A Survey on Zero-Knowledge Proof with a Focus on Blockchain

Xinyi Chen Jia Zhao

Dec 2020

Abstract

This is a survey on zero-knowledge proof and its applications, with a focus on blockchain. We present the history and key concepts of zero-knowledge proof. We then investigate three applications and further focus on the mechanism of blockchain.

1 Introduction

There are many surveys on zero-knowledge proofs or the blockchain, but the main goal of this survey is to bridge the gap between those papers, discuss both topics and its relationship.

This paper starts with an introduction on zero-knowledge proof, with a focus on the applications, followed by a detailed discussion on the trending one, the blockchain, and its potentials.

In section 2, we explain the concept of zero-knowledge proof and present a brief related history. In section 3, we provide a few critical real-world applications of zero-knowledge proof. In section 4, we describe in detail one main application of zero-knowledge proof, that is, the blockchain, with examples of implementations. Finally, we briefly discuss the current open problems in zero-knowledge proof and the blockchain.

2 Basics of Zero-Knowledge Proof

Zero-knowledge proofs (ZKPs) allow the prover to communicate with the verifier back and forth to prove that a statement is true, without revealing any information beyond the validity of the statement itself.

The concept of zero-knowledge was first introduced in the 1980's by Goldwasser, Micali, and Rackoff in their paper "The Knowledge Complexity of Interactive Proof-Systems"[13]. This paper proposed the Interactive Polynomial time (IP) hierarchy of interactive proof systems, which, together with the previous Arthur-Merlin (AM) protocol[6], forms the two main complexity classes describing interactive proof systems. This paper also conceived the concept of knowledge complexity, which measures the amount of knowledge perceived by the verifier during the proof. This paper further showed how to construct zero-knowledge systems from any NP-set, and demonstrated a concrete construction for deciding quadratic non-residues mod m, which lies in the intersection of NP and co-NP.

Given the proof system, Goldreich, Micali, and Wigderson showed a ZKP for the NP-complete 3-coloring problem, with the assumption that unbreakable encryption exists[12]. This assumption is necessary because the proof system requires encryption of information provided by the prover. If the

encryption can be broken with efficient efforts, both the prover and verifier can cheat. Since every problem in NP can be efficiently reduced to this NP-complete problem, all problems in NP have ZKPs under the same assumption. On top of this, they also showed that the graph non-isomorphism problem, which lies in co-NP as the complement of the graph isomorphism problem, has a ZKP.

Moreover, Impagliazzo and Yung showed that everything provable by an interactive proof is provable in zero-knowledge, also assuming unbreakable encryption[14]. In order words, based on the IP complexity class proposed by Goldwasser, Micali, and Rackoff, they showed that all problems in IP=PSPACE could be proved with zero-knowledge.

3 Applications of Zero-Knowledge Proof

In this section, we will briefly introduce three main applications of zero-knowledge proof, where security is attached great importance to and strictly required.

3.1 Authentication System

Prompted by the authentication systems where a prover wants to prove his or her identity to a verifier, studies have been done on how to authenticate while not revealing anything about the secret, such as the password, in the authentication systems. This type of zero-knowledge proof is considered as a "zero-knowledge proof of knowledge".

With the use of zero-knowledge proof, authentication can be done without revealing anything about privacy or secret information, which has shown the necessity and the advantage of embedding zero-knowledge proof into authentication systems.

According to the definition in IEEE P1363.2, zero-knowledge proof of knowledge (ZKPK) is a proof of knowledge that only proves the "validity of the assertion that the prover knows something", and zero-knowledge password proof (ZKPP) is basically an interactive zero-knowledge proof of knowledge of "a small secret"[4]. Zero-knowledge password proof is a special type of ZKPK that is based on password-derived information and data.

ZKPP is one important feature of a password-authenticated key exchange (PAKE) protocol. PAKE, first introduced by Bellovin and Merritt[8], is an authenticated key exchange protocol. It allows multiple parties (the prover and the verifiers) to agree on a shared key based on their same knowledge of a password. PAKE protects the prover's password by applying the zero-knowledge password proof. Namely, after any login attempt, either valid or invalid, the only thing learned by both parties is whether the prover's password matched the verifier's expected value or not.

PAKE prevents attacks from eavesdroppers. Unauthorized parties who do not possess the password will not be able to obtain the password through the communication process and cannot guess the password without interaction with parties engaged.

3.2 Nuclear Disarmament

In 2016, the Princeton Plasma Physics Laboratory and Princeton University demonstrated a possible way for future nuclear disarmament, the act of reducing or eliminating nuclear weapons. This technique, with a foundation on zero-knowledge proof, allows inspectors to confirm whether or not an object is a nuclear weapon without sharing or revealing any of the internal secret mechanisms.

The inherent problem with nuclear verification is that neither side wants to show their workings. If a team of scientists, maybe from the United States, were allowed unrestricted access to dismantle a Russian warhead, they could easily tell if the device was a real nuclear weapon. But no country

with nuclear weapons would ever allow this disclosure because it would expose some of the most closely guarded national secrets.

Professor Glaser, from Princeton University, is preparing the novel verification technique. His experiment, which mimics a potential situation if the U.S. and Russia agreed to verification on individual nuclear warheads, is an interactive back and forth game with many rounds between the two parties.

The back and forth communication involves a trusted reference item (the so-called "template") and another item that the inspected party offers for dismantlement, which is supposedly identical to the template. There is also a neutron source and a row of gel-filled tubes, which are sensitive to neutron radiation and, with increasing exposure, produce tiny bubbles that can be later counted. The mechanism can be abstracted as follows:

1. Two objects A and B are provided. The host, the prover in this mechanism, prepares two radiographic films with the inverse image of object A and hide them in two sealed envelopes.

2. The inspector, the verifier here, randomly assigns object A and B between two envelopes and a radiation source, respectively.

3. The objects are exposed to the radiations, which will cause fluence onto images in the envelopes. This process is essentially equivalent to adding a positive image on top of a negative image.

4. Both parties then unseal both envelops.

5. If both images are flat gray, meaning the positive images match the negative images and thus the items are the same, the verifier accepts the proof. Otherwise, the verifier rejects the proof.

The above idea is already in use by the Defense Advanced Research Projects Agency, or DARPA, who released a statement claiming that they were working on a new project called SIEVE, i.e., Securing Information for Encrypted Verification and Evaluation. This new project makes use of ZKP to determine the origin of highly secure data without any real revelation from the U.S. government.

3.3 Blockchain

The most interesting application with rapid development is blockchain. Details and applications of the blockchain will be discussed in section 4.

Here, we first introduce some basic concepts that this survey is based on.

Record A record is a data structure also known as a structure, a struct in C programming, or a compound data. It is a collection of fields that are of the same or different data types.

Merkle Tree A Merkle tree, also known as a hash tree, is a tree where each leaf node contains the cryptographic hash of a block of data, and every non-leaf node contains the cryptographic hash of the labels of its child nodes.

Cryptographic Hash Function A cryptographic hash function (CHF) is a mathematical algorithm that maps a message with an arbitrary length to a bit array of a fixed size, which is also called the "hash", "hash value", or "message digest". It is a one-way function, that is, a function which is easy to compute on every input in polynomial time, but is practically infeasible to invert.

Cryptocurrency A cryptocurrency is a network-based medium of exchange. Strong cryptography is used to verify and store the transaction data, and to control the creation of new coins. Cryptocurrencies generally do not exist in physical forms and are typically not issued by any central authorities.

Double-Spending Double-spending is a potential flaw in a digital cash scheme. It refers to the situation where the same single digital token can be spent more than once[11]. Different from physical currencies, a digital token is recorded in a digital file, which renders the possibility of duplication or falsification, thus leading to double-spending. Such double-spending will result in inflation by creating a new amount of copied currency that did not previously exist, which depreciates the currency and undermines the trust from users.

4 Blockchain and Related Applications

4.1 Introduction to Blockchain

The blockchain was invented in 2008 by a person or a group of people using the pseudonym Satoshi Nakamoto. Designed to function as the public transaction ledger of the cryptocurrency bitcoin, it is managed by a peer-to-peer (p2p) network[17].

In terms of the structure, a blockchain is a growing list of blocks, i.e. records, linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, a nonce, and the transaction data that is normally hashed in a Merkle Tree[18]. A blockchain is a public digital ledger that stores records of users' transactions and their identities anonymously. It is decentralized, meaning that it is not controlled by any central authority, and dispersed across multiple servers.

Nonce is the abbreviation for "number only used once". Nonce is the number that blockchain miners are solving for. Miners need to find a nonce that when appended to the previous hash, and rehashed together, the target hash is obtained. After such process, the block is added to the chain.

4.2 Blockchain and Zero-Knowledge Proof

First formalized by Markus Jakobsson and Ari Juels, proof-of-work (PoW) is where zero-knowledge proof is applied in blockchain. It is a form of cryptographic zero-knowledge proof where the prover demonstrates to the verifiers that a certain amount of computational work or effort has been expended in a specified interval of time for some purpose[15].

Hashcash is a cryptographic hash-based proof-of-work system proposed by Adam Back[7]. It is first used to limit email spam and denial-of-service attacks. In terms of email use, a hashcash stamp encoding is appended to the header of an email. To find such a header that satisfies the property, brute force is the only algorithm. This hashcash stamp proves that the sender has expended a certain amount of CPU time before sending the email. The receiver can verify the validity of the stamp with little computational cost. Based on the hypothesis that spammers usually value the profits greatly and are unlikely to send emails with hashcash stamps if it costs much, emails with hashcash stamps are less likely to be spam. Therefore, this algorithm can be used to limit email spam.

The proof-of-work implemented in the blockchain prompted by Nakamoto is similar to Adam Back's Hashcash. It will scan for such a value that when hashed, for example with SHA-256, the hash begins with a number of zero bits.

According to the blockchain introduced by Satoshi Nakamoto [17], when new transactions are generated, the network broadcasts the information to all nodes. Each node will then collect the new transactions into a block, and works on a hard proof-of-work for the block. Once a proof-of-work is found at any node, the block is confirmed and broadcast to every node in the network. After checking all transactions in the block are valid and the coins are not already spent, the nodes receiving the information will accept the block by using the hash of the accepted block as the previous hash to work on future blocks.

With such a design, modification of data on a blockchain is highly unlikely. Once recorded in the blockchain, any attackers cannot modify the data in a certain block without changing the data in all the following blocks. If the honest nodes in the system control a majority of the CPU power, the honest chain will grow faster than any competing chains produced by the attacker nodes. In order to modify one block in the chain, the proof-of-work of the block and all subsequent blocks need to be redone, and the attacker has to catch up with and surpass the work of other honest nodes.

As pointed out by Satoshi Nakamoto, the first transaction in a block starts a new coin owned by the creator of the block[17]. This not only offers a way to distribute new coins into circulation, but also provides an incentive for nodes to support the network. Nodes expending CPU time and electricity to distribute new coins into circulation is analogous to resources expended by gold miners to find new gold.

It is noteworthy that this incentive also helps reduce malicious nodes. With enough CPU power to falsify the blocks, it becomes more appealing and more profitable for nodes to behave honest and mine new coins rather than going against the rules, undermining the system and giving away possible wealth.

4.3 Advantages and Disadvantages of Zero-Knowledge Proof in Blockchain

Recall that zero-knowledge proof, as we discussed above, can guarantee different parties' information without revealing their true identities. Blockchains using ZKP have five main advantages and one disadvantage. Such mechanisms reveal no important data between communications, protect data from criminals, ensure the validity of the entire network, and further keep online payments and transactions safe. Since ZKP requires no password during communications, especially transactions, it also replaces the risky nature of password-only authentication, and secures public cloud accounts.

One big disadvantage then comes. Since all the communication and the data are only accessible via a master password, if the user forgets or loses the master password, all the data is forever lost.

4.4 Real-World Applications

4.4.1 Bitcoin

In the paper "Bitcoin: A Peer-to-Peer Electronic Cash System" where blockchain was first introduced, Satoshi Nakamoto proposed the bitcoin system and the cryptocurrency Bitcoin (₿)[17]. Transactions of the bitcoin are recorded in the distributed ledger, blockchain, proposed by Nakamoto. The implementation of bitcoin was released as an open-source software in 2009. The bitcoin network was created then when Nakamoto mined the starting block of the chain, which is known as the genesis block.

The transaction fee in the bitcoin system is optional, but the bitcoin protocol allows a transaction to leave a transaction fee for the miner. If the value paid out of a transaction (in bitcoins) is less than the amount put in, the difference is treated as a transaction fee that can be collected by whoever manages to mine a block containing that transaction[16]. Miners are able to choose which

transactions to process and they can give priority to the ones that pay higher transaction fees with regard to storage.

In the blockchain, bitcoins are registered to bitcoin addresses. Computation of a bitcoin address based on a valid private key can be done easily, while given a bitcoin address, it is unfeasible to compute the corresponding private key. One has to know the private key and digitally sign the transaction to use the bitcoins. The network verifies the signature using the public key, and the private key is never revealed[5]. If the private key is lost, the only evidence of ownership for bitcoins is lost[10]. The coins are therefore unusable, and effectively lost.

4.4.2 Zcash

Zcash is a privacy-protecting digital currency, or a cryptocurrency, like Bitcoin. In fact, Zcash was built on the original Bitcoin code base. As we've briefly discussed in 3.3, a major difference between most cryptocurrencies and traditional money is that cryptocurrencies are network-based and managed on decentralized public blockchains, instead of by a centralized bank. This means you truly own your money and your holdings with full transaction history are exposed to everyone, such as auditors.

Zcash has three important features like most cryptocurrencies.

(1) Zcash provides a fast and reliable, inexpensive means for digital transactions. The transaction fee is 0.0001 Zcash, which is only $0.0072, provided that the current Zcash price is $72.

(2) Zcash is audit- and regulation-friendly because users can disclose their addresses and full transaction data to third-party for auditing or regulations.

(3) Zcash is decentralized and attack-resistant. Built upon network rather than physical centralized authorities, Zcash and all other cryptocurrencies are decentralized and maintained by a wide network. Then, there is no central database to be hacked and the entire network cannot be shut down by any centralized authority.

Zcash goes beyond most cryptocurrencies in which full transaction history has to be exposed based on private, or shielded, addresses and transactions. Most cryptocurrencies only have transparent addresses that are publicly visible on their blockchain. On the contrary, private addresses and transactions allow people to send and receive Zcash without disclosing the sender, receiver or the amount transacted. Shielded addresses are not visible and transactions between shielded addresses do not reveal either addresses, the transaction amount or the contents of the encrypted memo field. An encrypted memo field allows the sender to include relevant information, such as instructions, to the receiver, in a completely encrypted version. Owners of shielded address can disclose transaction details on their need, but the sender and the receiver have access to different transaction details. Both the sender or receiver of a transaction can disclose transaction-specific details as needed by a 3rd party, auditing for example. However, the receiver can only disclose all transaction histories and data on the encrypted memo field, but not the sender's address unless the corresponding identifying information is included in the memo field. According to Zcash official website, it is a temporary experimental feature for receivers having no access to the sender's address. Full viewing keys will be soon supported to reveal all transaction values in and out of the address.

We present a comparison between Zcash and other currencies in Fig.1.

The strong privacy of the shielded transactions in Zcash is guaranteed because all shielded transactions can be verified as valid under the network's consensus rules by using zk-SNARK proofs, an acronym for Zero-Knowledge Succinct Non-Interactive Argument of Knowledge. Zero-knowledge proofs, as what we learned from the class, allow the prover to communicate with the verifier back

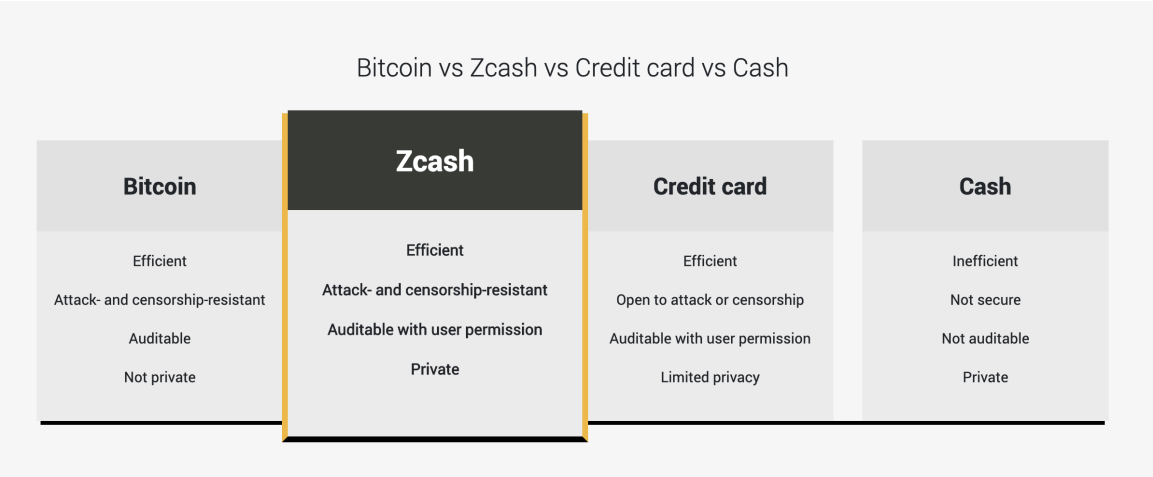

Figure 1: Bitcoin vs Zcash vs Credit card vs Cash[2]

and forth to prove that a statement is true, without revealing any information beyond the validity of the statement itself. The succinct version of zero-knowledge proofs can be verified within a few milliseconds, with a proof length of only a few hundred bytes even for very large programs. In the succinct version, the proof is non-interactive and consists only a single message sent from the prover to the verifier so that they do not have to communicate back and forth. Currently, the most efficient way to produce efficient and non-interactive zero-knowledge proofs is to have an initial setup phase that generates a common reference string shared between prover and verifier. If someone had access to the secret randomness used to generate these parameters, they would be able to forge proofs that would look valid to the verifier, which means that the malicious party could create fake coins in Zcash. To prevent this from happening, Zcash generated the public parameters through an elaborate, multi-party ceremony. The comprehensive protocol is very dense in math and will be temporarily skipped. For more info, refer to the paper "Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model" [9].

Combining transparent and shielded transaction types together, we have four different transaction types. Zcash addresses are either private or transparent, and private z-addresses start with a "z" with transparent t-addresses starting with a "t".

A Z-to-Z transaction appears on the public blockchain. The public can know that such a transaction happened and the fees were paid. But the addresses, transaction amount and the memo field are all encrypted and not publicly visible. This type of encryption is only possible through the use of zk-SNARKs as discussed above.

A T-to-T transaction works just like Bitcoin and other traditional cryptocurrencies: the sender, receiver and transaction value appears on the public blockchain and are publicly visible. Zcash suggests that while many exchanges exclusively use t-addresses today, many are moving to shielded addresses for better user privacy [3].

It is also noted that the two Zcash address types are interoperable, meaning that one party can use z-address while the other uses t-address. All four types are presented in Fig.2.

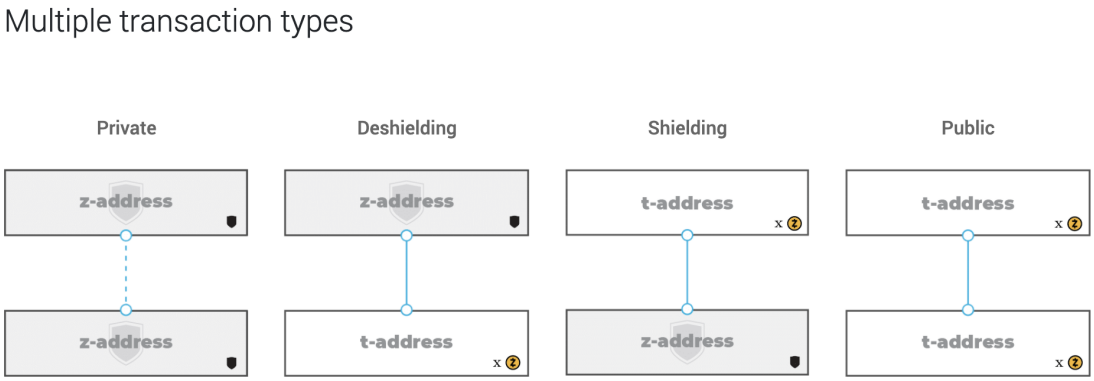

Figure 2: Zcash Transaction Types[3]

4.4.3 ThunderCore

We now briefly introduce a state-of-the-art blockchain designed by a team of computer scientists, led by Rafael Pass at Cornell Tech. ThunderCore uses PaLa consensus algorithm, mathematically proved to be secure, has been shown to be simpler and faster than any other consensus algorithm including Hotstuff, Tendermint, FBFT, and FBA. All the traditional algorithms allow all nodes to compete to solve hard cryptographic puzzles in order to extend the blockchain, which are slow because data must propagate through the entire network. They are also super time- and energy-consuming. For example, bitcoin mining consumes more electricity a year than Ireland [reference from the Guardian].

PaLa, the algorithm used by ThunderCore, solves the above problems by allowing only one node to do the computations. The full algorithm can be found in the whitepaper [1]. We here present the rough idea. The elected node is proposed by a committee of nodes in order to ensure honest behaviors. The primary proposer rapidly creates new blocks and sends it over a high speed network connection to all voters. The voters then respond with their votes. When 2/3 of all the votes are received, the block is notarize. When there are two notarized blocks in a row, the second block becomes confirmed, meaning that it never changes afterwards, and becomes part of the immutable blockchain history.

PaLa is mathematically proved by researchers to be 100× faster than traditional proof-of-work algorithms, as we introduced in the blockchain basics, and has sub-second confirmation times with very high throughput.

However, ThunderCore also encounters a vital criticism that may undermine its essential status as a blockchain. Some researchers argue that a mechanism with such a voting algorithm is not totally decentralized because the voting algorithm can be designed maliciously. Then the mechanism is indeed centralized and controlled by the algorithm designers. While this problem is open for further discussion, we here believe that this may be over-concerned because the voting algorithm is open source for checking and updating. If the algorithm is not under frequent modification, it is a set of rules but not an authority. Moreover, the digital ledger is still kept by everyone, and new coins are generated by individuals rather than distributed.

5 Conclusion

This survey investigates several state-of-the-art research with a theoretical foundation on zero-knowledge proof and fills the gap between theoretical concepts of zero-knowledge proof and its current wide applications. Most applications are in the financial area. Besides financial applications, future research can explore more possibilities in supply chain management, healthcare such as MedRec developed by MIT, and maybe digital voting. Moreover, since it would be a great loss if one's hard disk storing his master password is destroyed, and that password links to hundreds of bitcoins. Therefore, a more notable issue, as we have discussed, is how we can ensure the safety of the users' master passwords to the blockchain.

References

[1] Thundercore whitepaper. Tech. rep. [Accessed: 2020].

[2] Zcash basics. Tech. rep. [Accessed: 2020].

[3] Zcash technology. Tech. rep. [Accessed: 2020].

[4] Ieee standard specification for password-based public-key cryptographic techniques. *IEEE Std 1363.2-2008* (2009), 1–140.

[5] Antonopoulos, A. M. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, 1st ed. O'Reilly Media, Inc., 2014.

[6] Babai, L., and Szemerédi, E. On the complexity of matrix group problems i. In *FOCS* (1984).

[7] Back, A. Hashcash - a denial of service counter-measure.

[8] Bellovin, S. M., and Merritt, M. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy* (1992), pp. 72–84.

[9] Bowe, S., Gabizon, A., and Miers, I. Scalable multi-party computation for zk-snark parameters in the random beacon model. https://eprint.iacr.org/2017/1050.

[10] Böhme, R., Christin, N., Edelman, B., and Moore, T. Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives 29*, 2 (May 2015), 213–38.

[11] Chohan, U. The double spending problem and cryptocurrencies. *SSRN Electronic Journal* (01 2017).

[12] Goldreich, O., Micali, S., and Wigderson, A. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)* (1986), pp. 174–187.

[13] Goldwasser, S., Micali, S., and Rackoff, C. The knowledge complexity of interactive proof systems. *SIAM J. Comput. 18*, 1 (1989), 186–208.

[14] Impagliazzo, R., and Yung, M. Direct minimum-knowledge computations. vol. 293, pp. 40–51.

[15] Jakobsson, M., and Juels, A. *Proofs of Work and Bread Pudding Protocols(Extended Abstract)*. Springer US, Boston, MA, 1999, pp. 258–272.

[16] Kroll, J. A., Davey, I. C., and Felten, E. The economics of bitcoin mining, or bitcoin in the presence of adversaries.

[17] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system.

[18] Narayanan, A., Bonneau, J., Felten, E. W., Miller, A., and Goldfeder, S. *Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction.* Princeton University Press, 2016.