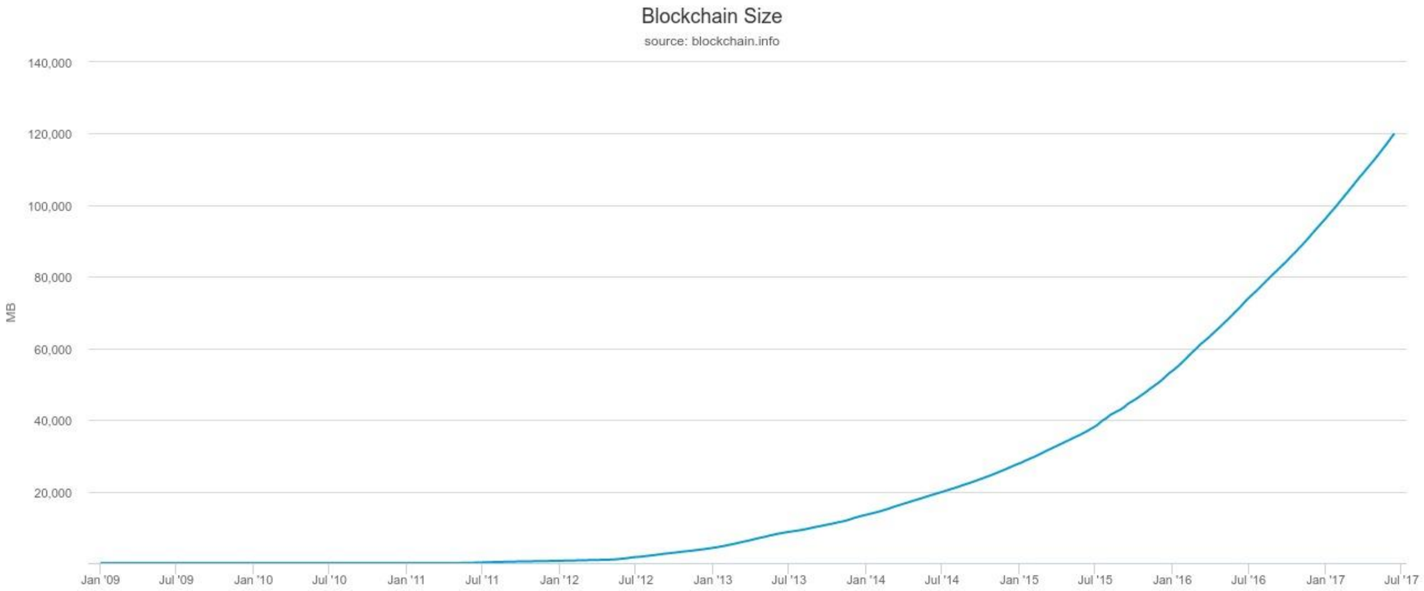
- Zero-Knowledge Proofs
- zk-SNARK

▷ Zcash is privacy preserving protocol designed on top of Bitcoin

▷ Zcash enables users to do transactions without revealing source, destination and amount.

▷ Potential application of zk-SNARK[11] for Bitcoin:

  ○ Lightweight Clients: Blockchain Compression
  “Here’s a summary of the 24 GB Blockchain with head H”



Source :
https://www.newsinbit.com/bitcoin-blockchain-size-reaches-120gb-and-increasing/

Zcash

- Zero-Knowledge Proofs
- zk-SNARK
- Homomorphic Hidings
- Blind Evaluation of Polynomials
- Knowledge of Coefficient Test and Assumption
- Making Blind Evaluation of Polynomials Verifiable
- From Computations to Polynomials

- zero knowledge- Succinct Non-interactive ARgument of Knowledge
  - Homomorphic Hidings [12]
  - Blind Evaluation of Polynomials [13]
  - Knowledge of Coefficient Test and Assumption [14]
  - Making Blind Evaluation of Polynomials Verifiable [15]
  - From Computations to Polynomials [16]

- The following slides are based on all the blogs cited above.

Zcash

- Zero-Knowledge Proofs
- zk-SNARK
- Homomorphic Hidings
- Blind Evaluation of Polynomials
- Knowledge of Coefficient Test and Assumption
- Making Blind Evaluation of Polynomials Verifiable
- From Computations to Polynomials

▷ Homomorphic Hiding (HH) are essentially same as blind signature schemes which are based on discrete-log problem

▷ An HH E(x) of a number x is a function which satisfies:

- ○ Given E(x), it is hard to find x
- ○ If x != y, then, E(x)!=E(y)
- ○ If someone knows E(x) and E(y), then one can compute E(x+y)

▷ Example: $E(x)=g^x$, for x in Z^*_p, a cyclic group with generator g and p being prime.

- ○ Given E(x), E(y), we can compute E(ax+by) as -
  $E(ax+by) = g^{(ax+by)} = g^{(ax)} . g^{(by)} = (E(x))^a . (E(y))^b$

Zcash

- Zero-Knowledge Proofs
- zk-SNARK
- Homomorphic Hidings
- Blind Evaluation of Polynomials
- Knowledge of Coefficient Test and Assumption
- Making Blind Evaluation of Polynomials Verifiable
- From Computations to Polynomials

- ▷ Polynomial P of degree d over F_p :
  $P(x) = a_0 + a_1 . x + a_2 . x^2 + ... + a_d . x^d$
- ▷ Let Alice have a polynomial P of degree d and Bob have point s in F_p that he chooses randomly. Bob wishes to learn E(P(s))
- ▷ Two Simple ways:
  - ○ Alice sends P to Bob, and he computes E(P(s)) by himself
  - ○ Bob sends s to Alice, she computes E(P(s)) and sends it to Bob
- ▷ But, in Blind Evaluation Problem, we want Bob to learn *E*(*P*(*s*)) without learning *P* - which precludes the first option; and, most importantly, we don't want Alice to learn *s*, which rules out the second.
- ▷ We can do it as follows:
  - ○ Bob sends to Alice the hidings *E(1), E(s), ..., E(*s^d*)*
  - ○ Alice computes *E(P(S))* from the elements she received and sends *E(P(S))* to Bob

Zcash

- Zero-Knowledge Proofs
- zk-SNARK
- Homomorphic Hidings
- Blind Evaluation of Polynomials
- Knowledge of Coefficient Test and Assumption
- Making Blind Evaluation of Polynomials Verifiable
- From Computations to Polynomials

- We saw in the above protocol that Alice is *able* to compute E(P(S)), but it does not mean she will indeed send E(P(S)) to Bob. She can send some other value as well.
- Hence we need a way to force Alice to follow the protocol. We achieve this by Knowledge of Coefficient (KC) Test.
- The KC Test
  - For q in F^*_p, we call a pair (a,b) a q-pair if a,b!=0 and b=q.a
  - The test is:
    - Bob chooses a random q in F^*_p and a in Group G and computes b=q.a
    - Bob sends (a,b) the challenge pair to Alice.
    - Alice must respond with a different (a’,b’) that is also q-pair
    - Bob accepts if (a’,b’) is q-pair
  - Alice can do this easily by choosing c in F^*_p and responding with (a’,b’) = (c.a,c.b)
- Knowledge of Coefficient Assumption (KCA): If Alice returns a valid (a’,b’), then she knows r s.t. a’=c.a

Zcash

- Zero-Knowledge Proofs
- zk-SNARK
- Homomorphic Hidings
- Blind Evaluation of Polynomials
- Knowledge of Coefficient Test and Assumption
- Making Blind Evaluation of Polynomials Verifiable
- From Computations to Polynomials

▷ Extended KCA:
  - ○ Suppose in the protocol, B sends multiple q-pairs to Alice - (a_1,b_1), ..., (a_d,b_d) and then Alice generates a different q-pair (a’,b’)
  - ○ It turns out now Alice has more ways to generate (a’,b’)
  - ○ Alice can take any linear combination of the given d pairs, ie, choose c_1, ..., c_d in F_p s.t.
    $a' = (c_1 . a_1 + c_2 . a_2 + ... + c_d . a_d)$
    $b' = (c_1 . b_1 + c_2 . b_2 + ... + c_d . b_d)$
  - ○ Extended KCA states that the above is the only way for Alice to generate new q-pair.

▷ Formally, suppose G is a group of size p with generator g written additively. The d-KCA in G is as follows:

▷ d-KCA: Suppose Bob chooses random q in F^*_p and s in F_p and sends q-pairs to Alice - (g , q.g), (s.g , qs.g), ..., $(s^d . g , q\, s^d . g)$. If Alice outputs q-pair (a’,b’), then Alice knows c_0, ..., c_d in F_p such that $c_0\, s^0 . g + ... + c_d\, s^d . g = a'$

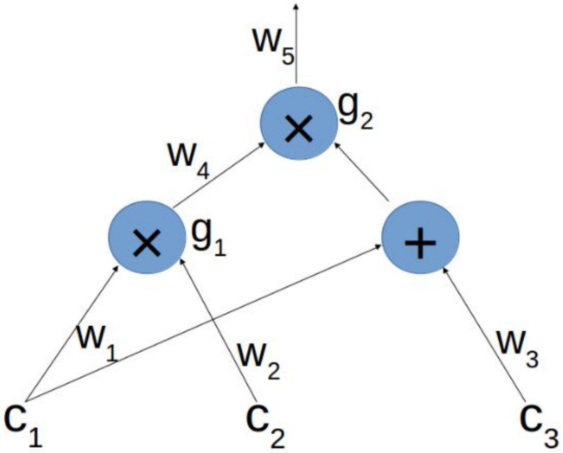
▷ Verifiable Blind Evaluation Protocol is exactly what is described above. Here polynomial P is $P(x) = c_0 . x^0 + ... + c_d . x^d$

Zcash

- Zero-Knowledge Proofs
- zk-SNARK
- Homomorphic Hidings
- Blind Evaluation of Polynomials
- Knowledge of Coefficient Test and Assumption
- Making Blind Evaluation of Polynomials Verifiable
- From Computations to Polynomials

▷ In summary, till now we have developed a protocol for verifiable blind evaluation of a polynomial. All that remains to show is how to translate statements we want to prove and verify to the language of polynomials.

▷ In 2013, Gennaro, Gentry, Parno and Raykova defined an extremely useful translation of computations into polynomials called a *Quadratic Arithmetic Program* (QAP). QAPs are the basis for modern zk-SNARK construction used in Zcash.

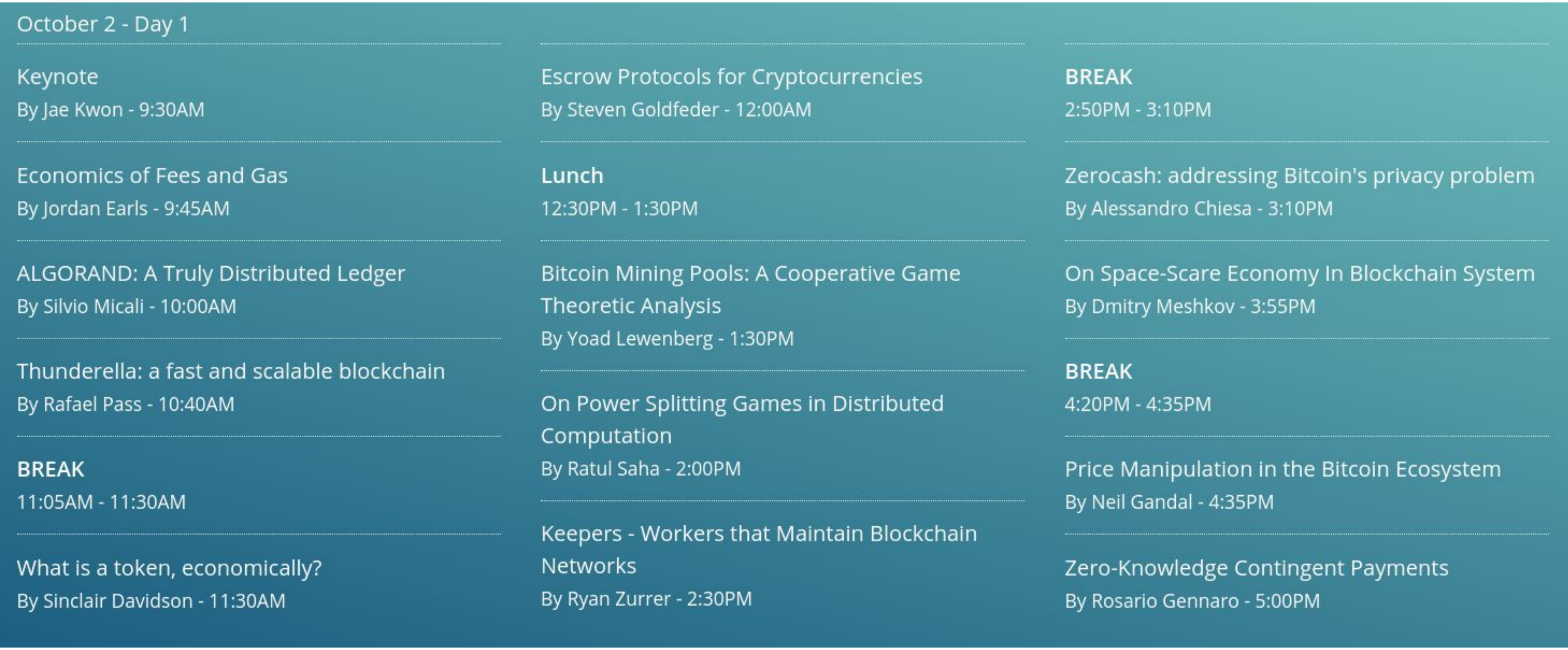


Source: https://z.cash/blog/snark-explain5.html

Conclusion

- Among more than 1000s of blockchains running publicly as on today, only few have tried all together new protocols. Most of them are variants of existing protocols, which are experimental approaches to solve the problems but we feel more focus needs to be on solving theoretical fundamental problems first.
- Consensus Protocols are the first most important step towards deploying blockchains in a practical scenario. Problems related to scalability, forks and regulation essentially boil down to the consensus protocol being used.
- One of the key ideas behind bitcoin was to provide anonymity, but with time people realized that it essentially is pseudonymous. Zero Knowledge Proofs have opened up the possibility of achieving anonymity as well. The idea of private blockchains was to provide anonymity and ZKPs have the potential of doing away with their need (though governments may not like this !) and take us a step closer to an ideal world of truly democratic and privacy ensuring blockchains.
- In our limited knowledge, till date no one has been able to come up with convincing enough proof of concept of the blockchain they are developing. This essentially deals with network evolution which we did not touch upon. Nonetheless, it is safe to say that even after resolving all the other problems, this will be an essential problem that needs to be solved for mass scale adoption of the technology.

Recent Developments - (CESC'17 at UCB) CryptoEconomics and Security Conference



October 2 - Day 1

Keynote
By Jae Kwon - 9:30AM

Economics of Fees and Gas
By Jordan Earls - 9:45AM

ALGORAND: A Truly Distributed Ledger
By Silvio Micali - 10:00AM

Thunderella: a fast and scalable blockchain
By Rafael Pass - 10:40AM

BREAK
11:05AM - 11:30AM

What is a token, economically?
By Sinclair Davidson - 11:30AM

Escrow Protocols for Cryptocurrencies
By Steven Goldfeder - 12:00AM

Lunch
12:30PM - 1:30PM

Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis
By Yoad Lewenberg - 1:30PM

On Power Splitting Games in Distributed Computation
By Ratul Saha - 2:00PM

Keepers - Workers that Maintain Blockchain Networks
By Ryan Zurrer - 2:30PM

BREAK
2:50PM - 3:10PM

Zerocash: addressing Bitcoin's privacy problem
By Alessandro Chiesa - 3:10PM

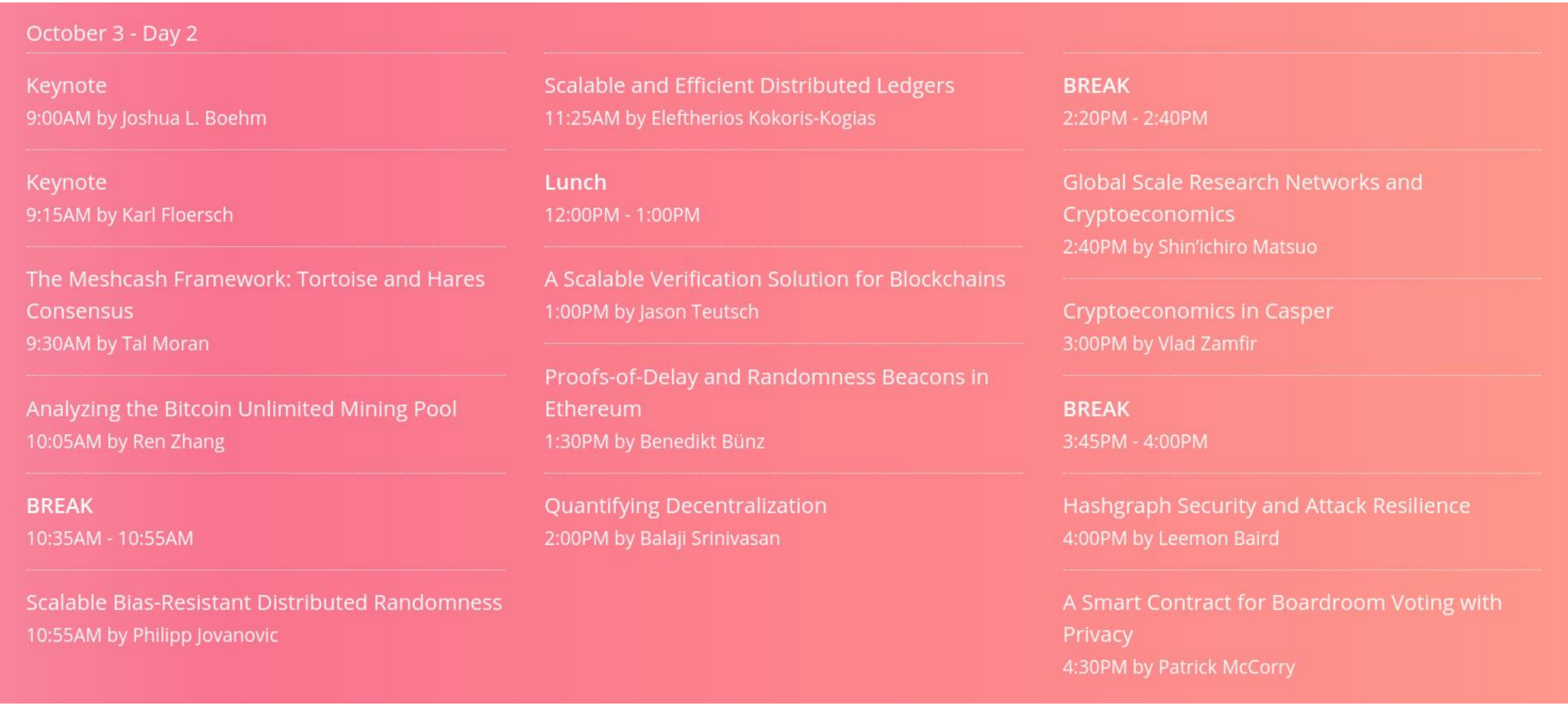
On Space-Scare Economy In Blockchain System
By Dmitry Meshkov - 3:55PM

BREAK
4:20PM - 4:35PM

Price Manipulation in the Bitcoin Ecosystem
By Neil Gandal - 4:35PM

Zero-Knowledge Contingent Payments
By Rosario Gennaro - 5:00PM

Recent Developments - (CESC'17 at UCB) CryptoEconomics and Security Conference



October 3 - Day 2

Keynote
9:00AM by Joshua L. Boehm

Keynote
9:15AM by Karl Floersch

The Meshcash Framework: Tortoise and Hares Consensus
9:30AM by Tal Moran

Analyzing the Bitcoin Unlimited Mining Pool
10:05AM by Ren Zhang

BREAK
10:35AM - 10:55AM

Scalable Bias-Resistant Distributed Randomness
10:55AM by Philipp Jovanovic

Scalable and Efficient Distributed Ledgers
11:25AM by Eleftherios Kokoris-Kogias

Lunch
12:00PM - 1:00PM

A Scalable Verification Solution for Blockchains
1:00PM by Jason Teutsch

Proofs-of-Delay and Randomness Beacons in Ethereum
1:30PM by Benedikt Bünz

Quantifying Decentralization
2:00PM by Balaji Srinivasan

BREAK
2:20PM - 2:40PM

Global Scale Research Networks and Cryptoeconomics
2:40PM by Shin'ichiro Matsuo

Cryptoeconomics in Casper
3:00PM by Vlad Zamfir

BREAK
3:45PM - 4:00PM

Hashgraph Security and Attack Resilience
4:00PM by Leemon Baird

A Smart Contract for Boardroom Voting with Privacy
4:30PM by Patrick McCorry

Thank You !
Questions ?

I THINK WE SHOULD BUILD A BLOCKCHAIN
UH-OH
E-mail: SCOTTADAMS@AOL.COM
S. Adams
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
© 1995 United Feature Syndicate, Inc.(NYC)
WHAT COLOR DO YOU WANT THAT BLOCKCHAIN?
I THINK MAUVE HAS THE MOST RAM.

Source : http://ericsammons.com/what-is-the-blockchain/

References

[1]: http://www.bankchain.org.in/
[2]: http://www.idrbt.ac.in/assets/publications/Best%20Practices/BCT.pdf
[3]: Dwork, C., Naor, M.: Pricing via processing or combatting junk mail
[4]: Dziembowski, Faust, Kolmogorov, Pietrzak: Proofs of Space
[5]: A. Miller, Y. Xia, K. Croman, E. Shi, D. Song. : The Honey Badger of BFT protocols
[6]: A. Kiayias, I. Konstantinou, A. Russell, B. David, R. Oliynykov. : Ouroboros: A provably secure proof-of-stake blockchain protocol.
[7]: E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, B. Ford. Enhancing Bitcoin security and performance with strong consistency via collective signing
[8]:R. Pass and E. Shi. Hybrid consensus: Efficient consensus in the permissionless model
[9]: Y Gilad, R Hemo, S Micali, G Vlachos, N Zeldovich : Algorand: Scaling Byzantine Agreements for Cryptocurrencies
[10]: https://z.cash

References

[11]: E Ben-Sasson, A Chiesa, D Genkin, E Tromer, M Virza : SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge
[12]: **https://z.cash/blog/snark-explain.html**
[13]: **https://z.cash/blog/snark-explain2.html**
[14]: **https://z.cash/blog/snark-explain3.html**
[15]: **https://z.cash/blog/snark-explain4.html**
[16]: **https://z.cash/blog/snark-explain5.html**

Declarations and Resources

- For a complete list of references please refer to the project report.
- Any use of copyrighted material if inadvertently has missing references, kindly be brought to the notice of the authors.