SoK of Used Cryptography in Blockchain

Mayank Raikwar[*]   Danilo Gligoroski[*]   Katina Kralevska[*]

June 20, 2019

Abstract

The underlying fundaments of blockchain are cryptography and cryptographic concepts that provide reliable and secure decentralized solutions. Although many recent papers study the use-cases of blockchain in different industrial areas, such as finance, health care, legal relations, IoT, information security, and consensus building systems, only few studies scrutinize the cryptographic concepts used in blockchain. To the best of our knowledge, there is no Systematization of Knowledge (SoK) that gives a complete picture of the existing cryptographic concepts which have been deployed or have the potential to be deployed in blockchain. In this paper, we thoroughly review and systematize all cryptographic concepts which are already used in blockchain. Additionally, we give a list of cryptographic concepts which have not yet been applied but have big potentials to improve the current blockchain solutions. We also include possible instantiations (references) of these cryptographic concepts in the blockchain domain. Last but not least, we explicitly postulate 18 challenging problems that cryptographers interested in blockchain can work on.

[*]Department of Information Security and Communication Technologies, Norwegian University of Science and Technology - NTNU

Contents

7   Conclusion   35

1 Introduction

Blockchain, a distributed ledger managed by a peer-to-peer network collectively adhering to some consensus protocol, is arguably considered as a new and disruptive technology. Both academia and industry are profoundly affected by new solutions to some old problems which are based on this new technology. The success of the blockchain concept is ultimately connected with the financial success of Bitcoin [1] that was developed just one decade ago, and the subsequent avalanche of more than 2140 other crypto-currencies that all together built a financial market worth around $285 billion (as of 16 June 2019) [2].

We can trace the origins of the ideas to use cryptography for secure and private transactions for paying access to databases, paying for services such as online games, transferring money over the Internet, Internet shopping and other commercial activities back in 1990's with David Chaum's eCash system [3]. One of the negative aspects of eCash was that it was a centralized system, controlled by a trusted third party. Another hurdle for a broader acceptance of eCash was the fact that it was covered by a long list of patented algorithms – something that is considered as a big obstacle to acceptance among the crypto community.

In parallel, in 1990's we saw the development of several cryptographic ideas not directly connected but somehow still related to the ideas of using cryptography in financial transactions. We mention some of them such as the proposal on how to combat junk email [4] by Dwork and Naor that was published in 1992, and which used computationally expensive functions. Then in 1996, there was a proposal for time-lock cryptographic puzzles [5] by Rivest, Shamir, and Wagner by using RSA based CPU expensive computations. At the end of 90's and early 2000's several patent free cryptographic concepts were proposed, implemented and released as open source projects by an online movement and a community of cryptographers and programmers known as "Cypherpunks" [6]. Those cryptographic concepts and implementations include Adam Back's "hashcash" proposal for a currency based on the hardness of finding partial hash collisions [7], Wei Dai's "b-money" [8] and Nick Szabo's[1] "Bitgold" proposal [9]. These concepts have been the basis of the Satoshi Nakamoto's decentralized cryptocurrency, nowadays known as Bitcoin [1, 10]. As a recognition of their pioneering activities in the decentralized cryptocurrencies, Ethereum [11] – the second most popular cryptocurrency – named the three of its denominations as "Wei", "Szabo" and "Finney" [12].[2]

The underlying core technology in Bitcoin is blockchain. Blockchain is a distributed ledger maintaining a continuously growing list of data records that are confirmed by all of the participating nodes. The data is recorded in this public ledger in a form of blocks of valid transactions, and this public ledger is shared and available to all nodes.

Blockchain is envisioned as a promising and powerful technology but it still encounters many research challenges. Some of the main challenges are constant improvement of its security and privacy, key management, scalability, analysis of new attacks, smart contract management, and incremental introduction of new cryptographic features in existing blockchains. These challenges arise due to the network structure and the underlying consensus mechanisms and cryptographic schemes used within the blockchains. To overcome these challenges and to find enhanced solutions, many of the cryptographic concepts such as signature schemes, zero-knowledge proofs, and commitment protocols are scrutinized and applied. As cryptography is a vast research field,

---

[1]Nick Szabo was also part of the eCash development team in late 90's.

[2]Hal Finney was a cypherpunk and the receiver of the first Bitcoin transaction of 10 Bitcoins from the anonymous Satoshi Nakamoto [13].

there is always a scope to find new cryptographic schemes in order to improve the solutions in blockchain.

The majority of the ongoing research in Blockchain focuses on finding and identifying improvements to the current processes and routines, mostly in industries that rely on intermediaries, including banking, finance, real estate, insurance, legal system procedures, and healthcare. The study on business innovation through blockchain [14] presents some blockchain enabled business applications and their instantiations. These blockchain enabled applications still need a proper way for selecting the cryptographic technique employed in their respective solution in order to meet the business requirements. Not only these blockchain applications but also the research community will benefit from an overview in a form of systematization of the current state of knowledge of all available cryptographic concepts which have been applied or can be applied in existing and future blockchain solutions. To the best of our knowledge, this is the first systematization of knowledge that gives a complete picture of the existing cryptographic concepts related to blockchain. We have tried to depict most of the cryptographic concepts in the blockchain domain. Although there are various works about specific cryptographic concepts used in blockchain, there are only few works which merge all these atomic works and present them in a single paper. Most of the review and survey works such as [15, 16] discuss security, privacy, consensus or other challenges in blockchain. A recent work of Wang et al. [17] gives a comprehensive analysis of cryptographic primitives in blockchain. Their analysis presents the functionality and the usage of these primitives in blockchain. However, the analysis is based only on existing cryptocurrencies and it lacks many of the cryptographic protocols which are used in blockchain.

1.1 Our Contribution

In this study, we classify cryptographic concepts based on their use in blockchain. We have divided them into two categories: 1. Concepts which are well used in blockchain, and 2. Concepts which are promising but not yet implemented in blockchain. This classification does not have a clear boundary. We incorporate cryptographic concepts, in the class of promising concepts, which require further research and scrutiny in order to be deployed in blockchain. As a result, the following points are the main contributions of our Systematization of Knowledge (SoK) paper:

- We provide a description of cryptographic concepts which have been applied in the blockchain field. We also include instantiation of these concepts in blockchain.

- We provide a list of cryptographic concepts which are rarely used or have not been used in blockchain but they have the potential to be applied in this field. These concepts open many possible research directions and they can be examined in different blockchain applications.

- We identified 18 research challenges that we formulate as *Research Problem i*. Some of them are rephrased research challenges already published in the literature and some of them are newly formulated research problems.

In this study, we do not claim that we have exhausted all of the cryptographic concepts which are employed in blockchain, but we have tried to cover the concepts which we felt are propitious for the blockchain domain. We also describe each cryptographic concept along with

its associated properties and its instantiation in the blockchain field. Additionally, in order to give one unified presentation about blockchain, we give a brief explanation about:

- Enabling concepts of blockchain such as hash function, consensus protocol, network architecture.

- Layered architecture of blockchain and emphasis on some of the major challenges associated with blockchain.

2   Research Methodology

To perform a systematization of knowledge of the existing cryptographic concepts related to blockchain, we established and followed a methodology that we explain in this Section. Since the invention of Bitcoin, there has been a growing interest in blockchain from both academia and industry. The number of publications in the blockchain field has been rapidly increasing in recent years. Not all of these publications are research works; some of these works discuss different use-cases of blockchain. Therefore, to review these many papers in the blockchain field, we pursued a research methodology which defines the inclusion criteria, a search strategy to search for respective publications and a data collection mechanism to accumulate the relevant publications. The collected data is later processed based on inclusion and exclusion criteria. The publications which meet the inclusion criteria go through one final step of quality assessment. Once a publication passes the quality assessment, it is included in our systematization.

We use keyword search to make the first selection of potentially relevant scientific publications. For the keyword search, we typed keywords such as ≪ *cryptographic concept name* ≫≪ *in blockchain* ≫ or ≪ *use of* ≫≪ *cryptographic concept name* ≫≪ *in blockchain* ≫. We use Google Scholar as our primary source to search for the relevant literature, but as Google Scholar does not exhaust all of the available literature, we also searched in databases such as: 1) IACR eprint archive, 2) IEEE Xplore, 3) ACM Digital Library, 4) ScienceDirect, and 5) Springer Link.

The inclusion criteria for this study is based on the following questions:

- Is the elaborated cryptographic concept useful in blockchain? The usefulness of the cryptographic concept is measured as whether we achieve some essential properties in blockchain by using the concept or whether the cryptographic concept can be beneficial for some use-case compared to an already implemented concept.

- Which properties can be achieved by using the cryptographic concept in blockchain?

- Is there any instantiation of the cryptographic concept in a blockchain study or application? If not, is there any potential?

The criteria for excluding a paper is:

- Informal literature discussing some cryptographic concepts in blockchain.

- Literature which claims on using a cryptographic concept but it does not give any guarantees about the feasibility and prospects of a potential implementation.

The quality of the papers that meet the inclusion criteria is assessed. For quality assessment, we apply the following questions:

- Is the cryptographic concept implemented in blockchain? If not, is it possible to implement it and will it be more efficient than the existing solution?

- Is there any security analysis or does the implemented concept rely on another underlying platform?

- Are the fundamental concept and its related properties adequately described?

3 Supporting and enabling concepts of Blockchain

As previously mentioned, blockchain is a way to encapsulate transactions in the form of blocks where blocks are linked through the cryptographic hash, hence forming a chain of blocks. Figure 1 shows the basic blockchain structure. Each block in the blockchain contains a block header and a representation of the transaction. For instance, in Figure 1, each block consists of its hash, the hash of the previous block, a timestamp and some other block fields (e.g., version, nonce). This depends from the block design. Merkle root hash represents the set of transactions in the Merkle tree, and this representation of transactions varies according to the design of the blockchain implementation. Figure 2 depicts the Bitcoin blockchain data structure showing in details the block format.
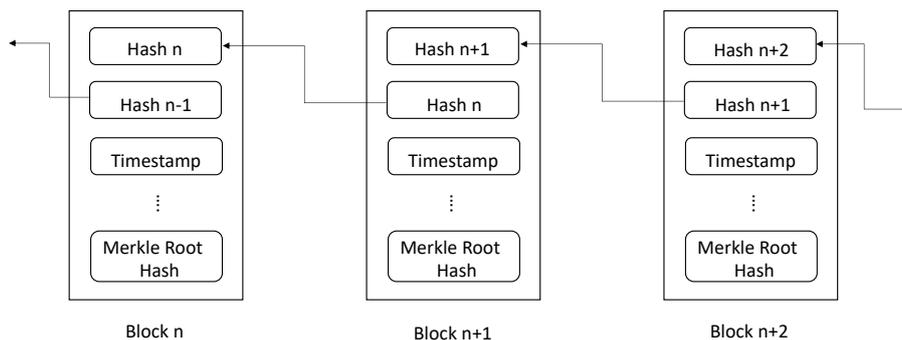

Figure 1: Basic blockchain structure.

Blockchain relies on different constituents which serve different purposes. In this Section, we give an overview of the main underlying concepts used to build a blockchain. A detailed technical explanation of all these concepts is out of the scope of this paper, but we have tried to cover the essentials of their functionality.

3.1 Cryptographic Hash Function

A hash function H is a function which takes an input of an arbitrary size and maps it to a fixed size output. Cryptographic hash functions have some additional properties such as: **a**) *collision resistance* - it is hard to find two inputs a and b such that $H(a) = H(b)$; **b**) *preimage resistance* - for a given output y it is hard to find an input a such that $H(a) = y$; and **c**) *second preimage resistance* - for a given input a and output $y = H(a)$ it is hard to find a second input b such that $H(b) = y$. Readers interested in an extensive cover of the field of cryptographic hash functions are referred to [18].
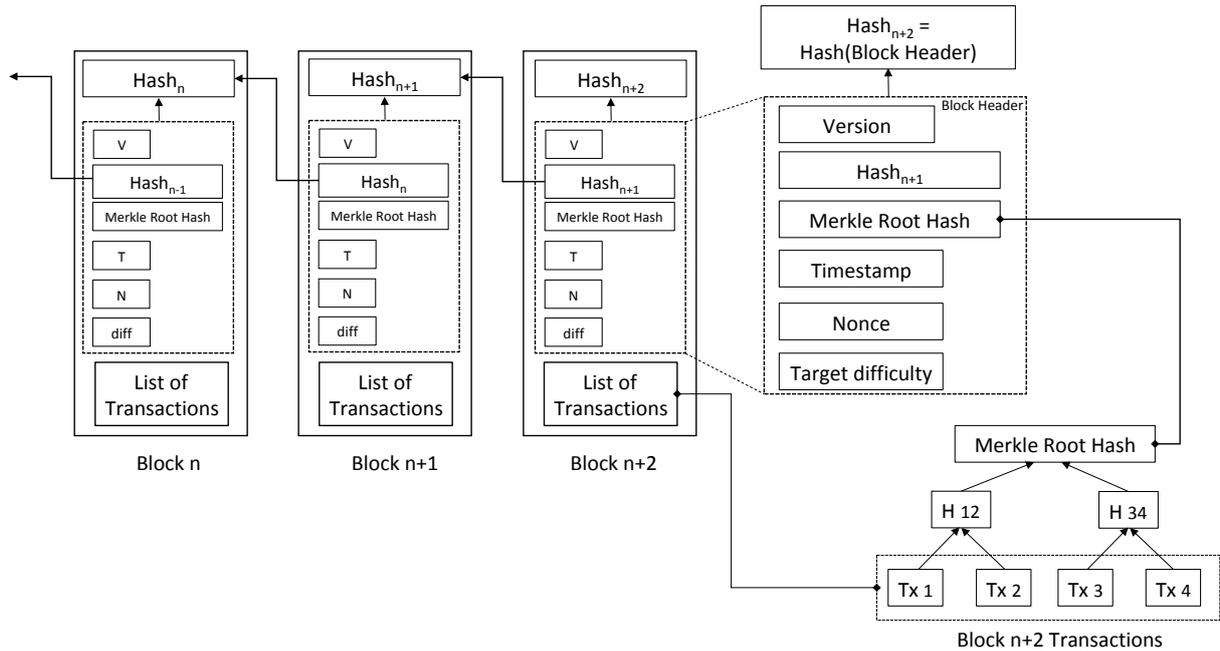
Figure 2: Blockchain data structure with block format.

Cryptographic hash functions in blockchain are used for various purposes such as:

1. solving cryptographic puzzles (the Proof of Work (PoW) in Bitcoin [1]);

2. address generation (for public and private keys);

3. shortening the size of the public addresses;

4. message digests in signatures.

The most popular cryptographic hash functions used in blockchains are SHA-2 [19] (especially the variant SHA256 - a variant that produces outputs of 256 bits), and some of the well analyzed hash functions from the NIST SHA-3 competition and standardization that went to the later stages of that process (final 5 proposals or some of the 14 proposals from the second phase [20]). Some of the existing blockchain designs such as IOTA constructed their own "homebrewed" cryptographic hash function called Curl-P, that was received very critically and negatively by the crypto community [21, 22].

A typical way how cryptographic hash functions are used in blockchain designs is in a form of a mode of operation, i.e., a combination of several invocations of a same or different hash functions. For example, in Bitcoin [1], SHA256 is used twice and that construction is called SHA256d, i.e.,

$$\text{SHA256d}(message) = \text{SHA256}(\text{SHA256}(message)). \qquad (1)$$

Mining is a process of creating a new block of transactions through solving a cryptographic puzzle, and the participant who solves the puzzle first is called a *miner of the block*. If we look at the Bitcoin PoW puzzle, we can see that a miner has to find a *Nonce* (similar to Hashcash

protocol [7] that we discuss in the next subsection) to create the next block in the blockchain. The puzzle looks like this:

$$\text{SHA256d}(Ver||HashPrevBlock||\ldots||Nonce) \leq T \tag{2}$$

where T is 256-bit target value.

Target Value T	Fraction of SHA256d outputs $\leq T$
$0\text{x}7 \underbrace{t_1 t_2 t_3 t_4 \ldots t_{62} t_{63}}_{63\,times}$	$\frac{1}{2}$
$0\text{x}0 \underbrace{t_1 t_2 t_3 t_4 \ldots t_{62} t_{63}}_{63\,times}$	$\frac{1}{16}$
$0\text{x} \underbrace{00 \ldots 00}_{16\,times} \underbrace{t_1 t_2 t_3 t_4 \ldots t_{47} t_{48}}_{48\,times}$	$\frac{1}{2^{64}}$

Table 1: Fraction of SHA256d outputs with respective target value.

Looking into the fraction of SHA256d outputs that are less than the target value T for different values of T in Table 1 helps us to understand why mining is hard in PoW. Namely, the probability of finding a nonce that will cause the whole block to have a hash that is less than the target value is

$$Pr[\text{SHA256d}(Block) \leq T] \approx \frac{T}{2^{256}}. \tag{3}$$

We next discuss the research and innovative activities in the area of cryptographic hash functions that were either remotely or directly connected and inspired by the trends in blockchain.

Several years after the launch of the Bitcoin and its source code being published as an open source on Github, blockchain designers started to clone and fork its basic code, and started to introduce different variants and innovations. One of the earliest forks from 2011 that is still popular nowadays is Litecoin [23]. The basic idea by the Litecoin design was to use a different hash function for its proof of work puzzles. The motivation came from the fact that even in 2011 there were trends to build specialized application-specific integrated circuit (ASIC) hardware implementations of SHA256d that will mine the blocks several orders of magnitude faster than ordinary CPUs and GPUs. Instead of SHA256d, Litecoin uses *Scrypt* [24] - a memory-intensive compilation of use of the HMAC [25] construction instantiated with SHA256 and use of the stream cipher Salsa20/8 [26]. The idea was that the use of *Scrypt* will be impractical to implement it in ASIC, thus, giving chances of individual owners of regular computers and GPUs to become a significant mining community. While with no doubts we can say that Litecoin is a very successful alternative cryptocurrency, we can for sure claim that its initial goal to be ASIC resistant blockchain design was not successful. Nowadays, you can find commercial products for Litecoin hardware mining[3].

Actually, we can say that the 10 years of history of blockchain, in general, and cryptocurrencies, in particular, is a history of failed attempts to construct a sustainable blockchain that will prevent the appearance of profitable ASIC miners that can mine the blocks with hash computing rates that are several orders of magnitude higher than the ordinary users of CPUs and

---

[3]One such a product that can compute 580 billion Scrypt hashes per second, is offered by the company Bitmain and is called "Antminer L3++". As of the time of writing this article, this product was advertised at `https://shop.bitmain.com/` for a price of $213.00 and for a 10 days delivery (2 June 2019).

GPUs. In that short history, we can mention Ethash used in Ethereum [11] for which there are now commercially available ASIC miners by at least two companies. In 2013, QuarkCoin [27] introduced the idea of using a chain of six hash functions (five SHA-3 finalists BLAKE, Grøstl, JH, Keccak and Skein [28]) and the second round hash function Blue Midnight Wish [29]. One of the motivations behind the QuarkCoin PoW function was to be more ASIC resistant than SHA256d. The cascading idea of QuarkCoin was later extended to a cascade of eleven hash functions in Darkcoin (later renamed DASH [30]). Needless to say, nowadays there are commercially available ASIC miners for X11 as well.

The frictions between ASIC miners and the cryptocurrency community seem to remain to the present days, and are somewhat evolving and inspiring novel proposals in blockchain protocols. The latest is the Programmatic Proof-of-Work (ProgPoW) initiative for Ethereum blockchain ecosystem that aims to make ASIC mining less efficient and to give some advantages to graphics processing units (GPU) mining [31].

3.2 Consensus Mechanisms

Consensus is the key component of blockchain to synchronize or update the ledger by reaching an agreement among the participants. In order to maintain the ledger in a decentralized way, many consensus mechanisms have been proposed. The first introduction of the use of a consensus mechanism in blockchain is implicitly given by Bitcoin. Bitcoin uses Proof of Work (PoW) mechanism as consensus where the idea came from *Hashcash Protocol* [7]. The objective of Hashcash was to prevent spam in public databases. The *Hashcash Protocol* is as follows. Suppose an email client wants to send an email to an email server. In the beginning, the client and the server both agree on a cryptographic hash function H which maps an input string to an n length output string. Then, the email server sends a challenge string c to the client. Now the client has to find a string x such that $H(c||x)$ starts with k zeros. Since H has pseudorandom outputs, the probability of success in a single trial is

$$\frac{2^{n-k}}{2^n} = \frac{1}{2^k}.$$

Here x corresponding to c is considered as PoW and the process of finding that x is called mining. PoW is difficult to generate but easy to verify.

Many literature studies on consensus mechanisms, for instance, the survey by Wang et al. [16] and "SoK: Consensus in the age of blockchains" [32], have been carried out in the past few years. Since consensus mechanisms have already been thoroughly studied in the literature, in this paper, we present the basic idea about how consensus mechanisms work and their classification.

In a consensus protocol, depending on the network architecture and blockchain type, some or all of the participants take part and maintain the ledger by adding a block consisting of transactions to their ledger. However, the creation of a new block to be added to the ledger is performed by a participant who is known as a leader of the consensus protocol in that particular execution. This leader is elected by different mechanisms of leader election process, and some of these mechanisms are given in Table 2.

After the leader is elected and the new block is created in order to achieve consensus or agreement on this block, two types of voting mechanisms are followed: *explicit* and *implicit*. In explicit voting, multiple rounds of voting occur and then based on the votes, consensus is reached. However, in implicit voting, the new block created by the leader is accepted by others who implicitly vote for the new block and add it to their ledgers. A leader election through

Leader Election Criteria	Reference Protocols
PoW Puzzle Competition	Bitcoin-NG [33], Casper [34], Proof of Stake velocity [35]
Verifiable Random Function	Tendermint [36], Algorand [37], Secure Proof of Stake [38]
Trusted Random Function	Proof of luck [39], Proof of elapsed time [40]
Modified Preimage Search	Snow White [41]
Sub-network of Masternodes	Darkcoin and DASH [42]

Table 2: Leader election in consensus protocols.

PoW puzzle competition (e.g., PoW puzzle 2 in Bitcoin) followed by an implicit voting to reach an agreement is also called "Nakamoto Consensus".

Consensus mechanisms also determine the performance of the blockchain network in terms of consensus finality, throughput, scalability, and robustness against various attacks. In some manner, consensus orchestrates the state of the programs executed in the blockchain network nodes by providing a runtime environment to collectively verify the same program and hence reach to a finality. There is no exact classification of consensus mechanisms, but in general they can be classified as consensus protocols with proof of concept and consensus protocols with byzantine fault-tolerant replication. These consensus protocols can be chosen based on the blockchain network and type. Most of the proof of concept consensus protocols are used in permissionless blockchains. There are many proof of concept schemes which have been proposed and implemented, e.g., Proof of Work (PoW) [43], Proof of Stake (PoS) [44], Equihash [45], having Masternodes in Dash [42], etc. As described in Section 3.1, in PoW puzzle based consensus protocols, miners try to solve the cryptographic puzzle by mining and these miners are also responsible for verification of the transactions, and an incentive (*reward*) is given to the first miner who solves the puzzle.

In case of a permissionless network, as there is no authentication and no proper synchronization, the underlying consensus algorithm should be able to handle the synchronization problem, scale well and mitigate different attacks in order to maintain canonical blockchain state in P2P network. To solve this synchronization issue, most of the blockchains use "Longest chain rule" to have a consistent canonical state of blockchain in this P2P blockchain network. On the contrary, in the permissioned blockchain, as there are restrictions and privileges associated with the peers, there is a strict control on the synchronization among the peers. Byzantine fault-tolerant protocols are usually adopted in permissioned blockchains to provide consensus properties such as validity, agreement, and termination. Practical Byzantine Fault Tolerant (PBFT) [46], Proof of Elapsed Time [40], Ripple consensus [47] are some of the consensus protocols used in permissioned blockchains.

Mining, Pool Mining and Incentive Mechanisms

In Proof of Work based blockchains, the addition of new transactions in the blockchain is performed by the mining process. In the Bitcoin mining process, a puzzle is solved by computing many hashes repeatedly (Equation 2) by putting different values for the nonce to satisfy the condition. When a miner successfully solves the puzzle first among all of the miners, it gets
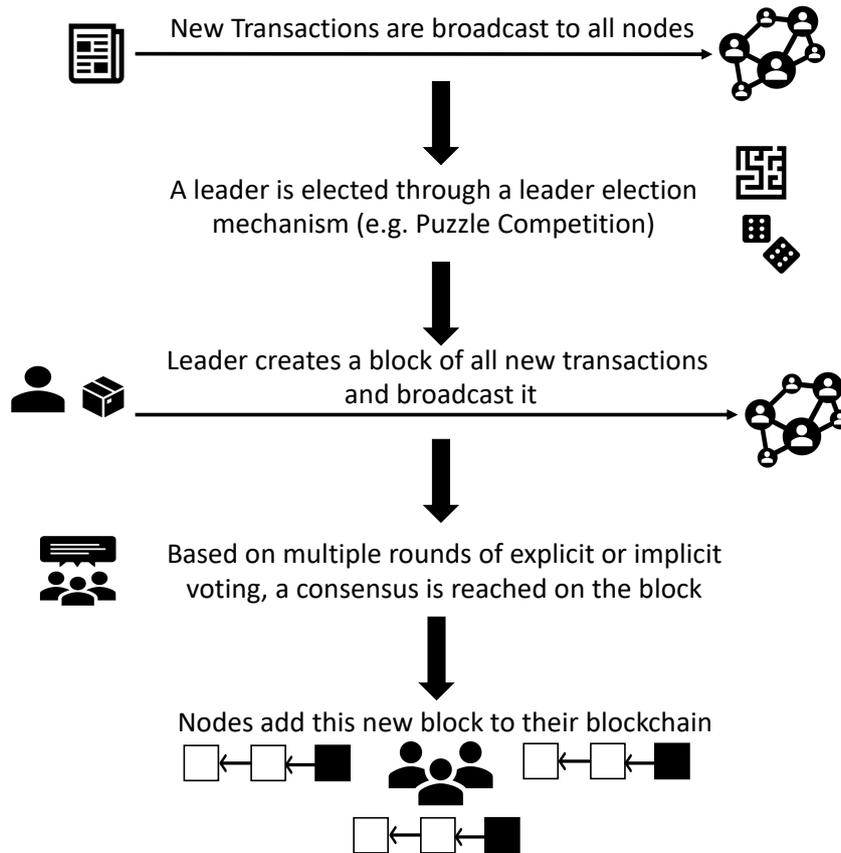
Figure 3: Blockchain consensus scenario.

a monetary incentive for solving the puzzle. Because of this incentive process, all consensus nodes or miners follow the rules of the blockchain state transition during the puzzle competition. Mining is a resource-intensive process where the main resources are computational power and memory. Mining can be performed either by a solo miner or by a group of miners, called a mining pool, who collectively try to solve the puzzle. Mining pools may operate on different mining techniques and incentive mechanisms. These incentive mechanisms can vary based on the used mining technique or the decision of the pool operator. Reference [16] gives a brief idea about the mining strategy management in blockchain networks, while reference [48] provides a strategic study of mining through stochastic games. Different incentive mechanisms are proposed and tested in blockchains. Reference [49] analyzes Bitcoin pooled mining reward systems, and a reward system based on information propagation in blockchain network is presented in [50].

3.3 Network Infrastructure

Blockchain is maintained by a peer-to-peer (P2P) network. P2P network is an overlay network which is built on the top of the Internet. This P2P blockchain network can be modeled as structured, unstructured or hybrid based on several parameters such as the consensus mechanism and the type of blockchain. Regardless of the representation of the network, a blockchain network should quickly disseminate the newly generated block so that the global view of the blockchain

remains consistent. Consequently, a synchronization protocol is needed, but a routing protocol might or might not be needed. A traditional P2P network uses a routing protocol to route the information through multihop; however, in many blockchains (e.g., Bitcoin), routing is not required because a peer can get information through at most one hop, so no routing table is maintained.

Almost all cryptocurrencies and blockchains such as Bitcoin [1], Ethereum [11], Litecoin [23] use unstructured P2P network where the idea is to have equal privileges for all of the nodes and to create an egalitarian network. A P2P network can follow flat or hierarchical organization for building a random graph among the peers. This graph is not fully connected, but in order to receive all of the communication and to maintain the ledger, each peer maintains a list of peer addresses. Thus, if any peer propagates a message in the network, eventually all peers receive it through their available connections. In an unstructured network, techniques like flooding and random walk are used to make new connections with the peers. In the unstructured network, peers can leave and join at any time. This can be exploited by an adversary that can join and see the messages floating in the network and can further do source spoofing, reordering or injecting of messages.

Blockchain can also use structured P2P network where nodes are organized in a specific topology and thus finding any resource/information becomes easier. In this structured P2P network, an identifier is assigned to each node to route the messages in a more accessible way. Each node also maintains a routing table. A structured P2P network maintains a distributed hash table (DHT) where (key, value) pairs are stored corresponding to the peers which help in the resource discovery. Ethereum has started the adoption of structured P2P network by using Kademlia protocol [51]. However, most of the blockchain networks are unstructured, and moreover, if the blockchain is public where no restriction to join or leave the network is enforced, then many possible attacks can happen. Thus, the security of blockchain depends heavily on the network architecture. A propagation delay or a synchronization problem in a P2P network can affect the consensus protocol of blockchain, leading to a non-consistent global view in blockchain. In addition to these problems, an adversary can cause several attacks in a P2P network, where few of the main attacks are as following:

- Netsplit (Eclipse) attack: An adversary monopolizes all of the connections of a node and splits that node from the entire network. Further, the node cannot participate in consensus or validation protocol and this causes inconsistency in the network [52].

- Routing attack: A set of participants are isolated from the blockchain network by the adversary and thus the block propagation is delayed in the network [53].

- Distributed Denial-Of-Service (DDOS) attack: An adversary exhausts the network resources and targets honest nodes so that honest nodes do not get the services or information which they are supposed to receive [54, 55].

3.4   Types of Blockchain

Blockchains can be classified depending on the implementation design, administration rules, data availability, and access privileges. From an academic point of view, they have been classified as "public" and "private". While from the administrative point of view, they are described

Blockchain Type	Application Domain	Anonymity	Scalability	Challenges	References
Permissionless Public	Decentralized P2P Networks	High	Low	Privacy, Scalability	Bitcoin [1], Zerocash [56], Monero [57]
Permissioned Public	Decentralized Organizations	High	Moderate	Privacy, Centralization	Ripple [58], EOS [59]
Permissionless Private	Intra-Organization Networks	Moderate	Moderate	Consensus, Scalability	LTO [60]
Permissioned Private	Organizational restricted ledgers	Low	High	Consensus, Centralization	Hyperledger fabric [61], Monax [62], Multichain [63]

Table 3: Blockchain classification.

as "permissioned" and "permissionless". Nevertheless, these terms are used interchangeably in most of the blockchain studies and applications in industries, which is not the correct way to use these terms. Even though the classification of blockchains is not very clearly specified in the literature, we can still classify blockchains by coupling public, private, permissioned and permissionless.

1. *Permissionless Public:* In this type of blockchain, anyone can join or leave the network at any time and participate in consensus as well to maintain the ledger. Everyone also has read and write access to the blockchain. Thus, it provides minimum trust among the participants, but it still achieves maximum transparency. Most of the cryptocurrencies and blockchain platforms are permissionless public, e.g., Bitcoin [1], Zerocash [56] and Monero [57].

2. *Permissioned Public:* This type of blockchain allows everyone to read the blockchain state and data, but in order to write the data and take part in consensus, there are permissions/privileges associated with the participants provided by the network administrator which in a certain way makes the system not fully decentralized. In this type of blockchain once a participant has some privileges, based on that it can become a validator as well. Ripple [58] and EOS [59] are permissioned public blockchain.

3. *Permissionless Private:* This type of a blockchain allows organizations to collaborate without the need of sharing information publicly. Being permissionless, allows anyone to join or leave the blockchain at any time, which is also acknowledged by other nodes as well. The smart contracts on these networks also define who is allowed to read the contract and the related data, not only just who is allowed to perform the actions. Some permissionless private blockchains use Federated byzantine agreement as a consensus protocol. LTO [60] network is an example of a permissionless private blockchain which creates "live contract" on the network.

4. *Permissioned Private:* These blockchains are mostly used in organizations where data/ information is stored in the blockchain with permissioned access control by members of the organization. The membership in the network is provided by the network administrator or some membership authority. Read and write access to the data is also provided by the network administrator. Hyperledger fabric [61], Monax [62], Multichain [63] are examples of permissioned private blockchains.

Table 3 proffers a clear picture of the classification of blockchains with associated advantages, challenges and application domains. However, in general, permissionless public blockchains are

commonly referred to as public blockchains and permissioned private blockchains are referred to as fully private blockchains. A combination of permissioned public and permissionless private makes "consortium blockchain" which is also called a federated blockchain. A consortium blockchain is neither completely public nor completely private, and it makes blockchain as partially decentralized. In consortium blockchain, the consensus is reached by a selected group of participants. Nowadays most of the organizations have embraced consortium blockchains for their blockchain enabled solutions.

4   Challenges in Blockchain

Blockchain as an emerging technology comes with many challenges. In order to solve these challenges, various solutions have been proposed and implemented in the blockchain. The proliferation of cryptocurrencies across multiple payment systems brings many risks in social, economic and technical terms. Blockchain encounters many challenges due to network architecture, underlying consensus protocol and applied cryptographic primitives. Some of these major challenges are security and privacy associated with blockchain, scalability of blockchain, and resource consumption (computational power, memory, network bandwidth).

We can summarize our discussion in Section 3.2, in a form of generic research problems and research challenges in the area of blockchain consensus mechanisms as follows. Construct a new blockchain consensus mechanism that is better than the existing ones from the following perspectives:

1. Less energy consumption;

2. More efficient consensus achievements;

3. Better security than the existing consensus mechanisms.

However, further in the paper when we identify a more concrete and focused research challenge, we formulate it in a form of a Research Problem. For example, from the discussion given in the 3.1 we can formulate the following:

Research Problem 1 *Construct sustainable blockchain systems that have one of the following properties:*

1. *They are provably resistant to give mining advantages to ASIC miners as opposite to GPU and CPU miners;*

2. *They are provably resistant to give mining advantages to ASIC and GPU miners as opposite to CPU miners.*

If we observe the blockchain as a layered architecture, we can identify the challenges that occur in each layer. Table 4 shows blockchain as a stack of five layers. These five layers serve the following purposes:

- **Smart contract layer** processes contract data and send the result data to the transaction layer.

- **Transaction layer** creates the transactions and sends those to consensus layer.

	Confidentiality	Integrity	Availability	Data Privacy	Anonymity
Smart Contract	Encryption	MAC	–	Data Privacy Preserving Computation	Identity Privacy Preserving Computation
Transaction	–	Signature Scheme	Access Structure of Transactions	Zero-Knowledge Proofs, Mixing Techniques	Zero-Knowledge Proofs
Consensus	–		Consensus	Access Control	Blind or Ring Signature
Network	Encryption	MAC	Protocols e.g. Gossip	–	IP Anonymity e.g. TOR
Database			Access Control	Access Control	–

Table 4: Layered architecture of blockchain.

- **Consensus layer** runs the consensus algorithm and adds the transactions to the block.

- **Network layer** deals with all P2P communication among blockchain nodes.

- **Database layer** stores the blockchain data in a respective database used by respective blockchain platform.

Table 4 gives a glimpse of blockchain layered architecture and also mentions some of the cryptographic techniques to achieve properties like security and privacy. In Table 4, the first column defines the layers of blockchain, and the first row illustrates the properties which can be accomplished in the different layer using different cryptographic techniques. Thus to understand, each cell corresponds to the deployed cryptographic method to attain the property in the corresponding column in the respective blockchain layer (corresponding row). For example, encryption can be used to achieve confidentiality in smart contract layer, Message Authentication Code (MAC) can be used to achieve integrity in the network layer of blockchain. Table 4 names few of the techniques used in the blockchain but there are more available cryptographic techniques which can be employed in blockchain. "–" in Table 4 represents that the corresponding property for the corresponding layer does not make much sense. Some of the significant challenges of blockchain are as follows.

4.1 Security and Privacy

For any blockchain, a key evaluation parameter is how well the security and privacy conditions meet the requirement of the blockchain. Analyzing the security and privacy issues of blockchain is a broad research area, and some studies have been conducted in this area. Here we do not cover those details, instead we only define these terms. Security is defined as three components: confidentiality, integrity, and availability. In a generic context, (i) confidentiality is a set of rules that limits access to information, (ii) integrity is the assurance that the information is trustworthy and accurate, and (iii) availability is a guarantee of reliable access to the information by authorized people. However, in case of blockchain, the term *Information* used in the above context can have multiple meanings such as data in the database, smart contract data or transactions. Privacy can be defined as data privacy and user privacy (anonymity). Table 4 includes some cryptographic mechanisms for achieving security and privacy of information subjected to different blockchain layers.

4.2 Scalability Issues

The size[4] of blockchain is continuously growing, and scalability is becoming a big problem in the blockchain domain. Scalability depends on the underlying consensus, network synchronization and architecture. To scale the blockchain, the computational power and the bandwidth capabilities should be high for each node in the blockchain, which is practically infeasible. Most of the current blockchains grant limited scalability.

One proposal how to address the scalability problems of the blockchain ledger is so called: "SPV, Simplified Payment Verification" [64]. It verifies if particular transactions are valid but without downloading the entire ledger. This method is used by some wallet and lightweight Bitcoin clients, and its security was first analyzed in [65].

Another issue in connection with the scalability is the issue of the interoperability. Namely, it is a fact that the number of different public ledgers is increasing rapidly. While some sort of a rudimentary interoperability has been implemented in cryptocurrencies exchange platforms [66], the risks and insecurities with these platforms are vast and well documented [67].

Research Problem 2 *Construct a new blockchain mechanism that periodically prunes its distributed ledger (reduces its size), producing a fresh but equivalent ledger, while provably keeping correct state of all assets that are subject of the ledger transactions.*

Research Problem 3 *Construct secure protocols for blockchain interoperability.*

4.3 Forking

A blockchain fork is essentially caused when two miners find a block at almost the same time due to a software update or versioning. In a blockchain network, each device or computer is considered as "a full node" which runs software to keep the blockchain secure by verifying the ledger. The software is updated to adjust some parameters and to install new features in the blockchain. This updated software may not be compatible with the old software. Consequently, the old nodes which have not updated their software and the new nodes which have performed a software update can cause a fork in the blockchain when they create new blocks. There are two types of forks: one which is not compatible with previous software version, called a hard fork, and another one which is compatible with the previous version (backward-compatible), called a soft fork. A hard fork happens when there is a significant change in the software such as change of block parameters or change of consensus mechanism. In the case of Ethereum, a hard fork will occur when it will migrate from Proof of Work to Proof of Stake. One example of a soft fork is Segregated Witness (SegWit) which was implemented in Bitcoin by changing the transaction format. Figure 4 depicts a blockchain forking scenario where the correct chain can be any of these two forked chains depending on the case of the hard or soft fork.

Research Problem 4 *Construct Forking-free consensus mechanism for permissionless public blockchain.*

---

[4] https://bitinfocharts.com gives most of the statistics (including size) of popular cryptocurrencies.
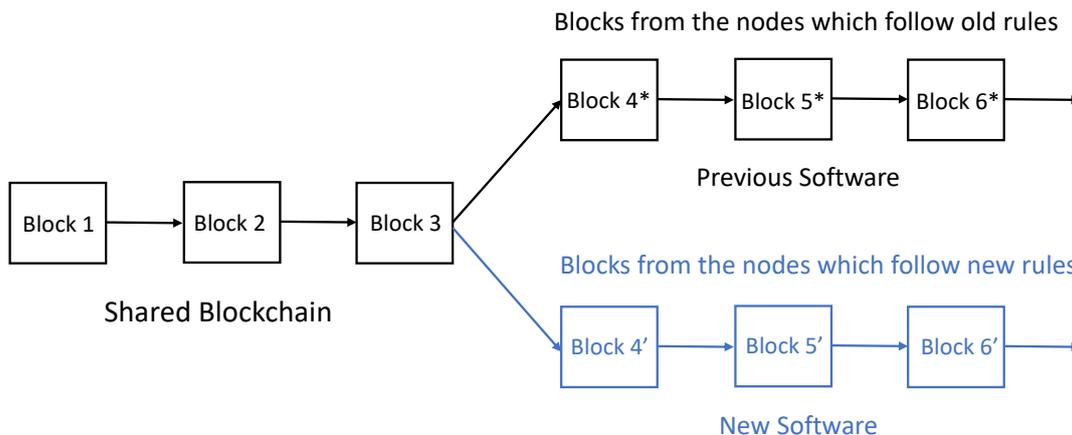
Figure 4: Blockchain forking.

4.4 Throughput

It is a measure of the number of blocks appended in blockchain per second which effectively means the number of transactions processed per second. Throughput depends on many factors such as the underlying consensus algorithm, the network structure, the node behavior, and the block parameters. The important factors affecting the throughput are the transaction size and design. Attaining high throughput is a bit hard in blockchain. However, to achieve high throughput, the size of the transaction can be reduced by excluding some information from the transaction. The throughput can be increased by increasing the block size and the bandwidth of the network till a certain level.

The number of transactions per second was recognized as a serious problem in Bitcoin network. While in the peak holiday period Visa and MasterCard can handle up to 50,000 transactions per second worldwide, the Bitcoin network can handle just 7 transactions. One proposal how to address this scalability issue is the "The Bitcoin Lightning Network" [68]. It is a network that handles instantly the Bitcoin transactions off the main ledger. It establishes a network of micropayment channels that addresses the malleability by using Bitcoin multi-signatures 2-of-2. Special nodes are needed for these micropayment networks and as of June 2019, there were around 4,500 nodes. The first financial transaction via the Lightning network was reported in January 2018. Litecoin decided to follow the Bitcoin Lightning network, and as of March 2019 there were more than 1000 registered nodes that handle the micropayments for that alternative cryptocurrency. Many other solutions were proposed to solve the scalability issue, similar to the Lightning off-chain computation and off-chain state channels, such as Sharding [69], Plasma [70], Liquid [71] and the recent Channel Factories [72].

As the Lightning network is gaining in popularity, new research challenges emerge as depicted in [73], and here we rephrase one of their research challenges:

Research Problem 5 ([73]) *Develop scalable protocols that will perform multi-hop payment-channel and path-based transactions with strong privacy guarantees even against an adversary that has network-level control.*

18

4.5 Energy Consumption

The mining process of blockchain (e.g., bitcoin mining) consumes a lot of energy. Most of the PoW puzzle based consensus protocols waste a huge amount of energy[5]. Many alternative consensus algorithms are introduced which use less energy than Bitcoin's PoW such as Proof of Stake [44], Equihash [45], and PBFT [46]. Energy is also consumed during communication over the network. Some cryptographic mechanisms also consume high energy so the selection of a proper cryptographic mechanism should be based not only on the memory requirement and the computational load but also on the amount of energy consumption. The use of blockchain should be energy efficient and to fulfill that, energy management techniques should be utilized in different use cases like in Internet-Of-Things (IoT).

4.6 Infrastructure Dependencies

The blockchain infrastructure is built with several elements of network protocols, cryptographic concepts, and mining hardware. All these elements depend on each other in some sense. If we look into the layered architecture of blockchain in Table 4, each layer is dependent on its upper and lower layers for some input/output. Thus, there are many infrastructure dependencies in blockchain. For instance, the data from the smart contract layer is an input to the transaction layer that outputs actual transactions; the data from the consensus layer results in an input to the network layer through a communication protocol; and the data from the network layer data is sent to the database through database storage management. These dependencies must be taken into account while building a comprehensive blockchain framework for any use case; otherwise, some of the blockchain functionalities will not be fulfilled.

From the blockchain infrastructure perspective, we have to mention here one evolving and enabling technology that will be very important in the next decade: 5G. 5G will connect hundreds of billions of IoT devices, but that vast number of devices can be governed securely only by strong decentralized mechanisms offered by the blockchain technologies [74, 75]. We formulate this debate as the following.

Research Problem 6 *Construct efficient, scalable, inexpensive and sustainable blockchain systems capable to handle and securely manage up to billions of IoT devices connected via the 5G network infrastructure.*

5 Overview of used cryptographic concepts in blockchain

From the cryptographic point of view, many of the cryptographic techniques have already been exhibited and heavily employed in various blockchain platforms and blockchain use-cases [17]. As the spectrum of the cryptographic concepts is vast, there is always a scope to dig out some of the existing cryptographic schemes and use them in blockchain services.

In Table 5 we give a comprehensive summary of all cryptographic concepts that we will cover in this and in the next Section. It serves as a handy overview and quick reference table for our systematization of the cryptographic knowledge used in blockchain.

Following are some of the cryptographic concepts which have already been well analyzed and implemented in blockchain.

---

[5]`https://digiconomist.net/bitcoin-energy-consumption` depicts Bitcoin energy consumption index charts in TWh per year. It also shows the energy consumption per country.

Cryptographic Concept	Properties	Instantiation (Reference)
Signature Scheme	Integrity and Authentication	In Every Blockchain e.g. Multichain [63],CryptoNote [76]
Zero-Knowledge Proof	User and Data privacy	Zerocoin [77], Zerocash [56]
Access Control	Data privacy	Hyperledger Fabric [61], FairAccess [78], [79]
Encryption Scheme	Confidentiality and Anonymity	Kadena [80], Hyperledger Fabric [61], Tendermint [36]
Secure Multiparty Computation	Privacy of Peers and Smart Contract	Enigma [81], Hawk [82]
Secret Sharing	Data privacy	SHARVOT [83]
Aggregate Signature	Fast Signature Verification	Tested in Bitcoin [84]
Identity Based Encryption	No Public Key Distribution Infrastructure	BAVP [85]
Commitment Scheme	Non-Repudiation	Used in Bullteproof [86] and in Monero [57]
Accumulator	Provides Membership Proofs, Anonymity	Batching Techniques for Accumulators in Blockchain [87]
Oblivious Transfer	Data Privacy	Searchain [88], [89]
Oblivious RAM	Confidentiality and Integrity	Solidus [90], [91]
Obfuscation	Privacy	Tested in Bitcoin [92]
Private Information Retrieval	Data Privacy	Private Blockchain Queries from PIR [73]
Proof of Retrievability	Cloud Data Recovery	Permacoin [93], Retricoin [94], Storj [95]
Decentralised Authorization	Data Privacy	BlendCAC [96], WAVE [97]
White-Box Cryptography	Data Privacy	Runtime Self-Protection in Blockchain Ledger [98]
Post-Quantum Cryptography	Quantum Resistant	Post-Quantum Blockchain [99], [100], [101]
Incremental Cryptography	Efficiency Improvement	Kadena [80]
Lightweight Cryptography	Fast, less Memory/Energy Consumption	LSB [102], EVCE [103]
Verifiable Random Function	Verifiable Pseudorandom Output	Algorand [37], Ouroboros Praos [104], Dfinity [105]
Verifiable Delay Function	Less Parallelism, Fast Verification	Chia Network [106]

Table 5: Summary of Cryptographic Concepts in Blockchain.

5.1 Signature Scheme

A standard digital signature is a mathematical scheme based on public-key cryptography that aims to produce short codes called signatures of digital messages by the use of a private key, and where those signatures are verifiable by the use of the corresponding public key. In this context, digital signatures guard against tampering and forgeries in digital messages.

A signature scheme is used in blockchain to sign the transaction, hence, authenticating the intended sender and providing transaction integrity as well as non-repudiation of the sender. Many of the signature schemes are widely accepted to employ integrity and anonymity in blockchain. The digital signature is one of the most important cryptographic primitives that makes blockchain to be publicly verifiable and with achievable consensus. Signature schemes are used in almost every blockchain. Blockchain applies different signature schemes to provide additional features like privacy, anonymity, and unlinkability. Some of the signature schemes applied in blockchain are:

1. Multi-Signature: In a multi-signature scheme, a group of users signs a single message. In a blockchain network, when a transaction requires a signature from a group of participants, it might be advantageous to use a multi-signature scheme. Blockchain platforms such as Openchain [107] and MultiChain [63] support $M-of-N$ multi-signature scheme which reduces the risk of theft by tolerating compromise of up to M-1 cryptographic keys. Boneh et al. also designed compact multi-signatures for smaller blockchains [108].

2. Blind Signature: In this scheme [109], signatures are employed in privacy-related protocols where the signer and the message authors (transaction in case of blockchain) are different parties. Blind signatures are used to provide unlinkability and anonymity of the transaction. In a blockchain setup, a blind signature might be helpful to provide anonymity and unlinkability where the transacting party and the signing party are different. Blind signatures have been used in BlindCoin [110] distributed mixing network to provide the

unlinkability of transactions. Blind signatures are also tested in Bitcoin to provide the anonymity for the Bitcoin on-chain and off-chain transactions [111].

3. Ring Signature: This scheme [112] uses a protocol where a signature is created on a message by any member of a group on behalf of the group while preserving the identity of the individual signer of the signature. Ring signatures are used to achieve anonymity of the signing party in the blockchain network. CryptoNote [76] technology uses a ring signature scheme to create untraceable payments in the cryptocurrencies. A trustless tumbling platform [113] also uses ring signature for anonymity.

4. Threshold Signature: This signature scheme is a (t, n) threshold signature where n parties receive a share of the secret key to create the signature and t out of n parties create a signature over any message. As the parties directly construct the signature from the shares, the key is never revealed in the entire scheme. Threshold signature can be helpful to provide anonymity in the blockchain network. CoinParty [114] uses a threshold signature scheme for multi-party mixing of Bitcoins.

While digital signatures produced with the keys used in Public Key Infrastructure (PKI) are well legally regulated and can be used in different types of legal disputes, it is a big challenge how to achieve similar regulations with all types of digital signatures used in the existing blockchain solutions. Additionally, in the physical world if an asset is stolen (for example an expensive car, or an expensive watch), it can be traced back to its legal owner.

Research Problem 7 *Develop security protocols that merge the existing standardized and legalized PKI systems with some of the the developed blockchain systems.*

Research Problem 8 *Design an anti-theft blockchain system, i.e., a system that guarantees a return of stolen assets back to their legitimate owners.*

5.2 Zero-Knowledge Proofs

In Zero-knowledge proofs [115], two parties, a prover and a verifier, participate. First, the prover asserts some statement and proves its validity to the verifier without revealing any other information except the statement. Thus, a zero-knowledge proof proves the statement as 'transfer of an asset is valid' without revealing anything about the asset. Zero-knowledge protocols are extremely useful cryptographic protocols for achieving secrecy in the applications. They can be used to provide the confidentiality of an asset (transaction data) in the blockchain while keeping the asset in the blockchain. Some of the public blockchains use zero-knowledge proofs such as Zerocoin [77] or Zerocash [56] for untraceable and unlinkable transactions. Zerocoin is a decentralized mix and extension to Bitcoin for providing anonymity and unlinkability of transactions by applying zero-knowledge proofs. In Zerocoin protocol, a user who has Bitcoins can generate an equal value of Zerocoins without the need of any third party mixing set. A user can spend his/her Bitcoin by 1) producing a secure commitment (i.e., Zerocoin), 2) recording it in the blockchain, and 3) broadcasting a transaction and a zero-knowledge proof for the respective Zerocoin. Hence, other users can validate the Zerocoin recorded in the blockchain and verify the transaction along with the proof. Here zero-knowledge proof protects the linking of Zerocoin to a user, yet Zerocoin is a costly protocol due to its high complexity and large proof size.
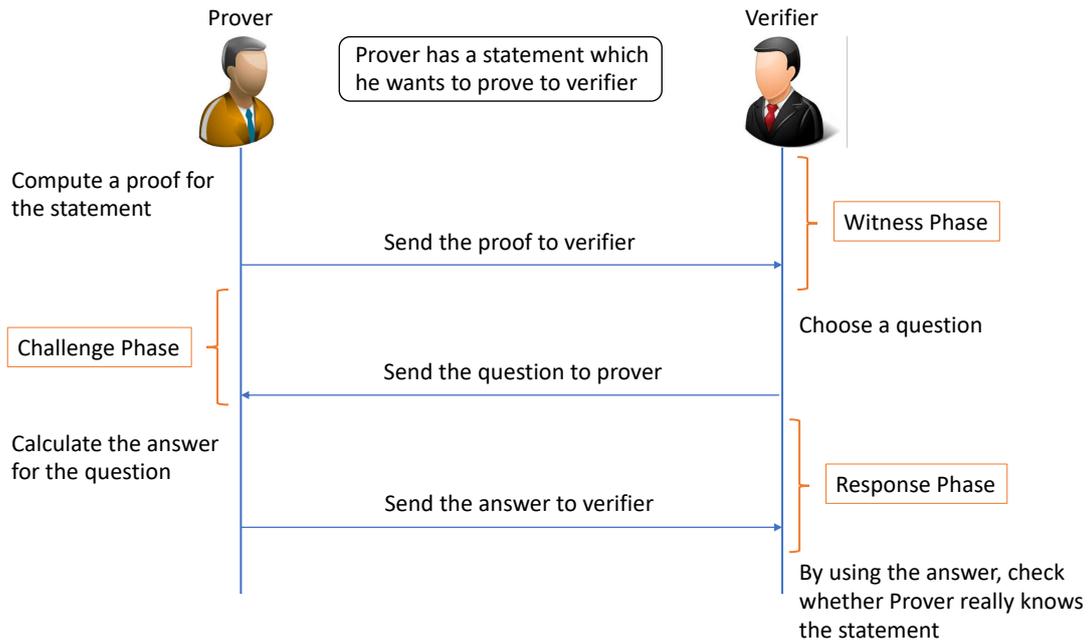
Figure 5: An interactive zero-knowledge protocol.

To reduce the complexity and the proof size, a variant of zero-knowledge proof known as Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) [116] is used by Zerocash protocol. zk-SNARK hides the information about the amount and the receiver address in a transaction. The main idea of zk-SNARK is 'any computational condition can be represented by an arithmetic circuit, which takes some data as input and gives *true* or *false* in response'. zk-SNARK reduces the proof size and the computational effort compared to the basic zero-knowledge proofs. An enterprise-focused version of Ethereum, Quorum blockchain platform [117] also uses zk-SNARK for transaction privacy and anonymity. Figure 5 illustrates an interactive protocol of zero-knowledge where the prover has a statement, and he/she wants to prove that the statement is correct without revealing any information related to the statement. In the interactive protocol, the verifier asks many questions related to the statement and the prover answers these questions in such a way where the prover proves the statement and does not reveal any necessary information.

5.3   Access Control

It is a selective restriction on information or resource based on some policy or criteria. These mechanisms [118] can be enforced to put a restriction or access in the blockchain. The access can be a read/write access or an access to participate in a blockchain protocol. There are many different access control mechanisms such as role-based, attribute-based, organizational-based access control which can be used in blockchain. Recent incidents show security breaches and data theft from certain blockchain platforms, which can be tackled and prevented by access control. The privacy of data can be ensured in blockchains by using access control [78, 79]. Nowadays, access control techniques are profoundly used in blockchain based medical applications [119] or blockchains for the insurance industry where the data is sensitive information that must be

accessible to only trusted and authorized parties. There are different types of access control mechanisms which can be utilized in blockchain applications; some of these mechanisms are explained to be used in the blockchain.

1. Role-based Access Control (RBAC): RBAC is an approach for restricting the system view to the users of the system according to their roles in the system. Thus, it can be applied in a blockchain framework where access is provided according to the user roles. RBAC is used in a blockchain based solution for Healthcare [120].

2. Attribute-based Access Control (ABAC): In ABAC, the access control rules are based on the attribute structure. These attributes can be user specific, environment-specific or object specific. For example, in a blockchain setup for the insurance industry, 'department' could be an attribute through which the access of the blockchain data is restricted, which means the claims handling department would have a different view of the blockchain compared to the audit department. ABAC can be used in a fair access blockchain model [78] by keeping attributes in policy.

3. Organization-based Access Control (OrBAC): OrBAC is one of the richest access control models. OrBAC consists of three entities (subject, action, object) which define that some subject has the permission to realize some action on some object. OrBAC has already been used in blockchain for IoT in a fair access blockchain model [78] and in dynamic access control model on blockchain [121].

Other access control mechanisms such as context-based access control and capability-based access control (proposed in blockchain solutions for autonomous vehicles, smart cities, IoT [122]) can also be useful for different blockchain solutions.

5.4 Encryption Scheme

It is a process of encoding a piece of information by which only authorized parties can access it. It can be used to achieve confidentiality of blockchain data by encrypting it. There are many encryption schemes which can be used in blockchain. Symmetric-key Encryption is used in Hyperledger fabric for confidentiality of smart contract [61] and Blockchain for Smart Home [123]. Although searching and computation over an encrypted data is a big challenge, there are many existing techniques which can be used for that purpose. Some of these techniques such as searchable encryption for searching on encrypted data in the cloud is already used in permissioned blockchain [124], and for computation over encrypted data, fully-homomorphic encryption and functional encryption can also be utilized in blockchain.

In order to assure simultaneously confidentiality and authenticity of data, an authenticated encryption can be used in blockchain. In authenticated encryption, two peers establish a connection, they both share their public keys and compute the shared secret which is used as the symmetric key for the authenticated encryption algorithm. The recently finished cryptographic competition CAESAR [125] has identified a portfolio of six ciphers for authenticated encryption. So far, as of this writing (June 2019), none of those ciphers has been deployed in some blockchain system.

Broadcast encryption can be used in blockchain to provide the anonymity of blockchain receiver nodes. [126] gives a proposal to use for Availability and Accountability for IoT by

blockchain. It has as every user in the group receives the encrypted message, although only users with the correct permission or key can decrypt it.

5.5 Secure Multi-party Computation (SMPC)

Secure Multi-party Computation enables parties to act together in a way that no single party has an access to all of the data, and hence no one can leak any secret information. The main idea of SMPC scheme is to jointly compute a function by parties over their inputs without disclosing their inputs. For example, a group of people can compute the average salary of the group without disclosing their actual individual salaries. The blockchain platform Enigma [81] leverages the concept of SMPC to achieve strong privacy. In Enigma platform, a blockchain network is combined with SMPC network, where the blockchain network contains the hashes and SMPC network contains the data corresponding to those hashes which split is among different nodes. For each node, the view over SMPC network differs as everyone has a different piece of information. Specifically, each node contains a random piece of data, and no single party ever has access to the entire data.

A blockchain model Hawk [82] for privacy-preserving smart contracts also specifies the use of SMPC to minimize the trust in the generation of common reference string in SNARK proof used in the model. SMPC can also be exercised for private data storage in a decentralized system, such as Keep [127]. Keep provides a privacy-focused storage solution for Ethereum. In this system, network nodes collaborate to provide secure decentralized data containers, called keeps, which can be accessed from smart contracts on Ethereum.

5.6 Secret Sharing

In this concept, a secret is divided into multiple parts among the participants, and it is reconstructed by using a minimum number of parts. These parts are called shares and they are unique for each participant. Secret sharing is used to secure sensitive information. Secret sharing scheme is advantageous in SMPC for distributing the shares among parties. Shamir's secret sharing [128] is already being used to distribute transaction data, without a significant loss in data integrity in blockchain [129]. Decentralized Autonomous Organizations (DAO) can take advantage of secret sharing by distributing the shares of information among the system nodes rather than storing full information in each node. Secret sharing in DAO can be practiced in consensus where each participating node stores a set of shares of the system state rather than storing full system state. These shares are points on polynomials which make up part of the state.

Secret sharing schemes are also used in different off-chain and on-chain bitcoin wallets to safeguard the private keys of the crypto holders. Secret sharing schemes can benefit blockchain by storing secret information in a decentralized way so that unauthorized parties cannot access it. Secret sharing is used in blockchain for different purposes such as secret share-based fair and secure voting protocol (SHARVOT) [83] and new cryptocurrency based on mini blockchain [130].

5.7 Commitment Scheme

A commitment scheme is a digital analog of a sealed envelop. It is a two-phase game between two parties where the phases are commit and open. Commit phase involves hiding and binding of a secret by the first party and send it to the second party; while open is to prove that the first

party did not cheat the second party in the commit phase. Therefore, a commitment scheme satisfies the aforementioned two security properties: *hiding* and *binding*. Hiding ensures that the receiver cannot see the message before the open phase, while binding ensures that the sender cannot change the message after the commit phase. The following example shows a binding commitment:

1. Pick a secret value s to commit from 0 to $p-1$ where p is a large prime number;

2. Calculate the value $c = g^s \mod p$;

3. Publish the value c as a commitment.

In the above example, the binding property follows as it is infeasible for the sender to find any other value y which gives the same c. Here finding the value s from known c, p and g is a computationally hard problem of discrete logarithm but any party can verify the commitment value c if s is provided. There are many commitment schemes such as Pedersen commitment [131] and elliptic curve Pedersen commitment. Zerocoin [77] uses Pedersen commitment to bind a serial number s to Zerocoin z. The commitment c is given as follows:

$$c = g^s h^z \mod p.$$

Here g, h, and p are known to everyone, and the user chooses s, z and computes and publishes the commitment c. These s, z cannot be computed from c even if one is provided. As a consequence, in Zerocoin when the serial number s is published, the user can prove his/her ownership by providing z. Pedersen commitment has also been used to build blockchain-oriented range proof system, Bulletproof [86] and its elliptic curve version is also successfully implemented in Monero [57]. A switch commitment scheme is designed for confidential transactions in blockchain [132].

5.8 Accumulator

An accumulator is a one-way function which gives a membership proof without revealing individual identity in the underlying set. This can be used in blockchain to build other cryptographic primitives such as commitment, ring signatures, and zero-knowledge proofs. Merkle tree, used in many cryptocurrencies, fits under a more comprehensive class of cryptographic accumulators which is space and time efficient data structure to test for set membership. Figure 6 shows how blockchain transactions are represented in the Merkle tree, and the Merkle root is stored in the block structure of the blockchain. Non-Merkle accumulators are classified as RSA accumulators and elliptic curve accumulators.

In Zerocoin [77], an accumulator A is computed by the network overall coin commitments $(c_1, c_2, \ldots, c_n)$ along with the appropriate membership witnesses for each item in the set. The witness w is computed by the accumulation coins with the exception of one. In this way, during Zerocoin spend transaction, a user proves the knowledge of one coin by using that witness. This witness w and accumulator A are publicly verifiable without any trusted third party. Accumulator A in Zerocoin is defined as:

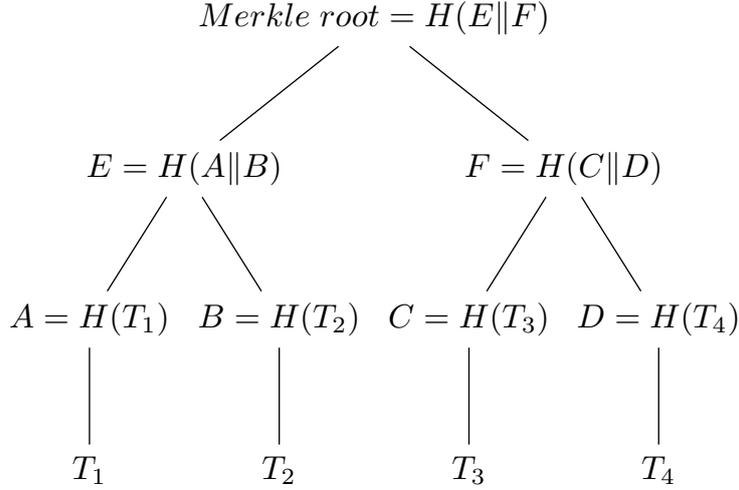$$A = u^{c_1 \ c_2 \ c_3 \ \ldots \ c \ \ldots \ c_n} \mod N,$$

Figure 6: Merkle tree of blockchain transactions.

where the integers A, u and N are known to everyone. The coin c is a prime number that is a Pedersen commitment of a coin serial number s and the random number z. The witness w of a coin c is defined as the accumulation of all coins with the exception of c:

$$A = u^{c_1 \ c_2 \ c_3 \ \cdots \ c_n} \mod N.$$

Accumulators can also be employed for range proofs in blockchain. Accumulators are used in [87] to design a stateless blockchain where in order to participate in consensus, the node only needs a constant amount of storage.

5.9 Oblivious Transfer (OT)

Oblivious Transfer is a two-party protocol between a sender S and a receiver R. The general type of oblivious transfer is k-out-of-n oblivious transfer $\binom{n}{k}$-OT, where $k < n$, in which S holds n messages and R retrieves simultaneously k of them without letting S know about which k out of n messages R received. Oblivious transfer is introduced by Rabin [133] in which a sender sends a message to a receiver with probability $\frac{1}{2}$. The protocol is called as $\frac{1}{2}$-OT, and it is as following:

1. Sender S chooses two large primes p, q and computes $N = pq$ and then the sender generates RSA public key (e, N) such that e is relatively prime to $(p-1)(q-1)$.

2. S computes cipher text c over message M as $c = E_{(e,N)}(M) = M^e \mod N$ and sends e, N, c to receiver R.

3. R chooses a random $x \in \mathbb{Z}_{N^*}$ and sends $a = x^2 \mod N$ to S.

4. S computes four square roots of $a \mod N$ and chooses one of the roots y at random and sends it to R.

5. R checks whether $y^2 \equiv a \mod N$ and if $y \not\equiv \pm x \mod N$, then R will be able to factor N and, hence, be able to decrypt c to recover M.

$\frac{1}{2}$-OT is *complete* for secure multi-party computation. Oblivious transfer has been realized in secure multiparty computation to create private and verifiable smart contracts on blockchain [134]. Oblivious transfer can also be utilized for exchange of secrets, private information retrieval, and building protocols for signing contracts. There has been loads of work done in oblivious transfer, and some of these works have been applied in blockchains such as Searchain [88] and APDB [89] (for automated penalization of data breaches using crypto-augmented smart contracts).

5.10 Oblivious RAM (ORAM)

Oblivious RAM is a cryptographic protocol through which a client can safely store his/her data in an untrusted server. The client performs read and write operations remotely. ORAM hides the memory access pattern from the server as well as from outside entities accessing to that part of the data. Therefore, if a client performs two operations of equal length, then the polynomial-bounded adversarial server cannot distinguish between these operations. ORAM bestows freshness, confidentiality of data and integrity so it can be used in various blockchain use-cases and applications. Solidus [90], a protocol for confidential transactions on public blockchain, uses oblivious RAM. Solidus framework operates on a modest number of banks where each bank maintains a large number of user accounts. Solidus introduces a new primitive called Publicly Verifiable Oblivious RAM Machine (PVORM). Most of the previous usage of Oblivious RAM is performed by a single client to outsource storage. In Solidus, ORAM is used to store user account balances and uses PVORM to verify the valid transaction set of a bank. Oblivious RAM is also used in the client-server ORAM protocol [91], Externally Verifiable Oblivious RAM, where Ethereum's automated crypto-currency contracts adjudicate the disputes occurred due to the malicious server by penalizing the server.

5.11 Proof of Retrievability (POR)

With the advent of cloud computing, a client might outsource his/her data to the cloud, but still, the client needs a guarantee that the old data has not been modified or deleted. This can be achieved by using proof of retrievability [135] which is an interactive mechanism between a client (verifier) and a server (prover) where the server provides a compact proof to the client that his/her data is intact and he/she can recover the data at any point of time. In this direction, to verify the proof, the client should be equipped with devices having some computational power and network access. This requirement hinders the large-scale adoption of POR by cloud users. To solve this issue, outsource proof of retrievability (OPOR) [136] is introduced where external auditors verify the POR with the cloud provider on behalf of the clients. OPOR protocol specification uses Bitcoin functionalities for the building blocks.

Permacoin [93] uses proof of retrievability. The primary goal of Permacoin is the distributed storage of archival data. As in Bitcoin's mining mechanism, the client continuously invests his/her computational power, and in addition to the computational power, his/her storage is invested. As a consequence, Permacoin requires storage overhead and high bandwidth consumption. To solve these issues, *Retricoin* [94] is proposed to repurpose the mining work in order to ensure the retrievability of a large file at any point of time. Retricoin also proposes a new algorithm for miners to mine collectively. Storj [95] also uses POR to prove the existence of a fresh copy of a shard on the storer side. As a result, POR can be employed in many cryptocurrencies and blockchain applications.

5.12 Post-Quantum Cryptography

Recent advances in quantum computing pose a severe threat to classical cryptography, as most of the widely used cryptography is based on the hardness of some problem which can be efficiently solved using quantum computers. Thus, research in the Post-Quantum cryptography [137] has taken a massive leap. The security impact of breaking public key cryptography by quantum computers would be tremendous. Elliptic curve cryptography (ECC), which is an approach to public key cryptography, is mostly used in blockchain applications. Using a variant of Shor's algorithm [138], a quantum computer can easily forge an elliptic curve signature that underpins the security of each transaction in blockchain and so breaking of ECC will affect blockchain in terms of broken keys, hence, digital signatures.

Research in this field is in the rise to create Post-Quantum resistant digital signatures (BPQS) [139] which is a hash-based signature and uses one-time signature (OTS) schemes as a building block. OTS does not depend on any number-theoretic hard problem, and it requires only a secure cryptographic hash function, hence, it is not vulnerable to Shor's algorithm. BPQS has advantages like shorter signatures, faster key generations, and customizable property. Post-Quantum cryptography is also used to design Post-Quantum blockchain [99] using one-time signature chains or to create secure crypto-currency based on Post-Quantum blockchain [100].

For the quantum proof solutions, research is now focused on Lattice-based cryptography [140], multivariate cryptography [141], hash-based cryptography [137], and code-based cryptography [142]. Most of the developed primitives within these areas offer either signatures or public keys that are orders of magnitude bigger than the currently used ones, and that is really a hard research challenge that we formulate as:

Research Problem 9 *Construct a new blockchain mechanism that has comparably efficient public key addresses and comparably small digital signatures as the currently used ones, but that is based on Post-Quantum cryptographic schemes.*

5.13 Lightweight Cryptography

Conventional cryptographic methods such as RSA and SHA256, work well on systems having reasonable memory and processing power, but these methods are not suitable for devices constrained with memory, physical size, and battery. Conventional cryptographic methods are challenging to implement in resource-constrained devices due to implementation size, large key size, throughput, speed, and energy consumption. Nevertheless, to solve these issues, lightweight cryptography has evolved. Lightweight cryptography targets sensor networks, embedded systems and other variety of resource-constrained devices such as IoT end nodes and RFID tags. Lightweight cryptography is simpler and faster than conventional cryptography but less secure (suffers from many attacks). In IoT, embedded devices having sensors are interconnected through a public or private network. As these are resource-constrained devices, lightweight cryptography solves the issues of communication, memory, and power consumption, but still lacks security. To provide better security, blockchain can be used in conjunction with the sensor network.

Reference [143] reinforces our point to use lightweight cryptography and blockchain for IoT devices to improve security (confidentiality and integrity of IoT device data). A lightweight scalable blockchain (LSB) [102] is also introduced to improve IoT security and privacy. LSB uses a lightweight hash function and lightweight consensus algorithm in order to achieve scalability, security, and privacy. Blockchain is also used to cater security in electric vehicles, cloud and

edge computing [103] which use lightweight cryptographic primitives like lightweight symmetric key encryption.

5.14 Verifiable Random Function (VRF)

This cryptographic primitive [144] is a pseudorandom function which gives a public verifiable proof of its output based on public input and private key. In short, it maps inputs to verifiable pseudorandom outputs. VRFs can be used to provide deterministic precommitments which can be revealed later using proofs. VRFs are resistant to pre-image attacks unlike traditional digital signature. VRF is a triple of the following algorithms:

- *KeyGen(r)→(VK,SK)*. Key generation algorithm generates verification key *VK* and secret key *SK* on random input *r*.

- *Eval(SK,M)→(O,π)*. Evaluation algorithm takes secret key *SK* and message *M* as input and produces pseudorandom output string O and proof π.

- *Verify(VK,M,O,π)→0/1*. Verification algorithm takes input as verification key *VK*, message *M*, output string *O*, and proof π. It outputs 1 if and only if it verifies that *O* is the output produced by the evaluation algorithm on input secret key *SK* and message *M*, otherwise it outputs 0.

In context of blockchain, many Proof of Stake blockchains use VRF to perform secret cryptographic sortition such that electing leader and committee as part of underlying consensus protocol. Proof of Stake blockchain protocols given in [145] use VRF to elect block proposers and voting committee members. Algorand [37] and Witnet network protocol [146] also employ VRF to conduct secret cryptographic sortition. Ouroboros Praos [104] uses VRF on current timestamp and nonce to determine whether a participant is eligible to issue a block. Dfinity [105] network is a decentralized cloud computing resource which uses VRF to generate stream of outputs over time. Thus, the usage of verifiable random function brings many advantages to be exploited in blockchain and opportunities for more research.

5.15 Obfuscation

Obfuscation is a way of transforming a program P into a "Black-box" equivalent of the program $Q = O(P)$ so that P and Q give the same output when the given inputs are same. It should be hard to find out the internal logic or structure of the program once it is obfuscated. Obfuscation aims to make reverse engineering difficult by making the program unintelligible while preserving its functionality. Finding a perfect black-box obfuscation is mathematically impossible. Along these lines, a weaker solution is to find an "Indistinguishability Obfuscation" so that one cannot determine whether the generated output is from the original program or the obfuscated program. A very simplified example for understanding the Indistinguishability Obfuscation, is the following: There are two equivalent programs $P = x * (y + z)$ and $P' = x * y + x * z$. They are obfuscated such that we have $O(P)$ and $O'(P')$. We say that the obfuscated programs O and O' are indistinguishable if on a received output o one cannot determine which of the programs O, O' gave that output.

Obfuscation can be applied for witness encryption, functional encryption, and restricted use of software. It can be applied in blockchain to turn smart contract into a black-box. An

obfuscated smart contract can also possess a secret key to decrypt an encrypted input to the smart contract. As a result, publicly running contracts can possess secret data inside it by obfuscating the smart contract. Figure 7 depicts an obfuscated smart contract which stores the private key corresponding to a public key which is used to encrypt the transaction data. It is hard to get the corresponding private key because of the obfuscated smart contract.
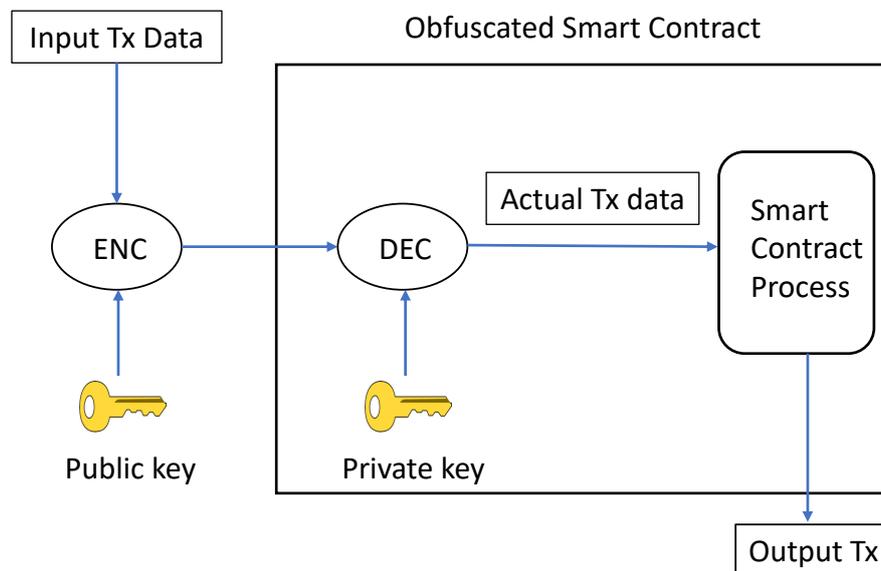

Figure 7: An example of smart contract obfuscation.

One of the very first successful attempts to offer a very limited variant of obfuscation in Bitcoin was the standardization of the "Pay to script hash (P2SH) transactions" [147]. The amounts of Bitcoins in P2SH transactions are sent to a script hash instead of a public key hash. We say that it was a limited variant of obfuscation because in order to spend Bitcoins received via P2SH, the recipient must provide a script that matches the script hash. Still, the successful acceptance of the P2SH transactions without causing a hard fork in Bitcoin showed that there is an interest in obfuscation in Blockchain, and that subject is a viable research area.

Research on obfuscation in Bitcoin [92] has been conducted and can be compiled for other cryptocurrencies and blockchain applications. Obfuscation is also used in blockchain for power grid consumption [148] where noise is added to the user's electricity consumption data through obfuscation, and the electricity consumption data is divided into random and non-random obfuscated data.

As noted in [149] the definition and characteristics of some languages determine how easy is to obfuscate programs written in those languages. For example C, C++, Java and Perl are languages that offer easier program obfuscation. What about scripting languages used in Blockchain? We reformulate this question as a research problem:

Research Problem 10 *Study the easiness/hardness of obfuscating programs written in the scripting languages used in the current blockchain systems. Study the feasibility of applying some of the developed obfuscation techniques in C, C++, Java and Perl for the blockchain scripting languages.*

6 Promising but yet not employed cryptographic primitives in blockchain

This Section construes some cryptographic concepts which are promising candidates to be utilized in blockchain. These cryptographic concepts are not yet well-studied and fully applied in blockchain but constitute of some excellent properties which overlap with some desired properties of blockchain. Therefore, some use cases and blockchain services can benefit from these concepts. The included concepts in this Section have either not at all been studied for use in blockchain or have been studied but not implemented yet. We include references which show some initial ideas how to use these concepts in blockchain, but these references do not give any details about concrete implementation.

6.1 Aggregate Signature

An aggregate signature allows creating a single compact signature from k signatures on k distinct messages from k distinct signers. It provides faster verification as well as reduction in storage and bandwidth. As in blockchain, the requirement of storage and computation is high; aggregate signatures can be used for reduction in storage and computation. Aggregate signatures are the non-trivial generalization of multi-signatures (where all users sign the same message). There are two primary mechanisms of signature aggregation: general and sequential aggregation. In order to describe these mechanisms, assume a set of k users having public-private key pair (PK_i, SK_i) and user i wants to sign message M_i.

1. In general signature aggregation scheme, each user i (from the group of k users) creates signature σ_i on his/her message M_i. Now to create aggregate signature, anyone can run public aggregation algorithm to take all k signatures $\sigma_1, \sigma_2, \ldots, \sigma_k$ and compress them into a single signature σ.

2. In sequential signature aggregation scheme, user 1 signs M_1 to obtain σ_1; user 2 then combines σ_1 and M_2 to obtain σ_2; and so on. The final signature σ is generated by user k which binds M_k and the signature σ_{k-1}. Sequential signature aggregation can only take place during the signing process.

Techniques for aggregating signatures are known for a variety of signature schemes such as DSA, Schnorr, pairing-based, and lattice-based. Aggregate signature schemes should restrict any adversary from creating a valid aggregate signature on his/her own. Aggregate signatures have been proposed for Bitcoin [84], and they can be applied to other cryptocurrencies and blockchain designs.

Research Problem 11 *Construct an efficient new signature scheme based on aggregate signatures, that is specifically tailored for blockchain transactions.*

6.2 Identity-Based Encryption (IBE)

Identity-Based Encryption first proposed as idea in [150] and later realized as complete cryptographic primitive in [151], allows the encrypting party to use any known (or supposedly known) identity of any receiving party as its public key. Upon receiving the encrypted message, the

receiving party asks a trusted third party "Private Key Generator (PKG)" to generate the corresponding private key. Then the receiver decrypts the message using the private key received by PKG. Nowadays, by using identity-based encryption, public keys can be generated using the social identities (Facebook, Twitter, LinkedIn).

There are many flavors and extensions of IBE such as Hierarchical IBE [152], Attribute-based encryption [153], Decentralized attribute-based encryption [154], Functional encryption [155] to name a few.

One of the specifics of IBE is that it replaced the role of the Public-Key Infrastructure with the trusted third party PKG. The presence of a trusted third party somehow defeats the purpose to use it in permissionless blockchain, but still there is a scope to use it in the distributed ledger. Namely, it seems that IBE can be used in permissioned blockchain network. In permissioned blockchain a consortium of trusted third parties that distribute the private keys to the users can take the role to be IBE PKG. Another variant could be a smart contract layer being responsible for the generation of public-private key pairs inside the PKG using IBE.

We identified that the use of IBE within blockchain has started in [85]. Still, there are a lot of challenges and opportunities for other blockchain applications and services.

Research Problem 12 *Construct an IBE based (or IBE related) permissioned blockchain network.*

6.3 Verifiable Delay Function (VDF)

Verifiable Delay Function (VDF) is a function $f : X \to Y$ which takes a prescribed number of sequential steps to compute; however, the output can be easily verifiable by anyone. This delay function prevents malicious miners from computing the random output, and it also provides a short proof which is used during the verification of the output along with previously generated public parameters. Boneh et al. described the concept of VDF [156] as well as illustrated the idea about how it can be applicable to blockchain. VDF can be efficiently used as a way to add a delay in decentralized applications. VDF can be used in the application of decentralized systems such as in leader election process of consensus mechanisms, constructing randomness beacons and proofs of replication.

Delay function was initially implemented in Ethereum prototype [157] where the main idea was verification of delay functions through smart contract by using a multi-round protocol. After this prototype implementation, the concept of verifiable delay function was proposed by Boneh et al. Nowadays several blockchain industries are trying to use VDF in their consensus mechanisms. Chia Network [106] which is open source blockchain is trying to use VDF in its "Proof of space and time" consensus mechanism. Ethereum is also trying to develop a pseudorandom number generator using VDF. In this way, VDF brings opportunities to dig deeper and to be applied in the blockchain domain.

6.4 Private Information Retrieval (PIR)

It is a cryptographic primitive in which a client queries to a server and retrieves the corresponding response from the server without exposing query terms as well as response. It is a weaker version of 1-out-of-n oblivious transfer. It can facilitate private blockchain queries to fetch transaction data privately from blockchain. Accordingly, it can be used to find out whether a particular transaction has been appended in the blockchain or can be used to check the transactions

associated with the set of public keys and find out the remaining balances. In addition, PIR can be helpful to query transaction data in simplified payment verification (SPV) clients without compromising privacy. PIR requires an adequate amount of processing, but in the future there might be efficient PIR techniques which can be implemented in blockchain. PIR has also been applied in distributed storage [158] which can be further investigated and adopted in blockchain.

Paper [73] sets several research problems in the area of blockchain transactions privacy and private information retrieval. We rephrase some of the research challenges postulated there:

Research Problem 13 ([73]) *Develop protocols where non-anonymous users can publish transactions that cannot be linked to their network addresses or to their other transactions.*

Research Problem 14 ([73]) *Develop protocols where non-anonymous users can fetch details of specific transactions without revealing which transactions they seek.*

Research Problem 15 ([73]) *Develop efficient and scalable protocols for anonymous publishing on permissioned blockchains, by combining the asynchronous Byzantine-tolerant consensus protocols for agreeing on transactions with the process of mixing users' announcements.*

6.5   Decentralized Authorization

Authorization and/or hiding sensitive data and actions are essential concepts of resource sharing in open and collaborative environments such as the Internet. Furthermore, in a decentralized form of authorization, parties have full control over their resources and authority to delegate it whether entirely or in part to other parties. An authorization system should provide only as little access to the users as possible to perform their jobs.

Traditional access control is a centralized authorization server which imposes a problem of single point of failure. The centralized authorization scheme has different methods of authorization such as access control list or role-based access.

In comparison, decentralized authorization is more efficient and easier in terms of time, resource and quality. A decentralized authorization system should be well administrated to give access privileges to the users. On the negative side, having in mind that the auditing is also a key component of authorization, in a decentralized manner, it is hard to efficiently implement it and to enforce it.

By using blockchain smart contract, some decentralized authorization systems have been designed, e.g., BlendCAC [96] and WAVE [97]. WAVE introduces an authorization layer for the name spaces and resources. Moreover, for the outside entities, a delegation of trust is used to obtain permission on a resource. Decentralized authorization and blockchain can be used to grow each other by combining one another in a specific way.

Research Problem 16 *Construct a decentralized authorization protocol for permissioned blockchain that will provide access privileges as well as a delegation of these access to the users.*

6.6   White-Box Cryptography

White-box attack is a threat model where the attacker has full visibility of the internal data flow and can modify the data and code. White-box cryptography [159] aims to address the challenge of implementing a cryptographic algorithm in software in such a way that cryptographic assets

remain secure even when subject to white-box attacks. A white-box cryptographic implementation must resist black-box (the attacker has access to only input and output of algorithm), grey-box (side-channel), and also white-box attacks. White-box cryptography is a way to implement cryptographic algorithms like RSA and AES so that the keys remain hidden all the time even during the execution. In some white-box implementations, the key is baked into the code and further concealed to use it in a cryptographic algorithm. In blockchain, it can be used to hide the private key inside the smart contract, and that key can be unlocked when smart contract executes and further it can be used to create a signature.

White-box cryptography can be orchestrated in blockchain to establish trust and privacy of assets. As in blockchain, key and seed secrets are a single point of compromise; these are the highly vulnerable and lucrative targets when stored in memory. To safely store the key, it can be obfuscated in white-box cryptography and further used for encryption/decryption. The implementation of white-box cryptography should be strong enough to facilitate the key storage in blockchain. It has been used in runtime self-protection in a trusted blockchain-inspired ledger [98] and can be promoted in other blockchain applications and services.

6.7 Incremental Cryptography

The idea behind incremental cryptography [160] is if there is a modification to some document M to M', then the time to update the result upon modification of M should be "proportional" to the "amount of modification" done to M. Incremental cryptography can be used in incremental collision free-hashing or incremental digital signature. The initial idea proposed for incremental cryptography uses the example of a digital signature. The idea was to have a digital signature which is easy to update upon the modification of the underlying message. Suppose M is a message and σ is the corresponding signature. If M is changed to M' by adding/deleting any block, then the time to update the signature from σ to σ' should be "proportional" to the "amount of modification" done to get M' from M.

A proposal for construction of an incremental hash function based on SHA-3 is given in [161], and a private blockchain "Kadena" [80] proposes the use of either Merkle tree or incremental hashing for transaction verification. The concept of incremental hashing in Kadena blockchain is to update the distributed log among the blockchain nodes.

Research Problem 17 *Construct a new blockchain mechanism that uses an incremental hash function for updates of the distributed ledger.*

6.8 Identity-based Broadcast Encryption (IBBE)

IBBE scheme [162] can be considered as a generalization of identity-based encryption scheme (Section 6.2) where instead of having one receiver, there are multiple receivers. In broadcast encryption the users are recognized by their identities rather than by their public keys. In a multi-receiver setting, IBBE proves as a powerful method to provide data security and privacy. In this scheme, a sender broadcasts the encrypted message to an intended set of users called privilege set. There can be many privilege sets with different cardinalities. A revocable IBEE scheme [163] shows a scenario of IBEE in which the involved players are the key authority, revoked and non-revoked users. In this setting, the decryption key is updated through the release of a key update material by the key authority. These decryption keys are updated only for the non-revoked users. In this scheme, a membership is revoked for a user if he/she is found

malicious or his/her keys are compromised. This RIBBE scheme is further implemented in Charm framework [164].

As blockchain is a multi-receiver setting, IBBE can be a propitious candidate to provide transaction data security and privacy. It can also be used in a permissioned blockchain to certify blocks of membership operation logs. RIBBE scheme as being very efficient in terms of computational complexity and communication can work efficiently as well in the case of blockchain.

Research Problem 18 *Develop protocols to certify the blocks of membership operation logs in permissioned blockchain setting.*

6.9 Other Techniques:

1. *Message Authentication Code (MAC)*: It is a short piece of information (known as a tag) to authenticate a message which states that the message comes from the stated sender and it has not been changed. It can be used in blockchain to provide integrity of smart contracts or network data. A blockchain-based system for secure mutual authentication (BSeIn) [165] uses MAC for the authentication.

2. *Non-Interactive Witness Indistinguishability (NIWI)*: These are proof systems which are weaker variants of Non-Interactive zero-knowledge (NIZK) proofs. Witness Indistinguishable property states that the verifier cannot distinguish which witness is used to prove the statement by the prover, considering the case of existence of many witnesses. NIWI has been used to construct NIZK over POS based blockchain protocol [166] as well as recently, a new construction of publicly verifiable NIWI proofs from blockchain [167] is also proposed. Hence NIWI proofs bring a new direction to be exploit within the blockchain domain.

3. *Position-based Cryptography*: In this cryptographic protocol [168], the identity or the credentials of a party are derived from his/her geographical location. These credentials can be further used for position-based secure communication and position-based authentication. Position-based cryptography has not been applied in blockchain yet, but it looks promising.

4. *Elliptic Curve Diffie-Hellman Merkle (ECDHM) addresses*: These addresses [169] can be used to exchange messages privately in the blockchain. It can be used in blockchain for secure communication among parties. ECDHM address is shared between the sender and the receiver as secret shares, and they use this shared secret to derive anonymous transacting addresses of each other. This address may only be exposed once they have the share to construct these addresses. In this way, it can be used for the privacy of transaction data.

7 Conclusion

The goal of this work was to offer a systematic study of available cryptographic concepts and to identify different research directions and problems. Based on these reviewed concepts and associated properties, we hope that the paper will help cryptographers interested in blockchain

to choose a challenging research problem and for practitioners to choose a suitable concept for their particular use case.

Current transitions to blockchain enabled solutions by different industries give rise to more research on this technology. Academic and industrial research is focused on making blockchain cost efficient in terms of computational power, memory requirements and security. Many existing cryptographic concepts have been embraced for blockchain use. This paper systematizes the current state-of-the-art knowledge of existing cryptographic concepts used in the blockchain. It also gives a brief description of the used cryptographic concept and points to the available blockchain models that are using that concept. The paper also identifies some concepts which have not yet been used in blockchain but can be beneficial if applied in the blockchain. Apart from existing cryptographic concepts, the paper also presents the basic building blocks of blockchain and how these building blocks are dependent on each other.

Table 5 summarizes all of the cryptographic concepts (used or with potentials to be used in blockchain) presented in this work.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf," 2009.

[2] CoinMarketCap, "Total market capitalization," https://coinmarketcap.com/charts/, May 2019, [Online; accessed 16-June-2019].

[3] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Boston, MA: Springer US, 1983, pp. 199–203.

[4] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *CRYPTO 92, Annual International Cryptology Conference*. Springer, 1992, pp. 139–147.

[5] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," Cambridge, MA, USA, Tech. Rep., 1996.

[6] E. Hughes, "A cypherpunks manifesto," *URL (accessed 18 April 2019): https://www.activism.net/cypherpunk/manifesto.html*, 1993.

[7] A. Back, "The Hashcash Proof-of-Work Function," *Draft-Hashcash-back-00, Internet-Draft Created,(Jun. 2003)*, 2003.

[8] W. Dai, "b-money, 1998," *URL (accessed 18 April 2019): http://www.weidai.com/bmoney.txt*, 1998.

[9] N. Szabo, "Bit gold, 2005," *URL (accessed 18 April 2019): https://unenumerated.blogspot.com/2005/12/bit-gold.html*, 1998.

[10] N. Satoshi, "Re: They want to delete the Wikipedia article," https://bitcointalk.org/index.php?topic=342.msg4508#msg4508, July 2010, [Online; accessed 18-Apr-2019].

[11] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Yellow Paper, 2014.

[12] E. Foundation, "The Ether denominations are called Finney, Szabo, and Wei. What/who are these named after?" https://ethereum.stackexchange.com/questions/253/, January 2016, [Online; accessed 30-Apr-2019].

[13] H. Finney, "Bitcoin and me (Hal Finney)," https://bitcointalk.org/index.php?topic= 155054.0, March 2013, [Online; accessed 30-Apr-2019].

[14] V. Morabito, "Business innovation through blockchain," *Cham: Springer International Publishing*, 2017.

[15] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3416–3452, Fourthquarter 2018.

[16] W. Wang, D. T. Hoang, Z. Xiong, D. Niyato, P. Wang, P. Hu, and Y. Wen, "A Survey on Consensus Mechanisms and Mining Managemen in Blockchain Networks," *CoRR*, vol. abs/1805.02707, 2018. [Online]. Available: http://arxiv.org/abs/1805.02707

[17] L. Wang, X. Shen, J. Li, J. Shao, and Y. Yang, "Cryptographic primitives in blockchains," *Journal of Network and Computer Applications*, vol. 127, pp. 43 – 58, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S108480451830362X

[18] B. Preneel, "The state of cryptographic hash functions," in *School organized by the European Educational Forum.* Springer, 1998, pp. 158–182.

[19] P. Gallagher and A. Director, "Secure hash standard (SHS)," *FIPS PUB*, vol. 180, p. 183, 1995.

[20] A. Regenscheid, R. Perlner, S.-j. Chang, J. Kelsey, M. Nandi, and S. Paul, "Status report on the first round of the SHA-3 cryptographic hash algorithm competition," *Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD*, pp. 20 899–8930, 2009.

[21] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Colavita, M. Virza, and T. Dryja, "Cryptanalysis of Curl-P and Other Attacks on the IOTA Cryptocurrency," *IACR Cryptology ePrint Archive*, vol. 2019, p. 344, 2019.

[22] E. Heilman, N. Narula, T. Dryja, and M. Virza, "Iota vulnerability report: Cryptanalysis of the curl hash function enabling practical signature forgery attacks on the iota cryptocurrency," 2017.

[23] C. Lee, "Litecoin," 2011. [Online]. Available: https://litecoin.org

[24] C. Percival, "Stronger key derivation via sequential memory-hard functions," 2009.

[25] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," Tech. Rep., 1997.

[26] D. J. Bernstein, "The Salsa20 family of stream ciphers," in *New stream cipher designs.* Springer, 2008, pp. 84–97.

[27] V. Buterin, "QuarkCoin: Noble Intentions, Wrong Approach," *Bitcoin Magazine*, Dec 2013, [Online; accessed 3-Jun-2019].

[28] M. S. Turan, R. Perlner, L. E. Bassham, W. Burr, D. Chang, S. jen Chang, M. J. Dworkin, J. M. Kelsey, S. Paul, and R. Peralta, "Status report on the second round of the SHA-3 cryptographic hash algorithm competition," *NIST Interagency Report*, vol. 7764, 2011.

[29] D. Gligoroski, V. Klima, S. J. Knapskog, M. El-Hadedy, and J. Amundsen, "Cryptographic hash function Blue Midnight Wish," in *2009 Proceedings of the 1st International Workshop on Security and Communication Networks*. IEEE, 2009, pp. 1–8.

[30] E. Duffield and D. Diaz, "Dash: A payments-focused cryptocurrency," Whitepaper, https://github.com/dashpay/dash/wiki/Whitepaper, 2018, [Online; accessed 3-Jun-2019].

[31] Open Source Community at Github, "ProgPoW - A Programmatic Proof of Work," *Github*, 2018, [Online; accessed 3-Jun-2019].

[32] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Consensus in the age of blockchains," *CoRR*, vol. abs/1711.03936, 2017. [Online]. Available: http://arxiv.org/abs/1711.03936

[33] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A Scalable Blockchain Protocol." in *NSDI*, 2016, pp. 45–59.

[34] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[35] L. Ren, "Proof of stake velocity: Building the social currency of the digital age," *Self-published white paper*, 2014.

[36] J. Kwon, "Tendermint: Consensus without mining," https://tendermint.com/static/docs/tendermint.pdf, 2014.

[37] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ser. SOSP '17. New York, NY, USA: ACM, 2017, pp. 51–68. [Online]. Available: http://doi.acm.org/10.1145/3132747.3132757

[38] A. Kiayias, I. Konstantinou, A. Russell, B. David, and R. Oliynykov, "A Provably Secure Proof-of-Stake Blockchain Protocol." *IACR Cryptology ePrint Archive*, vol. 2016, p. 889, 2016.

[39] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of Luck: An Efficient Blockchain Consensus Protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, ser. SysTEX '16. ACM, 2016, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/3007788.3007790

[40] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On Security Analysis of Proof-of-Elapsed-Time (PoET)," in *Stabilization, Safety, and Security of Distributed Systems*, P. Spirakis and P. Tsigas, Eds. Springer International Publishing, 2017, pp. 282–297.

[41] I. Bentov, R. Pass, and E. Shi, "Snow White: Provably Secure Proofs of Stake," *IACR Cryptology ePrint Archive*, vol. 2016, p. 919, 2016.

[42] E. Duffield, H. Schinzel, and F. Gutierrez, "Transaction locking and masternode consensus: A mechanism for mitigating double spending attacks," *CryptoPapers.info*, 2014, [Online; accessed 3-Jun-2019].

[43] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Advances in Cryptology - EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Springer Berlin Heidelberg, 2015, pp. 281–310.

[44] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 142–157.

[45] A. Biryukov and D. Khovratovich, "Equihash: Asymmetric proof-of-work based on the generalized birthday problem," *Ledger*, vol. 2, pp. 1–30, 2017.

[46] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, 1999, pp. 173–186.

[47] D. Schwartz, N. Youngs, A. Britto *et al.*, "The Ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, 2014.

[48] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, "Blockchain mining games," in *Proceedings of the 2016 ACM Conference on Economics and Computation*, ser. EC '16. New York, NY, USA: ACM, 2016, pp. 365–382. [Online]. Available: http://doi.acm.org/10.1145/2940716.2940773

[49] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *CoRR*, vol. abs/1112.4980, 2011. [Online]. Available: http://arxiv.org/abs/1112.4980

[50] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar, "On bitcoin and red balloons," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, ser. EC '12. New York, NY, USA: ACM, 2012, pp. 56–73. [Online]. Available: http://doi.acm.org/10.1145/2229012.2229022

[51] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 53–65.

[52] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman

[53] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 375–392.

[54] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004. [Online]. Available: http://doi.acm.org/10.1145/997150.997156

[55] M. Vasek, M. Thornton, and T. Moore, "Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem," in *Financial Cryptography and Data Security*, R. Böhme, M. Brenner, T. Moore, and M. Smith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 57–71.

[56] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized Anonymous Payments from Bitcoin," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 459–474.

[57] G. Maxwell and A. Poelstra, "Borromean Ring Signatures," https://raw.githubusercontent.com/Blockstream/borromean_paper/master/borromean_draft_0.01_34241bb.pdf, 2015, [Online; accessed 8-Jun-2019].

[58] R. F. Arthur Britto, David Schwartz, "Ripple," 2012. [Online]. Available: https://ripple.com

[59] E. IO, "EOS. IO technical white paper," *EOS. IO (accessed 18 December 2017) https://github.com/EOSIO/Documentation*, 2017.

[60] LTO Network, "Blockchain for decentralized workflows," 2014. [Online]. Available: www.lto.network

[61] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018, pp. 30:1–30:15.

[62] "Monax," 2014. [Online]. Available: https://monax.io/

[63] G. Greenspan, "MultiChain Private Blockchain," https://www.multichain.com/download/MultiChain-White-Paper.pdf, 2015.

[64] Bitcoin Org, "SPV, Simplified Payment Verification," https://bitcoin.org/en/glossary/simplified-payment-verification, 2012, [Online; accessed 8-Jun-2019].

[65] R. Skudnov, "Bitcoin clients," 2012. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/47166/Skudnov_Rostislav.pdf

[66] L. H. White, "The market for cryptocurrencies," *Cato J.*, vol. 35, p. 383, 2015.

[67] B. McLannahan, "Bitcoin exchange mt gox files for bankruptcy protection," *Financial Times (ft. com), February*, vol. 28, 2014.

[68] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable off-chain instant payments," https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf, 2016, [Online; accessed 8-Jun-2019].

[69] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol For Open Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 17–30. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978389

[70] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White paper*, 2017.

[71] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, p. 72, 2014.

[72] C. Burchert, C. Decker, and R. Wattenhofer, "Scalable funding of Bitcoin micropayment channel networks," *Royal Society open science*, vol. 5, no. 8, p. 180089, 2018.

[73] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: challenges and directions," *IEEE Security & Privacy*, vol. 16, no. 4, pp. 38–45, 2018.

[74] N. Kshetri, "5G in E-Commerce Activities," *IEEE IT Professional*, vol. 20, no. 4, pp. 73–77, 2018.

[75] R. H. Josias N. Dewey and R. Plasencia, "Blockchain and 5G-Enabled Internet of Things (IoT) Will Redefine Supply Chains and Trade Finance," *The Secured Lender*, Jan/Feb 2018.

[76] N. van Saberhagen, "Cryptonote," 2013. [Online]. Available: https://cryptonote.org/whitepaper.pdf

[77] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, May 2013, pp. 397–411.

[78] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1748

[79] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in iot," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Á. Rocha, M. Serrhini, and C. Felgueiras, Eds. Cham: Springer International Publishing, 2017, pp. 523–533.

[80] W. Martino, "Kadena: The first scalable, high performance private blockchain," 2016.

[81] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *CoRR*, vol. abs/1506.03471, 2015. [Online]. Available: http://arxiv.org/abs/1506.03471

[82] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 839–858.

[83] S. Bartolucci, P. Bernat, and D. Joseph, "SHARVOT: secret share-based voting on the blockchain," *CoRR*, vol. abs/1803.04861, 2018. [Online]. Available: http://arxiv.org/abs/1803.04861

[84] Y. Zhao, "Aggregation of gamma-signatures and applications to bitcoin," Cryptology ePrint Archive, Report 2018/414, 2018, https://eprint.iacr.org/2018/414.

[85] S. Wei, S. Li, P. Liu, and M. Liu, "Bavp: Blockchain-based access verification protocol in leo constellation using ibe keys," *Security and Communication Networks*, vol. 2018, pp. 1–14, 05 2018.

[86] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 319–338. [Online]. Available: doi.ieeecomputersociety.org/10.1109/SP.2018.00020

[87] D. Boneh, B. Bünz, and B. Fisch, "Batching techniques for accumulators with applications to iops and stateless blockchains," Cryptology ePrint Archive, Report 2018/1188, Tech. Rep., 2018.

[88] P. Jiang, F. Guo, K. Liang, J. Lai, and Q. Wen, "Searchain: Blockchain-based private keyword search in decentralized storage," *Future Generation Computer Systems*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17318630

[89] E. V. Mangipudi, K. Rao, J. Clark, and A. Kate, "Automated penalization of data breaches using crypto-augmented smart contracts," Cryptology ePrint Archive, Report 2018/1050, 2018, https://eprint.iacr.org/2018/1050.

[90] E. Cecchetti, F. Zhang, Y. Ji, A. Kosba, A. Juels, and E. Shi, "Solidus: Confidential distributed ledger transactions via pvorm," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 701–717. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134010

[91] J. Gancher, A. Groce, and A. Ledger, "Externally verifiable oblivious ram," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 149 – 171, 2017. [Online]. Available: https://content.sciendo.com/view/journals/popets/2017/2/article-p149.xml

[92] A. Narayanan and M. Möser, "Obfuscation in bitcoin: Techniques and politics," *CoRR*, vol. abs/1706.05432, 2017. [Online]. Available: http://arxiv.org/abs/1706.05432

[93] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in *2014 IEEE Symposium on Security and Privacy (SP)*, vol. 00, May 2014, pp. 475–490. [Online]. Available: doi.ieeecomputersociety.org/10.1109/SP.2014.37

[94] B. Sengupta, S. Bag, S. Ruj, and K. Sakurai, "Retricoin: Bitcoin based on compact proofs of retrievability," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ser. ICDCN '16. New York, NY, USA: ACM, 2016, pp. 14:1–14:10. [Online]. Available: http://doi.acm.org/10.1145/2833312.2833317

[95] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," 2014.

[96] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Blendcac: A blockchain-enabled decentralized capability-based access control for iots," *CoRR*, vol. abs/1804.09267, 2018. [Online]. Available: http://arxiv.org/abs/1804.09267

[97] M. P. Andersen, J. Kolb, K. Chen, G. Fierro, D. E. Culler, and R. A. Popa, "Wave: A decentralized authorization system for iot via blockchain smart contracts," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2017-234, Dec 2017. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-234.html

[98] C. Liem, E. G. AbdAllah, C. Okoye, J. O'connor, M. S. U. Alam, and S. Janes, "Runtime self-protection in a trusted blockchain-inspired ledger," 11 2017.

[99] P. Schwabe and A. Hülsing, "Post-quantum blockchain using one-time signature chains," 2018.

[100] Y. Gao, X. Chen, Y. Chen, Y. Sun, X. Niu, and Y. Yang, "A secure cryptocurrency scheme based on post-quantum blockchain," *IEEE Access*, vol. 6, pp. 27 205–27 213, 2018.

[101] D. Aggarwal, G. K. Brennen, T. Lee, M. Santha, and M. Tomamichel, "Quantum attacks on Bitcoin, and how to protect against them," *arXiv preprint arXiv:1710.10377*, 2017.

[102] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for iot security and privacy," *CoRR*, vol. abs/1712.02969, 2017. [Online]. Available: http://arxiv.org/abs/1712.02969

[103] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, May 2018.

[104] B. David, P. Gaži, A. Kiayias, and A. Russell, "Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain," in *Advances in Cryptology – EUROCRYPT 2018*, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 66–98.

[105] T. Hanke, M. Movahedi, and D. Williams, "DFINITY technology overview series, consensus system," *CoRR*, vol. abs/1805.04548, 2018. [Online]. Available: http://arxiv.org/abs/1805.04548

[106] B. Cohen, "Chia network," 2017. [Online]. Available: https://www.chia.net

[107] F. Charlon, "Openchain." [Online]. Available: https://www.openchain.org/

[108] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," in *Advances in Cryptology – ASIACRYPT 2018*, T. Peyrin and S. Galbraith, Eds. Cham: Springer International Publishing, 2018, pp. 435–464.

[109] D. Chaum, *Blind Signature System*. Boston, MA: Springer US, 1984, pp. 153–153.

[110] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in *Financial Cryptography and Data Security*, M. Brenner, N. Christin, B. Johnson, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 112–126.

[111] E. Heilman, F. Baldimtsi, and S. Goldberg, "Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 43–60.

[112] F. Zhang and K. Kim, "Id-based blind signature and ring signature from pairings," in *Advances in Cryptology — ASIACRYPT 2002*, Y. Zheng, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 533–547.

[113] S. Meiklejohn and R. Mercer, "Möbius: Trustless Tumbling for Transaction Privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 2, 2018.

[114] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "Coinparty: Secure multi-party mixing of bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*. New York, NY, USA: ACM, 2015, pp. 75–86.

[115] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, Dec 1994. [Online]. Available: https://doi.org/10.1007/BF00195207

[116] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, 2014, pp. 781–796. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/ben-sasson

[117] J. P. Morgan, "Quorum," 2016. [Online]. Available: https://github.com/jpmorganchase/quorum

[118] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, Sept 1994.

[119] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, Aug 2016, pp. 25–30.

[120] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *Journal of Medical Systems*, vol. 40, no. 10, p. 218, Aug 2016. [Online]. Available: https://doi.org/10.1007/s10916-016-0574-6

[121] A. Outchakoucht, E. Hamza, and J. P. Leroy, "Dynamic access control policy based on blockchain and machine learning for the internet of things," *Int. J. Adv. Comput. Sci. Appl*, vol. 8, no. 7, pp. 417–424, 2017.

[122] S. H. Hashemi, F. Faghri, and R. H. Campbell, "Decentralized user-centric access control using pubsub over blockchain," *CoRR*, vol. abs/1710.00110, 2017. [Online]. Available: http://arxiv.org/abs/1710.00110

[123] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 618–623.

[124] S. Tahir and M. Rajarajan, "Privacy-Preserving Searchable Encryption Framework for Permissioned Blockchain Networks," in *2018 IEEE Proceedings of iThings, GreenCom, CPSCom and SmartData*. IEEE, 2018, pp. 1628–1633.

[125] D. J. Bernstein, "CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness," 2014.

[126] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey, "Towards better availability and accountability for iot updates by means of a blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, April 2017, pp. 50–58.

[127] M. Luongo and C. Pon, "The Keep Network: A Privacy Layer for Public Blockchains," KEEP Network, Tech. Rep., 2018. [Online]. Available: https://keep.network/whitepaper

[128] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979. [Online]. Available: http://doi.acm.org/10.1145/359168.359176

[129] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *2018 Information Theory and Applications Workshop (ITA)*, Feb 2018, pp. 1–6.

[130] B. Franca, "Homomorphic mini-blockchain scheme," 2015.

[131] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology — CRYPTO '91*, J. Feigenbaum, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 129–140.

[132] T. Ruffing and G. Malavolta, "Switch commitments: A safety switch for confidential transactions," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 170–181.

[133] M. O. Rabin, "How to exchange secrets with oblivious transfer." *IACR Cryptology ePrint Archive*, vol. 2005, p. 187, 2005.

[134] D. C. Sánchez, "Raziel: Private and verifiable smart contracts on blockchains," *CoRR*, vol. abs/1807.09484, 2018. [Online]. Available: http://arxiv.org/abs/1807.09484

[135] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 584–597. [Online]. Available: http://doi.acm.org/10.1145/1315245.1315317

[136] F. Armknecht, J.-M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 831–843. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660310

[137] D. J. Bernstein, *Introduction to post-quantum cryptography*. Springer, 2009, pp. 1–14.

[138] A. Ekert and R. Jozsa, "Quantum computation and shor's factoring algorithm," *Reviews of Modern Physics*, vol. 68, no. 3, p. 733, 1996.

[139] K. Chalkias, J. Brown, M. Hearn, T. Lillehagen, I. Nitto, and T. Schroeter, "Blockchained post-quantum signatures," *IACR Cryptology ePrint Archive*, vol. 2018, p. 658, 2018.

[140] O. Regev, "Lattice-based cryptography," in *Annual International Cryptology Conference*. Springer, 2006, pp. 131–141.

[141] J. Ding and B.-Y. Yang, *Multivariate public key cryptography*. Springer, 2009, pp. 193–241.

[142] R. Overbeck and N. Sendrier, *Code-based cryptography*. Springer, 2009, pp. 95–145.

[143] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395 – 411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17315765

[144] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, Oct 1999, pp. 120–130.

[145] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, Eds. Cham: Springer International Publishing, 2017, pp. 297–315.

[146] A. S. de Pedro Crespo, D. Levi, and L. I. C. García, "Witnet: A decentralized oracle network protocol," *CoRR*, vol. abs/1711.09756, 2017. [Online]. Available: http://arxiv.org/abs/1711.09756

[147] Bitcoin Org, "Pay to script hash," https://en.bitcoin.it/wiki/Pay_to_script_hash, 2012, [Online; accessed 8-Jun-2019].

[148] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, July 2018.

[149] A. Binstock, "Obfuscation: Cloaking your Code from Prying Eyes," *DevX.com*, 2003. [Online]. Available: https://web.archive.org/web/20080420165109/http://www.devx.com/microsoftISV/Article/11351

[150] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 47–53.

[151] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology — CRYPTO 2001*, J. Kilian, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 213–229.

[152] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 440–456.

[153] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006, pp. 89–98.

[154] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 568–588.

[155] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Functional encryption: New perspectives and lower bounds," in *Annual Cryptology Conference*. Springer, 2013, pp. 500–518.

[156] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," in *Advances in Cryptology – CRYPTO 2018*, H. Shacham and A. Boldyreva, Eds. Cham: Springer International Publishing, 2018, pp. 757–788.

[157] B. Bünz, S. Goldfeder, and J. Bonneau, "Proofs-of-delay and randomness beacons in ethereum," *IEEE Security and Privacy on the blockchain (IEEE S&B)*, 2017.

[158] S. Kumar, E. Rosnes, and A. G. i Amat, "Private information retrieval in distributed storage systems using an arbitrary linear code," in *2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 1421–1425.

[159] S. Chow, P. Eisen, H. Johnson, and P. C. Van Oorschot, "White-box cryptography and an aes implementation," in *International Workshop on Selected Areas in Cryptography*. Springer, 2002, pp. 250–270.

[160] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental cryptography: The case of hashing and signing," in *Annual International Cryptology Conference*. Springer, 1994, pp. 216–233.

[161] H. Mihajloska, D. Gligoroski, and S. Samardjiska, "Reviving the idea of incremental cryptography for the zettabyte era use case: incremental hash functions based on sha-3," in *International Workshop on Open Problems in Network Security*. Springer, 2015, pp. 97–111.

[162] C. Delerablée, "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys," in *Advances in Cryptology – ASIACRYPT 2007*, K. Kurosawa, Ed. Springer Berlin Heidelberg, 2007, pp. 200–215.

[163] A. Ge and P. Wei, "Identity-based broadcast encryption with efficient revocation," Cryptology ePrint Archive, Report 2019/038, 2019, https://eprint.iacr.org/2019/038.

[164] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013. [Online]. Available: http://dx.doi.org/10.1007/s13389-013-0057-3

[165] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "Bsein: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *Journal of Network and Computer Applications*, vol. 116, pp. 42 – 52, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804518301619

[166] R. Goyal and V. Goyal, "Overcoming cryptographic impossibility results using blockchains," in *Theory of Cryptography*, Y. Kalai and L. Reyzin, Eds. Cham: Springer International Publishing, 2017, pp. 529–561.

[167] A. Scafuro, L. Siniscalchi, and I. Visconti, "Publicly verifiable proofs from blockchains," in *Public Key Cryptography*, 2019.

[168] N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky, "Position based cryptography," in *Advances in Cryptology - CRYPTO 2009*, S. Halevi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 391–407.

[169] Notes on Bitcoin privacy technology, "ECDHM Address," *Open Bitcoin Privacy Project*, 2019. [Online]. Available: http://wiki.openbitcoinprivacyproject.org/topics: ecdhm-address