Indian Institute of Technology
Kanpur

CS498A

Undergraduate Project

Blockchain Technology

Author:
Nikhil Vanjani (14429)
Abhishek Verma (14026)

Advisor:
Prof. Arnab Bhattacharya
IIT Kanpur

November 23, 2017

Contents

1 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.

2 Background

2.1 History

Blockchains as a technology had its seeds sown long back in 1970s when Ralph Merkle designed the ***Merkle Tree*** data structure, which is at the core of blockchains. More specifically, distributed blockchains are immutable, append only, public ledgers. Subsequently, there were some attempts to build solutions using it but nothing much significant until Bitcoin came about in 2008.
Bitcoin is basically a digital currency, the transactions of which are recorded in a distributed ledger,

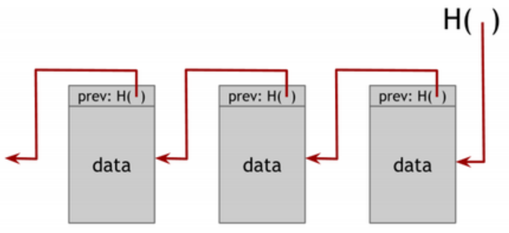


Figure 1: Blockchain data structure. Source : Coursera: Bitcoins and Cryptocurrencies

called the bitcoin blockchain. Before we go into the specifications of Bitcoin, it should be noted that this was not the first attempt of creating digital currency. There were other attempts as well, though not based on blockchains, the most prominent being David Chaum's idea of digital cash. Subsequently, PayPal emerged in 1998 which maybe called the last breakthrough before Bitcoin.
The primary reason behind Bitcoin's success as compared to all other attempts is that it provided a monetary incentive model for users in the network to validate transactions and also to approve only the correct transactions.

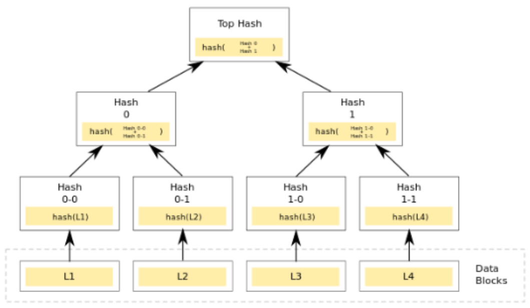


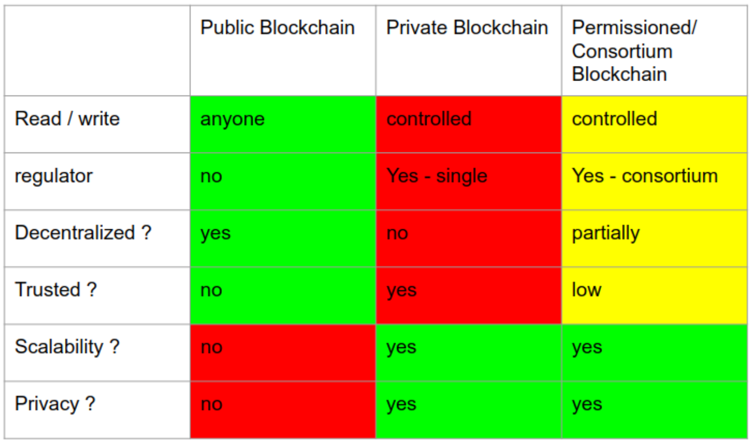
Figure 2: Merkle Tree. Source : By Azaghal - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=18157888

2.2 Use Cases

With the emergence of various applications of blockchains, we feel the use cases can be categorized as follows, though more cases may emerge as well:

- **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

2.3 Types of Blockchains



	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

Figure 3: Types of Blockchains

2.4 Major Challenges

In this section, we present our understanding of the major challenges. The list might not be complete as there are various aspects that we are yet to explore. Our study focused on Scalability, Forks and Privacy.

2.4.1 Scalability

The challenges associated with scalability seem to be three fold- speed, cost, space.

- **Speed:** At present, Bitcoin blockchain requires around 1 hour to arrive at distributed consensus which is impractical for scaling up various applications, like monetary transactions, uploading data records, etc. Other blockchains as well require some time, may be lesser than bitcoin but it still is impractical. Exchanges like Mt. Gox and Instawallet attempted to resolve this problem through Green Addresses but these systems relied on trusting the service provider that they won't attempt double spending attacks. But such exchanges could not stand the tide of time and their trust was eventually broken. So, the need is still of a trustless, ie, decentralised system to arrive at distributed consensus quickly. Silvio Micali, in a recent paper proposed Algorand which tries to solve this problem. We look forward to read this paper.
- **Cost:** In the model of blockchains, there are some entitites known as miners who validate transactions. Bitcoin is based on Proof of Work (PoW) concept, ie, miners need to guess a certain random string which will meet the conditions of the hash output. This guessing requires a lot of computational power and the model is such that the miner who guesses the fastest gets to mine the block. On an average whoever has the highest hash rate would be able to guess it. So, as of today, bitcoin blockchain consumes a lot of power due to this competition. The hash rate is increasing day by day and miners will only mine till the point of time when they are reaping profits out the business. To cut down on cost, there have been various alternate proposals like Proof of Stake, Proof of Capacity, Proof of Activity, etc.
- **Space:** There are two primary issues related to space and storage. Firstly, in blockchains like bitcoin, a block can have maximum size of 1MB and each transaction has a minimum size. Also, a block is mined in 10 minutes, and consequently one can carry out transactions only at the rate of 7 transactions per minute. The protocol implementation of micro-payments might be a viable solution for this problem, but micro-payments in their own essence are losing out practicality on bitcoin blockchain due to high transaction fees. People are coming up with alternate solutions to micro-payments. Secondly, as more and more people adapt the blockchains, more transactions get recorded on them and the total size of blockchains increase. For instance, the size of bitcoin blockchain is in excess of 100GBs, which makes it impossible for each user to store the complete blockchain. This was anticipated by the developers beforehand and hence there are two types of nodes in bitcoin- fully functioning nodes and non-fully functioning nodes. Fully functioning nodes store the complete blockchain and get to mine blocks as well, whereas non-fully functioning nodes can store only part of blockchain, say the blocks which contain their own transactions and they are unable to participate in mining activity as they don't store the pool of transactions yet to be mined. Researchers anticipate that due to the increasing size

of bitcoin blockchain, there are only thousands of fully functioning nodes on the network and that this figure might be gradually dropping which is a bad sign as it makes pooling majority of computational power in the network easier. Apart from mining issues, storage issues are common problem for scaling up any blockchain and people are yet to figure out how to do that using a blockchain. There are alternatives like MaidSafe, which implement decentralized consensus but blockchain data structure is not at its core. The consensus is achieved using something knwon as Close Groups.

2.4.2 Network Bootstrapping and Evolution

Bootstrapping the network is something which is done quite innovatively in most of the blockchains. What they do is to put an upper limit on the total number of cryptocurrencies ever in the network and the number of cryptocurrecnies usually is either constant from the beginning or its rate of generation decreases with time. This helps to prevent inflation as the demand 'might' keep increasing forever but the supply has an upper cap. To ensure that this 'might' happens, the early investors in the new blockchains keep themselves invested in it for long time as that is the only way to increase their profits. This investment of their helps to eastablish trust in the system and in turn boothstrap the network to grow. But putting an upper cap on total number of coins points to a challenge of how the system will evolve. To predict how it will evolve is challenging because the system is constantly changing. People are uncertain about what will happen when the mining rate becomes so low in bitcoin blockchain that the primary source of revenue for miners will be the transaction fees. Will the transaction fees be too steep ? Nowadays people are trying to understand and predict how such networks will evolve from a game theoretic perspective. The theory is catching up but is not yet up to the mark. This is one of the major hurdles the technology faces for widespread adaption.

2.4.3 Forks and Regulators

At times it so happens that a hacker exploits some vulnerability in a blockchain code or in a service built on top of a blockchain and the hacker is able to gain access to users' wallets and rob them. Such instances have fuelled philosophical debates in two aspects. Firstly, whether hard fork should be created or not. While the creation of hard fork may help to revert the robberies but it goes against the principles of blockchains, ie, it no longer remains immutable. Secondly, if someone thinks hard forks should be done, who should lead the process of spreading awareness among the network users to perform the hard fork in the blockchain to an earlier point of time ? Obviously, there will be some actors who will have enough influence on others so as to successfully propagate it into the majority of the network users. So in a way there still are some sort of regulators or regulatory body of the blockchain, which is principally against the idea of decentralization. These both are open debates and some blockchains happen to have created hard forks while some happen to have prevented that from happening and tried to find an alternate solution. Surely it would be impossible to avoid such occurences as there are always zero days out there. Nonetheless, it would be interesting to explore ways to decreases instances where such decisions need to be taken.

3 Indian Government's initiatives towards adapting blockchains

- BankChain[3]: It is a community of banks for exploring, building and implementing blockchain solutions.
  - It was 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: It is RBI's research Wing. It has recently published a whitepaper "Applications of Blockchain Technology to Banking and Financial Sector in India"[4].

4 Consensus Protocols

4.1 Detour

- Digital currencies are in existance since 1980's but they are not decentralized.
  - There is a need of 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

4.2 Proof of Work (PoW)

- PoW was introduced by Dwork and Naor[5] 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.
- 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.

4.3 Proof of Stake (PoS)

- 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.
- 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 too has issues. Attacker may split their credit among several users, so even when they are caught, they will be penalized less.

4.4 Proof of Space (PoSp)

- PoW: basic idea - dedicate non-trivial computational work for serving every request.
- PoSp[6]: 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

- 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

4.5 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[7] 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.
  - This is not decentralized.

- Hybrid consensus [8] [9] [10] 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.
  - Since this is based on PoW, forks are still possible.

5 Algorand : Scaling Byzantine Agreements for Crpyocurrencies [1]

5.1 The Big Picture

- 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.

5.2 Assumptions

- 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)

5.3 Procedure

- 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.
- 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.

5.4 Tentative Consensus

- 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.

5.5 Advantages

- 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.

5.6 BA* protocol

- 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.

5.7 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.

6 Zcash : a zero-knowledge blockchain[2]

6.1 Zero-Knowledge Proofs

- 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

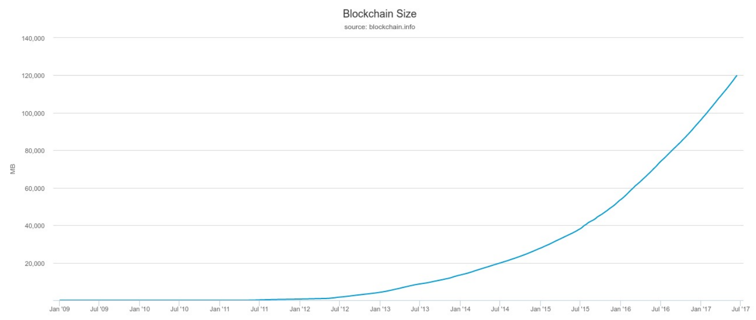


Figure 4: Trends in length of Bitcoin Blockchain. Source : https://www.newsinbit.com/bitcoin-blockchain-size-reaches-120gb-and-increasing/

6.2 zk-SNARK

- zk-SNARK stands for - ***z**ero **k**nowledge- **S**uccinct **N**on-interactive **AR**gument of **K**nowledge*
  In this subsection, we won't discuss the exact zk-SNARK algorithm but rather focus on its key aspects which are as follows. In the subsequent subsubsections, we explain them in detail. Parts of those subsubsections are taken from Zcash's blog on zk-SNARK which has been cited here.
  - 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]

6.2.1 Homomorphic Hidings

- 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 - $E(ax + by) = g(ax + by) = g^{(ax)}.g^{(by)} = (E(x))^a.(E(y))^b$

6.2.2 Blind Evaluation of Polynomials

- Polynomial P of degree d over $F_p : P(x) = a_0 + a_1.x + a_2.x_2 + \ldots + a_d.x_d$
- 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 do not want Alice to learn s, which rules out the second.
- What we can do is:
  - 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

6.2.3 Knowledge of Coefficient Test and Assumption

- 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
    - 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 $rs.t.a' = c.a$

6.2.4 Making Blind Evaluation of Polynomials Verifiable

- 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 $x(a', b')$
  - 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 c1, ..., cd in Fp s.t. $a' = (c_1.a_1 + c_2.a_2 + \ldots + c_d.a_d)$ and $b' = (c_1.b_1 + c_2.b_2 + \ldots + c_d.b_d)$
  - 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), ..., $(s_d.g, qs_d.g)$. If Alice outputs q-pair (a', b'), then Alice knows $c_0, \ldots, c_d$ in Fp such that $c_0 s^0.g + \ldots + c_d s^d.g = a'$
- Verifiable Blind Evaluation Protocol is exactly what is described above. Here polynomial P is P(x) $= c_0.x^0 + \ldots + c_d.x^d$

6.2.5 From Computations to Polynomial reduction

- 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.

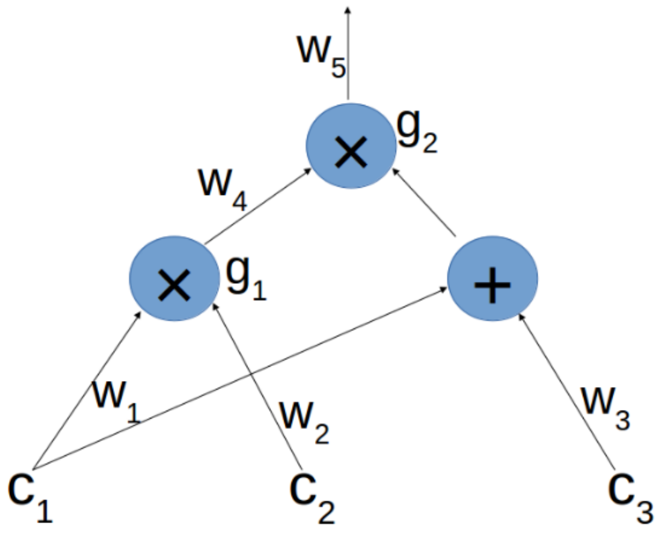


Figure 5: An Arithmetic Circuit. Source: https://z.cash/blog/snark-explain5.html

Explaining this reduction is too much technical and beyond scope of this report. Nonetheless, to see an example of it, Zcash's blog -[16] explains it well.

7 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.

References

[1] S Micali G Vlachos N Zeldovich Y Gilad, R Hemo. Algorand: Scaling byzantine agreements for cryptocurrencies.

[2] Zcash : https://z.cash.

[3] Bankchain : http://www.bankchain.org.in/.

[4] IDRBT. Applications of blockchain technology to banking and financial sector in india : http://www.idrbt.ac.in/assets/publications/best

[5] Naor Dwork. Pricing via processing or combatting junk mail.

[6] Kolmogorov Pietrzak Dziembowski, Faust. Proofs of space.

[7] K. Croman E. Shi D. Song. A. Miller, Y. Xia. The honey badger of bft protocols.

[8] A. Russell B. David R. Oliynykov A. Kiayias, I. Konstantinou. Ouroboros: A provably secure proof-of-stake blockchain protocol.

[9] N. Gailly I. Khoffi L. Gasser B. Ford E. Kokoris-Kogias, P. Jovanovic. Enhancing bitcoin security and performance with strong consistency via collective signing.

[10] E. Shi. R. Pass. Hybrid consensus: Efficient consensus in the permissionless model.

[11] D Genkin E Tromer M Virza E Ben-Sasson, A Chiesa. Snarks for c: Verifying program executions succinctly and in zero knowledge.

[12] Explaining snarks part i: Homomorphic hidings : https://z.cash/blog/snark-explain.html.

[13] Explaining snarks part ii: Blind evaluation of polynomials : https://z.cash/blog/snark-explain2.html.

[14] Explaining snarks part iii: The knowledge of coefficient test and assumption : https://z.cash/blog/snark-explain3.html.

[15] Explaining snarks part iv: How to make blind evaluation of polynomials verifiable : https://z.cash/blog/snark-explain4.html.

[16] Explaining snarks part v: From computations to polynomials : https://z.cash/blog/snark-explain5.html.