ZERO KNOWLEDGE PROOF: ORIGIN, THEORIES, AND APPLICATIONS

Di Kevin Gao

Mississippi State University,
Computer Science & Engineering, {dkg163}@msstate.edu

ABSTRACT

Zero-knowledge Proof (ZKP) is a cryptographic technology that allows the verification of the truth of information without revealing the information itself. It is a technology that has wide applications in privacy protection, authentications, Blockchain, cryptocurrencies, and Machine Learning. Our paper will delve into Zero-Knowledge Proof, elaborate on the interactive proof system, trace its theoretical development, and examine its wide range of applications.

***Index Terms*—** Zero Knowledge Proof, Zero-knowledge Proof, Zk-Proof, ZKP, Zero Knowledge, Machine Learning

1. INTRODUCTION

Zero-knowledge Proof (ZKP) is a cryptographic technique enabling one party, the prover, to persuade another party, the verifier, of the possession of specific knowledge or information without disclosing its precise content. Initially conceptualized in 1985 by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in their seminal paper “The Knowledge Complexity of Interactive Proof-Systems”, Zero-knowledge Proof was defined as “proofs that impart no supplementary knowledge beyond the validity of the proposition at hand.” [1] It has a wide range of applications in areas such as privacy protection, authentication, Blockchain, cryptocurrencies, and Machine Learning.

In the following sections, we will introduce interactive Turing machines, basic principles of Zero-knowledge Proof, and the Interactive Proof System.

1.1. Interactive Turing Machines

In this section, we will leverage Turing Machines, a staple of Computational Theory, and expand their capabilities to emulate both a prover and a verifier. We establish a framework for interactive machines as follows:

- Each Turing machine is equipped with randomness, possessing multiple tapes including the input, output, and random tape.
- Mutual tape access is not allowed between the machines.
- To facilitate communication, communication tapes (CA and CB) are introduced. Turing machine A reads CA and writes to CB, while Turing machine B reads CB and writes to CA.
- Both Turing machines can access additional switch states, denoted as S. When a transition activates the state, the current machine pauses, allowing execution to shift to the other machine, and vice versa.
- The interconnected pair of randomized Turing machines terminates upon the termination of either machine.

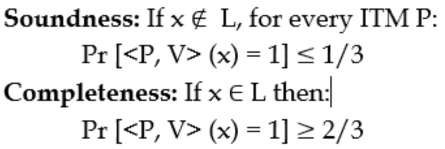
1.2. The Basic Principles

Zero-knowledge Proofs rely upon three foundational principles: completeness, soundness, and zero-knowledge:

- Completeness: This principle ensures that any valid proof presented will unfailingly be accepted by the verifier.
- Soundness: This principle guarantees that an invalid proof will be reliably rejected by the verifier. If the prover lacks the requisite knowledge or information, they cannot successfully persuade the verifier.
- Zero-knowledge: Perhaps the most captivating facet of Zero-knowledge Proofs is their capacity to convey information without revealing any additional content.

1.3. Interactive Proof System

With the groundwork laid by interactive Turing machines and principles, we now introduce Interactive Proofs: A pair of interactive Turing machines, P (prover) and V (verifier), constitute an interactive proof system for a language L if V operates in polynomial time, and the conditions outlined in Figure 1 are satisfied:



Soundness: If $x \notin L$, for every ITM P:

Pr [<P, V> (x) = 1] ≤ 1/3

Completeness: If $x \in L$ then:

Pr [<P, V> (x) = 1] ≥ 2/3

Fig. 1: Definition of Interactive Proof System

In the definition, note that only the verifier is required to be computationally bound, while the prover has no such limit.

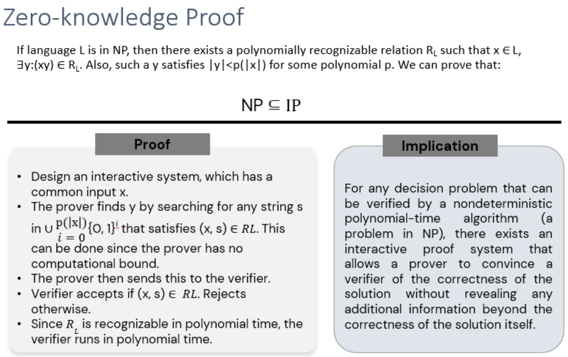
1.4. Any language in NP has an interactive proof system

Based on the definition provided earlier, we can demonstrate that any language within the class NP possesses an interactive proof system.

Let IP denote the collection of all languages equipped with an interactive proof system. If a language L belongs to NP, then there exists a polynomially recognizable relation R_L such that $x \in L \Leftrightarrow \exists y : (x, y) \in R_L$. Also, such a y satisfies $|y| < p(|x|)$ for some polynomial p [2].

We have devised the following interactive protocol, utilizing a common input x. First of all, the prover finds such y. This can be done by searching for any string s in $U^{p(|x|)}$ such that $(x, s) \in R_L$. Since the prover has no computational bound, this can be done without issues. The prover then sends this s to the verifier. On receiving message s, the verifier accepts if $(x, s) \in R_L$ and rejects otherwise. Since R_L is recognizable in polynomial time, the verifier runs in polynomial time as required.

We can verify completeness: if the probability is 1, then the verifier will accept. To prove soundness, consider the following: since $x \notin L \Rightarrow \forall y : (x, y) \notin R_L$, then the verifier will never accept if the probability is approaching 0. This implies that the above is an interactive proof system for L. Since L was an arbitrary element of NP, then $NP \subseteq IP$. Therefore, any language in NP should have an interactive proofing system. [3]. For details, please refer to Figure 2.



Zero-knowledge Proof

If language L is in NP, then there exists a polynomially recognizable relation R_L such that $x \in L$, $\exists y:(xy) \in R_L$. Also, such a y satisfies $|y|<p(|x|)$ for some polynomial p. We can prove that:

$NP \subseteq IP$

Proof

- Design an interactive system, which has a common input x.
- The prover finds y by searching for any string s in $\cup_{i=0}^{p(|x|)} \{0, 1\}^i$ that satisfies $(x, s) \in RL$. This can be done since the prover has no computational bound.
- The prover then sends this to the verifier.
- Verifier accepts if $(x, s) \in RL$. Rejects otherwise.
- Since R_L is recognizable in polynomial time, the verifier runs in polynomial time.

Implication

For any decision problem that can be verified by a nondeterministic polynomial-time algorithm (a problem in NP), there exists an interactive proof system that allows a prover to convince a verifier of the correctness of the solution without revealing any additional information beyond the correctness of the solution itself.

Fig. 2: Any decision problem solvable by a NP-time algorithm, there exists an interactive proof system

This proof holds significant importance in computational complexity theory. It asserts that for any decision problem solvable by a nondeterministic polynomial-time algorithm (a problem in NP), there exists an interactive proof system enabling a prover to persuade a verifier of the solution's correctness without disclosing any further information beyond the solution's validity itself.

1.5. Example: Hamiltonian cycle for a large graph

The following example is based on Manuel Blum and his 1986 work ”How to Prove a Theorem So No One Else Can Claim It” [4].

In this scenario, Peggy possesses knowledge of a Hamiltonian cycle for a large graph G, while Victor only knows G but not the cycle itself. Peggy aims to demonstrate her knowledge of the cycle to Victor without directly revealing it. Finding a Hamiltonian cycle in a large graph is a problem known to be NP-complete.

To establish Peggy's familiarity with this Hamiltonian cycle, she and Victor engage in multiple rounds of a game:

- At the onset of each round, Peggy constructs H, a graph that is isomorphic to G.
- Peggy commits to H by numbering its vertices. Subsequently, for each edge of H, she records the vertices it connects on separate pieces of paper, placing them face down on a table to demonstrate her inability to alter their contents.
- Victor has the option to request either the demonstration of isomorphism between H and G or proof of a Hamiltonian cycle in H.
- If Victor requests the demonstration of isomorphism, Peggy unveils H by revealing all pieces of paper on the table, followed by providing the vertex translations that establish the isomorphism between G and H. Victor can then verify their isomorphism.
- If Victor asks for proof of Peggy's knowledge of a Hamiltonian cycle in H, she translates her Hamiltonian cycle in G onto H and uncovers only the edges pertaining to this cycle. This enables Victor to confirm the presence of a Hamiltonian cycle in H.

If Peggy possesses knowledge of a Hamiltonian cycle in G, she can fulfill Victor's request by presenting either the initial commitment to graph isomorphism, yielding H from G, or a Hamiltonian cycle in H derived through the isomorphism applied to the cycle in G.

Peggy's responses are meticulously crafted to obscure the original Hamiltonian cycle in G. Consequently, Victor gains no insight into the specific Hamiltonian cycle in G based on the information revealed in each round.

If Peggy does not know the information, her chance of fooling Victor is 2^{-n}, where n is the number of rounds. For all realistic purposes, it is difficult to defeat a Zero-knowledge Proof with a reasonable number of rounds in this way.

2. ZERO-KNOWLEDGE PROOF'S ORIGIN AND DEVELOPMENT

In this section, we first cover the theoretical development and then the practical protocols.

2.1. Theoretical Developments

Zero-knowledge Proof was first conceived in 1985 by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in their paper "The Knowledge Complexity of Interactive Proof-Systems" [1]. "Usually, a proof of a theorem contains more knowledge than the mere fact that the theorem is true", said the authors[5]. The authors further defined Zero-knowledge Proofs as "those proofs that convey no additional knowledge other than the correctness of the proposition in question". Examples of Zero-knowledge Proof systems are given for the languages of quadratic residuosity and quadratic nonresiduosity. It has both an NP and a co-NP algorithm, and so lies in the intersection of NP and co-NP [5]. Their paper introduced the IP hierarchy of interactive proof systems and conceived the concept of knowledge complexity, a measurement of the amount of knowledge about the proof transferred from the prover to the verifier.

In December 1985, Babai published in the proceedings of the seventeenth annual ACM symposium on Theory of Computing an article titled "Trading group theory for randomness". In the article, Babai defined "a new hierarchy of complexity classes AM(k) 'just above NP', introducing Arthur vs. Merlin games.". He proved that in spite of the game's "analogy with the polynomial-time hierarchy, the finite levels of this hierarchy collapse to AM=AM(2). Using a combinatorial lemma on finite groups." They "construct a game by which the nondeterministic player (Merlin) is able to convince the random player (Arthur) about the relation [G]=N provided Arthur trusts conclusions based on statistical evidence" He proved that AM consists precisely of those languages.

In 1987, the Fiat-Shamir heuristic introduced a simple identification and signature schemes that enable any user to prove his identity and the authenticity of his messages to any other user without shared or public keys. The schemes are provably secure against any known or chosen message attack if factoring is difficult, and typical implementations require only 1% to 4% of the number of modular multiplications required by the RSA scheme. Due to their simplicity, security, and speed, these schemes are ideally suited for microprocessor-based devices such as smart cards, personal computers, and remote control systems [6]

In 1988, Russell Impagliazzo and Moti Yung showed that, also assuming one-way functions or unbreakable encryption, there exist Zero-knowledge Proofs for all problems in IP = PSPACE, or in other words, anything that can be proved by an interactive proof system can be proved with zero knowledge [7].

In 1991, Oded Goldreich, Silvio Micali, and Avi Wigderson presented further evidence for the existence of Zero-knowledge Proof systems within NP in their paper titled "Proofs that Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems". This seminal work establishes a groundbreaking finding, demonstrating that every language within the complexity class NP (Nondeterministic Polynomial time) possesses a Zero-knowledge Proof system under the assumption of unbreakable encryption. This implies that for any problem whose solutions can be efficiently verified, there exists a protocol enabling a prover to convince a verifier of the solution's validity without disclosing any additional information beyond its correctness. Given that every problem in NP can be efficiently reduced to this underlying problem, it follows that, under this assumption, all problems within NP have Zero-knowledge Proofs available. Here are additional discussions of their findings:

Universal Applicability: The paper's findings extend to all NP languages, covering a broad spectrum of computational problems with efficient verification algorithms. This universality underscores the versatile utility of Zero-knowledge Proofs, offering potential enhancements to security and privacy across diverse real-world scenarios.

Practical Implications: Beyond its theoretical significance, the paper holds practical relevance for cryptography and secure computation. The existence of Zero-knowledge Proof systems for NP problems establishes a theoretical framework for devising cryptographic protocols that uphold privacy, confidentiality, and integrity in distributed systems and cryptographic applications [8].

Graph Non-isomorphism Problem: Additionally, the authors demonstrated the existence of a Zero-knowledge Proof for the graph nonisomorphism problem, the complement of the graph isomorphism problem. While this problem resides in co-NP, it remains unclassified in NP or any practical class.

In 1997, Jan Camenisch and Markus Stadler introduce the concept of group signatures with efficient Zk-Proofs, allowing members of a group to anonymously sign messages while maintaining the integrity of the group [9].

The above paved the way for the development of Zk-Proof protocols. Zk-Proof protocols, especially the non-interactive Zk-Proof protocols, made applications easier. In the next section, we will cover the development in that area.

2.2. Non-interactive Zero-knowledge Proofs

Non-interactive Zero-knowledge Proofs are gaining traction due to their ability for the prover alone to authenticate information to the verifier. This feature proves beneficial in scenarios where direct interaction between the two parties is challenging, such as online transactions where real-time communication is impractical. This attribute holds particular appeal for Blockchain and cryptocurrencies, as transaction verification is decentralized across a network of nodes. Furthermore, non-interactive Zk-Proofs offer scalability advantages, allowing for the simultaneous processing of numerous transactions without being hindered by multiple rounds of interactions between provers and verifiers.

In 1988, Blum, Feldman, and Micali demonstrated that computational zero-knowledge could be achieved without interaction by utilizing a shared common reference string between the prover and verifier. Initially, it was conceptualized as a single theorem-proof system, where each proof necessitated its own unique common reference string. However, a common reference string is not inherently random; it may comprise randomly selected group elements utilized by all parties involved in the protocol. Despite the randomness of the group elements, the reference string itself possesses a discernible structure, such as group elements, which distinguishes it from a truly random string. [10]

In 1991, Feige, Lapidot, et al., introduced multi-theorem zero-knowledge proofs as a more versatile notion for non-interactive Zk-Proofs. [11]

2.2.1. Succinct Non-Interactive ARguments of Knowledge (SNARK)

In 2012, Alessandro Chiesa and colleagues introduced the zk-SNARK protocol [12]. This innovation found its initial widespread application in the blockchain protocol, where zero-knowledge cryptography serves as the computational foundation, enabling mathematical proofs of possession without divulging the actual information.

In 2013, Ian Miers and colleagues introduced Zerocoin, a cryptographic enhancement for Bitcoin. This extension enhances the protocol to enable fully anonymous currency transactions, with the goal of enhancing privacy through the use of zk proofs. [13]

2.2.2. Bulletproofs

In 2018, Bunz and colleagues unveiled Bulletproofs through their article “Bulletproofs: Short Proofs for Confidential Transactions and More”[14]. They demonstrated that verifying a committed value’s range requires only a logarithmic number of field and group elements (relative to the range’s bit length). Subsequently, Bulletproofs found implementation in the Mimblewimble protocol.

2.2.3. Scalable Transparent ARgument of Knowledge (STARK)

In 2018, Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev introduced the STARK protocol [15]. This protocol offers transparency, with no need for a trusted setup, along with quasi-linear proving time and poly-logarithmic verification time. Unlike SNARK, STARK does not necessitate a trusted setup, rendering it highly beneficial for decentralized applications such as blockchains. Moreover, zk-STARKs have the capability to verify multiple statements concurrently, making them scalable and efficient.

There are other protocols such as Verifiable Polynomial Delegation (VPD) and Succinct Non-interactive ARGuments (SNARG).

3. ZK-PROOF AND APPLICATIONS

Zero-knowledge Proofs have a wide range of applications in enhancing privacy, guarding privacy, authenticating passwords, proving humanity, complementing Blockchain and cryptocurrencies, and supporting verifiable and secure machine learning.

- **Improving privacy and preserving integrity in digital identity systems:** Zero-knowledge Proofs play a vital role in enhancing privacy and security within digital identity systems. Traditionally, identity verification procedures often entail disclosing sensitive personal information, introducing risks and inefficiencies. However, Zk-Proofs provide secure and efficient alternatives that avoid revealing specific details, thereby safeguarding privacy while upholding essential assurance levels.

- **Proof of Humanity** As AI-generated content increasingly dominates online platforms, Zk-Proofs will become indispensable for verifying the authenticity of human-generated content. “Proof of humanity” systems already utilize Zk-Proofs to accurately confirm human access to specific resources. Moreover, as AI threatens to replace more human jobs, there is heightened interest in Universal Basic Income (UBI), an age-old concept first proposed by English statesman and philosopher Thomas More. UBI entails providing a fixed subsidy to support displaced workers who lack alternative means of sustenance. In such a scenario, Zk-Proofs can be used to verify human identity without revealing additional information.

  In Figure 3, different entities can attempt to prove they are human, including AI, chip-implanted and AI-trained monkeys, or hackers. Zk-Proof can authenticate real humans without releasing additional information. If authenticating directly with the secure humanity database, it is very likely that humans will disclose more information than they are indeed human.

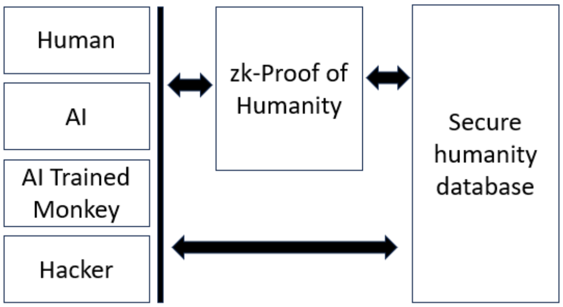


Fig. 3: Zk-Proof of Humanity has the advantage of not disclosing more info except that one is a real human.

- **Password Authentication Systems**: Traditional password-based authentication requires users to disclose their passwords during the authentication process, which poses security risks if passwords are intercepted or stored improperly. However, Zk-Proof systems provide a secure alternative by eliminating the need for password disclosure. These systems ensure that users can authenticate themselves without compromising their passwords, creating a cycle of secure authentication. This is achieved through the use of zk protocols, which allow for efficient and confidential authentication. However, by using Zero-knowledge Proof systems such as Secure Remote Password (SRP), users can engage in interactive proof protocols to prove their knowledge of a password without actually revealing it, thus mitigating the risk of password theft.
- **Blockchain Technology and Cryptocurrency**: The advent of blockchain technology reignited interest in Zero-knowledge Proofs. Zcash, a cryptocurrency employing zk-SNARKs, prioritizes transaction privacy. By incorporating Zero-knowledge Proofs, blockchain networks can validate transactions while preserving user anonymity. This allows verifiers to confirm transaction validity without accessing specific details or identities. In 2021, the Ethereum Foundation unveiled the zkEVM project, aimed at integrating zk proofs into the Ethereum blockchain. ZK-EVM facilitates private transactions and computations, empowering users to prove transaction or smart contract validity without revealing underlying data. This enhances privacy for users and applications by concealing sensitive information while ensuring transaction integrity and correctness. Zero-knowledge Proofs hold promises for enhancing the scalability of the Blockchain networks by reducing computational and storage overhead associated with verifying transactions and smart contracts.

  Secure Voting Systems: Another area of Zk-Proof application is secure voting systems, where Zero-knowledge Proofs can ensure the integrity and privacy of votes.

  Applications in AI and Machine Learning Zk-Proofs application in machine learning hold very big promise. As AI-generated content becomes increasingly indistinguishable from human-created content, zero-knowledge cryptography could be used to verify that a specific piece of content was produced by a human. This could be done by creating a zero-knowledge circuit representation of the model, which would allow the content to be verified without revealing the model itself or any of the input data. By leveraging zk-SNARKs, machine learning models can be trained on private data without exposing it to creators or users, enabling development in sensitive sectors like healthcare or finance while safeguarding individual privacy.

4. CHALLENGES AND POTENTIAL DISRUPTIONS

Despite the rapid progress and promising applications, ZK proof has some ongoing challenges that hindered its wider applications:

4.1. Challenges

- **Computational Overhead**: Zero-knowledge Proofs, especially earlier protocols, frequently involve substantial computational overhead. Although advancements like zk-SNARKs aim to alleviate this burden, the generation and verification of proofs remain computationally intensive. This complexity presents challenges for developers seeking to integrate zk proofs into other applications. However, hardware acceleration techniques, such as trusted execution environments (TEEs) or specialized hardware like ASICs, have facilitated quicker and more efficient computation of Zk-Proofs. This advancement holds the promise of rendering Zk-Proofs viable for real-time applications.
- **Scalability** Handling high transaction volumes with Zk-Proofs can pose challenges. Generating and verifying numerous proofs may create bottlenecks, affecting transaction speed, network overhead, or computational load. It's vital to strike a balance between privacy and performance, as robust privacy assurances may require navigating these trade-offs.
- **Data verification limitations** The constraints of data verification in Zk-Proofs mainly arise from their dependence on mathematical algorithms, making them particularly suitable for verifying numerical data. However, when handling categorical data, the conversion into numerical formats becomes imperative, potentially adding further complexities.
- **Interoperability** Interoperability between Zk-Proof systems is handicapped by multitudes of standards and

protocols. Zk Proof Community aimed to address the issue by instituting common standards.

- **Trusted setup** Trusted setup is essential for some ZK-Proof systems to establish initial parameters, relying on the honest behavior of the early users to ensure security. Any compromise or error in this setup can undermine the system's security and privacy guarantees. As the field evolves, ongoing research is crucial to address emerging vulnerabilities and enhance zk-proof system efficiency.

4.2. Potential Disruptions

Zk-Proof, while promising, may still be impacted by rapid advancement in other high-tech fields. Quantum computing and homomorphic encryption are two of them.

- **Quantum computing** Quantum computing presents a potential threat to the cryptographic primitives underpinning Zk-Proofs, as large-scale, fault-tolerant quantum computers could compromise them. In such a scenario, all cryptographic schemes in use would require updating. Nevertheless, considerable strides have been taken in the development of quantum-resistant encryption, with industry leaders also making headway in quantum-resistant SNARKs.
- **Homomorphic encryption:** Homomorphic encryption, especially fully homomorphic encryption (FHE), allows computations on encrypted data without decryption. While FHE and Zk-Proofs serve different purposes in cryptography, they frequently complement each other. FHE enables secure computation and privacy-preserving data analysis, while Zk-Proofs verify statements about data or knowledge without disclosing them. However, combining Zk-Proofs with FHE computations could achieve unparalleled privacy in computation, concealing both data and operations. This advancement has the potential to revolutionize secure cloud computing, private data analysis, and beyond.

5. CONCLUSIONS

Zero-knowledge Proofs can provide a new level of security and privacy by proving the validity of information or decisions without revealing the underlying data or algorithms. It also has great potential in areas such as proving of humanity, password authentication, Blockchain, and cryptocurrencies, voting systems integrity, and secure AI and machine leanings. That said, to have wider applications, we need to examine the challenges more closely, including computational overhead, scalability, data verification limitations, interoperability, and issues that can happen during the trusted setup. Quantum computing and Homomorphic encryption also pose potential disruptions to the Zk-Proof technologies. That said, Zk-Proof is a very valuable technology to explore and expand.

6. REFERENCES

[1] Shafi Goldwasser, Silvio Micali, and Charles Rackoff, "Knowledge complexity of interactive proof-systems.," 1985.

[2] Gregory Neven, Nigel P. Smart, and Bogdan Warinschi, "Hash function requirements for schnorr signatures," *Journal of Mathematical Cryptology*, vol. 3, 2009.

[3] Giacomo Fenzi, "Zero knowledge proofs theory and applications," *University of St. Andrewsl Cryptology*, vol. 3, 2019.

[4] Manuel Blum, "How to prove a theorem so no one else can claim it," in *Proceedings of the International Congress of Mathematicians*. Citeseer, 1986, vol. 1, p. 2.

[5] Shafi Goldwasser, Silvio Micali, and Charles Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, 1989.

[6] Amos Fiat and Adi Shamir, "How to prove yourself: Practical solutions to identification and signature problems," 1987, vol. 263 LNCS.

[7] Russell Impagliazzo and Moti Yung, "Direct minimum-knowledge computations," 1988, vol. 293 LNCS.

[8] Oded Goldreich, Silvio Micali, and Avi Wigderson, "Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems," *Journal of the ACM (JACM)*, vol. 38, 1991.

[9] Jan Camenisch and Markus Stadler Ubilab, "Efficient group signature schemes for large groups," 1997, vol. 1294.

[10] Manuel Blum, Paul Feldman, and Silvio Micali, "Non-interactive zero-knowledge and its applications," 1988.

[11] Uriel Feige, Dror Lapidot, and Adi Shamir, "Multiple non-interactive zero knowledge proofs based on a single random string," *IEEE Transactions on Industry Applications*, vol. 27, 1991.

[12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer, "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again," 2012.

[13] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," 2013.

[14] Benedikt Bunz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," 2018, vol. 2018-May.

[15] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," *Eprint.Iacr.Org*, 2018.