International Journal of Knowledge Based Computer Systems
12 (1) 2024, 34-45
http://www.publishingindia.com//ijkbcs/

Zero Knowledge Proof: An Overview

Tamanna Yesmin Rashme
Computer Science & Engineering, Jagannath University, Dhaka, Bangladesh.
Email: tamanna1991.cse@gmail.com
*Corresponding Author

Abstract: Zero-knowledge proofs (ZKPs) enable the validation of knowledge without revealing the underlying information, revolutionizing the way data is collected, utilized, and shared. In a ZKP transaction, a "prover" attempts to convince a "verifier" of the validity of a statement without disclosing the actual data or the method used. The prover proves their capability to solve a problem by presenting the result without exposing the input or process. This paper explores zero-knowledge proofs (ZKPs), their various types and classifications, and their research areas. It also delves into the challenges and limitations faced when applying zero-knowledge techniques for privacy protection and authentication across different scenarios. Additionally, this paper also discussed the differences between interactive zero-knowledge (IZK) and non-interactive zero-knowledge (NIZK) proofs, highlighting their respective advantages and applications as well as the challenges for ZKP, such as computational complexity and hardware requirements, are also examined, along with their integration in secure digital identity management and verifiable anonymous voting systems.

Keywords: Application, Challenges, IZK, NIZK, Prover, Verifier, Zero knowledge proof, ZKP.

I. Introduction

Zero-knowledge proofs (ZKPs) are a revolutionary concept in the field of cryptography, first introduced by researchers Shafi Goldwasser, Silvio Micali, and Charles Rackoff in the 1980s. It is feasible to demonstrate a statement's validity without disclosing any knowledge about the statement itself, which is the foundation of this novel concept. ZKPs have transformed the landscape of complexity theory and cryptography by allowing for the secure verification of information without exposing the actual data [1].

The essence of a ZKP lies in its ability to enable a prover to convincingly assure a verifier of the truth of a statement, while keeping the underlying information secret. This cryptographic method uses an interactive protocol where the prover undertakes a series of challenges to demonstrate their knowledge of a secret to the verifier [2]. The ingenuity of ZKPs ensures that at no point is sensitive information compromised, making it a powerful tool for preserving privacy.

This technique has significant applications, especially in the realm of authentication processes on public and private networks, including the internet. Traditional methods of authentication often involve sharing passwords or other sensitive details that are vulnerable to security breaches. ZKPs, on the other hand, allow for robust authentication without transmitting any such information, closing the gap that attackers exploit.

In general, zero-knowledge proof protocols have been incorporated into various authentication algorithms, providing a secure and efficient alternative to conventional systems. Using ZKPs for password authentication and identity verification is especially significant since it increases electronic communication safeguarding by one level. The development of ZKP-based systems represents a leap forward in the ongoing effort to safeguard digital identities and personal information in an increasingly interconnected world.

The article is organized as follows. The Overview of ZKP, types and classification Zero Knowledge

Proof are presented in the following section. Also, present application and challenges for ZKP in the field of technology. Open research problems are then analyzed. This article is concluded in the final section.

II. Overview of ZKP

Cryptographic procedures known as zero-knowledge proofs (ZKPs) enable one party, called the prover, to demonstrate to another party, called the verifier, that an accusation is legitimate without withholding any more details regarding the claim. This segment will offer a succinct overview of ZKPs, employing a specific example to illustrate their fundamental operational concepts.

A. Definition of ZKP

Zero-knowledge proofs (ZKPs) provide a sophisticated method for a prover (P) to confirm the accuracy of a statement to a verifier (V), all while withholding any additional information except for the correctness of the statement. What distinguishes ZKPs is their capacity to authenticate possession of certain information without actually disclosing the information itself or any related details [3].

A zero-knowledge proof system is a cryptographic protocol involving two parties: a prover (P) and a verifier (V), used for asserting the truth of a statement without revealing any underlying information. For any statement within a language L, and an associated auxiliary input $z \in \{0,1\}^*$ a zero-knowledge proof system is defined such that there exists a simulator S, which can replicate the interaction between P and any probabilistic polynomial-time verifier V^* in a way that the generated conversation is indistinguishable from the real interaction to any third party [4].

In mathematical terms, for every element $a \in L$ and string s, the relationship can be represented as:

$$\text{SimulatedView}[P(a) \leftrightarrow V^*(a,s)] \sim^c S(a,s)$$

Here, $\sim^c$ symbolizes computational indistinguishability, indicating that no probabilistic polynomial-time algorithm can distinguish between the simulated view and the real interaction with non-negligible probability. The utility of zero-knowledge proof as a method for authentication stems from its distinctive attributes.

(1) *Completeness:* for every mutual input a that belongs to the language L and a given polynomial $p(\cdot)$.

$$\Pr[(P,V)(a) = 1] \geq 1 - \frac{1}{p(|a|)}$$

(2) *Soundness:* For any mutual input a $\notin L$ and any interactive TM P' with polynomial p(.).

$$\Pr[(P',V)(a) = 1] < 1 - \frac{1}{p(|a|)}$$

(3) *Zero Knowledge:* A corresponding probabilistic polynomial-time algorithm PM^* exists for every element a in the set L, for each verifier modeled as a probabilistic polynomial-time Turing machine TM^*.

$$(P,V^*)(a) \approx cPM^*(a)$$

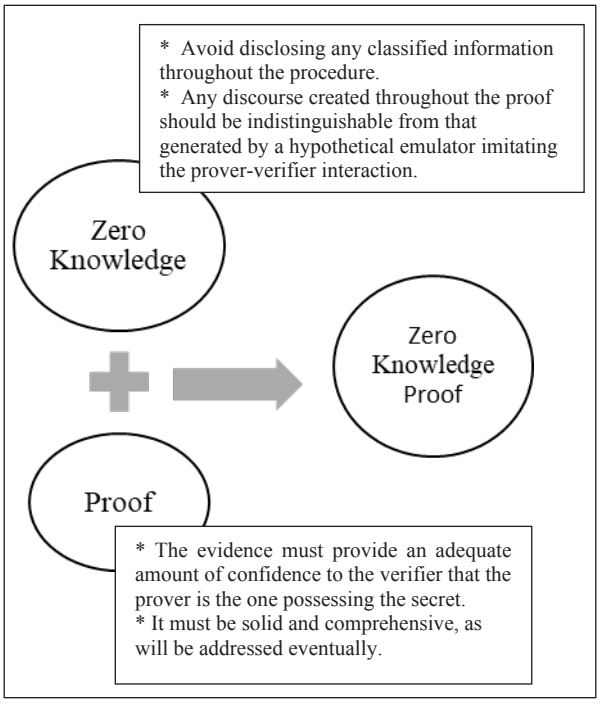

Fig. 1: Definition of Zero Knowledge Proof

At the heart of a zero-knowledge proof system are its essential properties: completeness, soundness, and the zero-knowledge aspect. Completeness ensures that for all legitimate inputs within the set L, an

honest prover can invariably convince the verifier of the proof's validity. Soundness protects against deceitful provers by ensuring that no false claims about inputs not in L can be erroneously verified. Zero-knowledge guarantees that a verifier learns nothing beyond the validity of the statement during the interaction. Moreover, these properties can adapt to the computational strength of the parties involved, distinguishing between statistical or perfect zero-knowledge and computational soundness, leading to what is known as a zero-knowledge argument system.

B. Example of ZKP

A particular method allows for the verification of graph isomorphism in a zero-knowledge proof manner. In this method, the prover aims to convince the verifier that they know a permutation p that maps one graph G_0 to another G_1 without revealing any information about π itself. The process unfolds in a series of steps where the prover first selects a permutation σ and a bit b and sends $S = \sigma(G_b)$ to the verifier. The verifier then chooses a random bit b' and sends it back to the prover. Based on b and b', the prover sends a permutation τ to the verifier, who then checks if S equals $\tau\,(G_{b'})$ to determine the validity of the claim.

The procedure of the protocol is detailed below.

$$\begin{cases} \sigma & if\ b = b' \\ \sigma\pi^{-1} & if\ b = 0, b' = 1 \\ \pi\sigma & if\ b = 1, b = 0 \end{cases}$$

Acceptance by the verifier hinges on the condition $S=\tau(Gb')$. The protocol ensures perfect completeness if π is truly an isomorphism and maintains soundness with the assumption that the verifier chooses b' randomly, leading to a half chance for any dishonest prover to succeed. The zero-knowledge property is maintained by the ability of a simulator to generate an indistinguishable view from the verifier's perspective by randomly choosing b' and σ, thus matching the verifier's view distribution when G_0 is isomorphic to G_1.

Another example is Alibaba Cave consider as explanation for child teaching [4]. In this cave, there are two characters: Peggy (the prover), who claims to know a secret passcode, and Victor (the verifier), who seeks proof of Peggy's claim without learning the secret itself. The cave is structured with a single entrance that leads to a fork with two indistinguishable paths, labeled C and D. These paths loop back around to the entrance. The correct path can only be accessed by using the secret passcode.

To prove she knows the secret despite demonstrating it, Peggy undergoes a series of tests:

- Peggy begins at the entrance (point A) and proceeds to either point C or D, out of Victor's view.
- Victor then positions himself at the fork (point B) and calls out to Peggy to return through either path C or D.
- If Peggy genuinely knows the secret, she can use it to return through the specified path, regardless of her initial choice.

This process is repeated multiple times. If Peggy emerges from the correct path each time, Victor becomes increasingly, convinced that Peggy knows the secret. However, if Peggy is bluffing, her odds of consistently guessing the correct path diminish exponentially with each round.

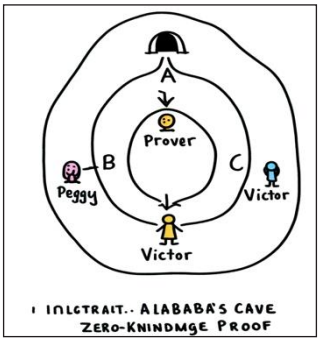

Fig. 2: Zero Knowledge Proof Example: Alibaba Cave

The probability that Peggy can fool Victor without actually knowing the secret decreases as $\frac{1}{2^n}$ where n is the number of times the test is performed. Mathematically, this is represented by the equation:

$$P(Peggy\ Fools\ Victor) = {}^{1}/_{2^n}$$

The value n is the optimal number of rounds needed for Victor to trust Peggy's proof without any doubt. If Peggy knows the secret, the proof is considered complete, as she will pass every test. This demonstrates both the completeness and soundness of the zero-knowledge proof: completeness through Peggy's ability to always pass the test if she knows the secret, and soundness through the improbability of her deceiving Victor without this knowledge.

III. Classification of ZKP

A. Interactive Zero Knowledge Proof

A sequence of probabilistic steps intended to persuade the verifier of the veracity of a given assertion constitute an interactive zero-knowledge proof. During this procedure, the prover may show the verifier the validity of the claim without giving any additional details about the underlying proofs or secret inputs. Interactive Zero Knowledge is a proof system in which a problem P is shown via an interactive communication between a prover p with unlimited computing resources and a verifier v working under polynomial time limitations. This system adheres to two critical principles:

- *Completeness:* For any instance *x* that is a valid example of P, the verifier *v* will likely accept the proof provided by *p* after their interaction based on the common input *x*.
- *Soundness:* If *x* is not a valid instance of ΠΠ, then regardless of the prover strategy p^* employed, even if it is computationally unbounded, *v* is highly likely to reject the proof following their interaction on the common input *x*.

The Fiat-Shamir heuristic takes an interactive zero-knowledge proof protocol, where the prover and verifier engage in a back-and-forth exchange, and converts it into a non-interactive zero-knowledge proof that can be publicly verified [6]. This method transforms an interactive proof of knowledge into a non-interactive digital signature. Developed by Amos Fiat and Adi Shamir in 1986 [7], the heuristic requires the original interactive proof to have the property of being a public coin, meaning the verifier's random coins are made public throughout the proof protocol. Initially, the security of the Fiat-Shamir heuristic lacked a formal proof, but If random oracles exist, as subsequent research by Point cheval and Stern showed, it is secure against selected message attacks in the random oracle paradigm. The significance of random oracles in cryptography is highlighted by the Fiat-Shamir heuristic. It can be seen as a way to convert a public-coin interactive proof of knowledge into a non-interactive proof of knowledge. If the original interactive proof is used for identification purposes, the resulting non-interactive version can be directly employed as a digital signature by incorporating the message as part of the input to the random oracle.

B. Non-Interactive Zero Knowledge Proof

A Non-Interactive Zero Knowledge (NIZK) proof system is characterized by three core components: the Setup, Prove, and Verify algorithms:

- *Setup Algorithm:* This component is tasked with generating parameters essential for the operation of the proof system. It takes a security parameter λ as input and produces parameters Parampara's, which are used throughout the NIZK system.
- *Prove Function:* The proof is built using this function. This method produces a zero-knowledge proof, which does not involve communication between the prover and the verifier, given an instance x from an NP-language L and a matching witness w.
- *Verify Algorithm:* This algorithm evaluates the proof. It takes the proof as input and outputs a Boolean value, typically denoted as *b*. If *b* is 1 (or true), it indicates that the proof has been accepted by the verifier, confirming the validity of the assertion made by the prover without revealing any additional information about the witness *w*.

NIZK proofs are particularly valuable in scenarios where communication must be minimized, or where repeated verification of the proof is expected without additional interaction. They offer robust security

features and are widely used in applications ranging from secure voting systems to confidential financial transactions, where it's crucial to verify a statement's accuracy without disclosing underlying data.

Non-Interactive Zero-Knowledge Proofs (NIZK) utilize a three-component framework consisting of the algorithms M, P, and V for setup, proving, and verification, respectively. In NIZK, M generates a common reference string σ using a security parameter λ, and P constructs a proof π asserting that a given statement C with witness w is valid within a polynomial-time computable binary relation B. Verification via V determines if C and π conform to σ, outputting 1 for acceptance. Crucial attributes of NIZK include perfect completeness, where valid proofs are always accepted, statistical soundness that prevents false proofs from being accepted by non-uniform polynomial-time adversaries, and computational zero-knowledge, which ensures proof simulation without the witness, maintaining the indistinguishability between real and simulated proofs under polynomial-time constraints.

The Fiat-Shamir heuristic is a process used to convert an interactive proof into a non-interactive one within the random oracle model [8]. The core concept involves substituting the verifier's random challenge with a hash function's output, which acts as a random oracle. This hash function processes the prover's initial message and the input to generate the challenge. For instance, in the Schnorr identification protocol adaptation, the prover generates a commitment $u= b^r$ and then uses the hash of the base b, commitment u, and public key k to create a challenge c. The response z is calculated as $z = r + cx$ and the proof π consists of (u, c, z). Verification checks if b^z equals uh^c.

Security Properties:

- *Completeness:* Maintains the interactive Schnorr protocol's property where valid proofs are always accepted.
- *Zero-Knowledge:* The simulation mimics the interactive setting where the challenge isn't issued by a verifier but is derived from the hash function, allowing the simulator to control the challenge by setting the hash function output.
- *Knowledge:* This adjusted method ensures that the prover's ability to consistently produce valid proofs implies possession of the secret (discrete log of h), akin to the original interactive protocol's security proof. If the prover can generate two valid proofs with different challenges for the same commitment, it reveals knowledge of the secret.

a) The Applied Researches of NIZK

Madhav Agal *et al.* [9], delve into the limitations of traditional password-based authentication systems, highlighting their susceptibility to cyber-attacks and security lapses. The paper explores the application of Non-Interactive Zero-Knowledge Proofs (NIZK) as a more secure alternative that eliminates the need to transmit passwords, thereby enhancing security. It also addresses vulnerabilities to Replay attacks within NIZK implementations and proposes a comprehensive two-pronged strategy to mitigate these risks. Group signatures, ring signatures, and electronic voting are areas where non-interactive zero-knowledge (NIZK) proofs are widely used. Blum *et al.* [10] groundbreaking work explores the possibility of non-interactive zero-knowledge (NIZK) proofs using a shared random string between prover and verifier. It proves the existence of NIZK proofs for number-theoretic languages and shows that NIZK is possible for NP-complete satisfiability if quadratic residuosity is hard. This foundational research established the feasibility of NIZK proofs without additional assumptions.

K. Yang and X. Wang [11] investigates multi-verifier zero-knowledge (MVZK) proofs, presenting a protocol for efficient and scalable MVZK proofs of circuit satisfiability, where security is maintained even when a minority of verifiers collude with the prover. The protocol requires minimal communication per verifier and memory proportional to the circuit size, making it practical for real-world applications requiring verifiable computation.

IV. Types of ZKP

In a zero-knowledge proof system, the prover is tasked with demonstrating the validity of a claim,

while the verifier's role is to assess and confirm the correctness of the proof. These protocols have the capability to publicly verify that the hidden information is valid, without the prover needing to disclose the underlying details. The prover can be highly confident that the verifier will be convinced of the claim's truthfulness through the execution of the zero-knowledge proof protocol [12]. The types of Zero Knowledge Proof's are following:

A. zk-SNARKs

Zero-knowledge, or zk-SNARKs, Simplified Non-interactive Arguments of Knowledge (also known as cryptographic proofs) enable one person (the prover) to demonstrate another individual (the verifier) that they are aware of a specific piece of knowledge without actually disclosing what information it is. The key properties of zk-SNARKs are:

- *Zero-Knowledge:* The prover does not reveal any information about the secret input beyond the fact that they know it.
- *Succinctness:* The proof is very compact, typically a few hundred bytes in size, regardless of the size of the computation.
- *Non-Interactivity:* The proof can be verified without any further interaction between the prover and verifier.

The core of a zk-SNARK is a mathematical problem called the "R1CS" (Rank-1 Constraint System) problem. In this problem, the prover must convince the verifier that they know a secret witness 'w' that satisfies a set of quadratic constraints represented by a matrix 'A' and a vector 'b'. Formally, the R1CS problem is defined as:

$$\exists\ w\ such\ that\ A * w = b$$

A is a matrix of dimensions m x n, b is a vector with size m, and w is a vector with size n.

a) Example: zk-SNARKs can be used for basic math to prove the square root of a number x without revelaing the actual value of x, can be used.

- The prover generates a zk-SNARK proof that they know a value w such that w^2 = x.
- The verifier can then quickly verify the proof without learning the value of w (the square root of x).

b) Application: zk-SNARKs have a wide range of applications in the field of cryptography and blockchain technology, including:

- *Privacy-Preserving Cryptocurrency Transactions*: zk-SNARKs can be used to build cryptocurrencies like Zcash, where transactions can be kept private.
- *Decentralized Finance (DeFi) Protocols*: zk-SNARKs can be used to build DeFi applications that preserve user privacy.
- *Scalable Blockchains:* zk-SNARKs can be used to build scalable blockchains by allowing for efficient verification of off-chain computations.
- *Secure Multi-Party Computation*: zk-SNARKs can be used to build secure protocols for multi-party computation, where parties can collaborate without revealing their private inputs.

B. zk-STARKs

zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge) are a type of cryptographic proof system that, like zk-SNARKs, allow a prover to convince a verifier that they possess knowledge of some information, without revealing that information [13]. The key properties of zk-STARKs are:

- *Zero-Knowledge*: The prover does not reveal any details about the secret input.
- *Scalability*: zk-STARKs are highly scalable, able to handle large computations.
- *Transparency*: The setup process for zk-STARKs does not require a trusted setup, making it more transparent than zk-SNARKs.

The core of a zk-STARK is based on the notion of a Algebraic Intermediate Representation (AIR). In an AIR, the prover must convince the verifier that they know a witness w that satisfies a set of linear and quadratic constraints represented by a matrix A and a vector b. Formally:

$$\exists\ w\ such\ that\ A * w = b$$

Where A is an m x n matrix, b is an m-dimensional vector, and w is an n-dimensional vector.

a) Example: Suppose A wants to prove to B that it has computed the 1000th Fibonacci number, without revealing the actual number. So, it can do this using zk-STARKs:

- A generates a zk-STARK proof that it knows a sequence of numbers that satisfies the Fibonacci recurrence relation, and the 1000th number in that sequence.
- B can quickly verify the proof without learning the Fibonacci number itself.

b) Application: zk-STARKs have a wide range of applications, including:

- *Privacy-Preserving Computations*: zk-STARKs can be used to perform computations on encrypted data without revealing the inputs.
- *Blockchain Scalability*: zk-STARKs can be used to scale blockchain systems by allowing efficient verification of off-chain computations.
- *Secure Multi-Party Computation*: zk-STARKs can be used to build protocols where multiple parties can collaborate without revealing their private inputs.
- *Decentralized Applications*: zk-STARKs can be used to build privacy-preserving decentralized applications on top of blockchain networks.

C. Bulletproof

Bulletproofs is a groundbreaking zero-knowledge proof protocol that provides concise, logarithmic-sized proofs without requiring a trusted setup [14]. It excels in range proofs for committed values, confirming n-bit ranges with just $2\log_2(n) + 9$ elements. Generation and verification are linear (in n). Bulletproofs enable the aggregation of multiple range proofs, allowing a party to verify multiple commitments within a range by adding only $O(\log(m))$ group elements to a single proof's length. Bulletproofs surpass previous methods used in confidential transactions within Bitcoin and other cryptocurrencies, which featured range proofs of linear size. Additionally, Bulletproofs can be created through multi-party computation (MPC) protocols without revealing inputs, optimizable for communication and computational efficiency. Verification times grow linearly but can be accelerated through batching, similar to ECDSA signatures. Bulletproofs can be applied to general arithmetic circuits under the discrete logarithm assumption, making them ideal for enhancing security and efficiency in various blockchain applications.

D. PLONK

PLONK (Permutations over Lagrange-bases for Oecumenical Noninteractive Arguments of Knowledge) is a zk-SNARK proof system designed to address the high computational overhead associated with the proof construction process in earlier systems like Sonic. Unlike previous zk-SNARK constructions, which required a circuit-specific trusted setup, PLONK introduced a universal and continuously updatable trusted setup. This innovative approach allows PLONK to be used across a wide variety of circuits, providing both versatility and enhanced security guarantees [15]. Additionally, PLONK boasts faster proving times and more succinct verification compared to prior zk-SNARK protocols, making it a significant advancement in the field of cryptographic proof systems. These improvements enable PLONK to be more efficiently deployed in real-world applications that require secure and scalable zero-knowledge proofs.

V. Difefrence Between IZK and NIZK

The two main types of Zero-Knowledge Proofs (ZKPs) are Interactive Zero-Knowledge Proofs (IZKPs) and Non-Interactive Zero-Knowledge Proofs (NIZKPs). Interactive Zero-Knowledge Proofs require multiple rounds of interaction between the prover and the verifier. In this setting, the prover sends proofs to the verifier, who in turn sends back challenges, thus establishing the validity of the statement through continuous communication. This interaction is crucial in ensuring that the verifier is convinced without learning any additional information about the statement. IZKPs are typically used in real-time communication environments

where the security of the protocol can be maintained through active participation by both parties [16, 17]. In contrast, Non-Interactive Zero-Knowledge Proofs eliminate the need for back-and-forth communication. NIZKPs rely on a shared random string or a pre-established common reference string that allows the prover to generate a single proof that can be verified independently by the verifier without any interaction. This makes NIZKPs more practical for scenarios where interaction is costly or impossible, such as in blockchain protocols and other distributed systems [18, 19]. The non-interactive nature of NIZKPs provides efficiency and scalability, as a single proof can be verified by multiple parties without the need for repeated communication.

VI. Security Implementation using NIZK vs IZKP

The security implications of using Interactive Zero-Knowledge Proofs (IZKPs) versus Non-Interactive Zero-Knowledge Proofs (NIZKPs) revolve around the nature of communication and the associated vulnerabilities. Interactive Zero-Knowledge Proofs require multiple rounds of interaction between the prover and verifier, ensuring that the verifier is actively involved in the verification process without learning anything beyond the validity of the statement. This continuous interaction can enhance security by allowing dynamic adjustments to potential threats during the verification process. However, it also introduces risks related to the integrity and security of the communication channel itself, as each interaction could be a point of potential attack [16].

In contrast, Non-Interactive Zero-Knowledge Proofs eliminate the need for such back-and-forth communication, relying instead on a shared random string or a common reference string. This makes NIZKPs more suitable for environments where interaction is costly or impractical, such as in blockchain protocols and distributed systems. The key security advantage of NIZKPs is that they reduce the attack surface by limiting the number of interactions required, thus minimizing the potential for communication-based vulnerabilities [19]. However, this also means that any compromise of the common reference string can lead to security breaches, as the same proof can be used multiple times without re-validation [20].

Additionally, the use of non-interactive proofs can be advantageous in scenarios requiring high scalability and efficiency, as a single proof can be verified by multiple parties independently. This scalability comes with the caveat that the security assumptions need to be robust against potential collusion among verifiers, especially in multi-verifier settings [21]. Overall, the choice between IZKPs and NIZKPs depends on the specific application requirements, including the desired balance between security, efficiency, and the feasibility of interaction.

VII. The Applied Researches of ZKP

Zero-knowledge proofs have been extensively studied and have generated significant academic interest. A prominent area within cryptography and computational complexity theory that has received widespread attention is the field of zero-knowledge proof systems, which have been the subject of numerous hypotheses and research.

X. Yang *et al.* developed a prototype system called BZDIMS featuring a challenge-response protocol that enables users to selectively reveal their attribute ownership to service providers, thereby safeguarding user behavior privacy. Our performance and security assessments indicate that this system provides enhanced attribute privacy protection and broader applicability than previous models. Muhammed F. Esgin *et al.* [22] introduces innovative techniques for efficient lattice-based zero-knowledge proofs (ZKPs), enhancing both computational and communication efficiencies with one-shot proof techniques for non-linear polynomial relations. It presents two new speed-enhancing techniques—CRT-packing and NTT-friendly tools—and demonstrates their application in creating effective proof systems for cryptographic constructs like ring signatures. The proposed methods not only minimize proof length and computational time but also eliminate the need for a trusted setup, making them ideal for applications in cryptocurrencies and e-voting systems. Manish S. and Yichen H. proposed a model that aims to

eliminate the involvement of third-party entities and the transmission of user passwords over the network, providing protection against vulnerabilities like key logging, shoulder surfing, and eavesdropping inherent in single-factor authentication schemes. This work explores the application of zero-knowledge proof protocols as a second-factor authentication mechanism for online banking systems, addressing the security limitations of traditional password-based authentication.

A novel decentralized identity authentication system proposed by Tianyu B. *et al.* [23] called Health-zkIDM leveraging zero-knowledge proof and blockchain technology, designed to address privacy concerns and interoperability limitations of centralized IDMs in healthcare. By integrating zero-knowledge proofs with blockchain, specifically on the Hyperledger Fabric platform, the system enhances privacy and security, allowing patients to securely verify their identities across different healthcare providers. Performance tests indicate that Health-zkIDM can handle over 400 transactions per second, demonstrating its efficiency and scalability. Houyu Zheng *et al.* [24] proposed a novel medical insurance claim scheme that leverages smart contracts, blockchain, and zero-knowledge proof techniques. The scheme ensures the legitimacy and privacy of transactions between patients and insurance companies through the use of non-interactive zero-knowledge proofs and homomorphic encryption, while also preserving the privacy of patient identities by integrating the Schnorr protocol and Fiat-Shamir heuristic. The security analysis and performance evaluation demonstrate the feasibility and efficiency of the proposed scheme compared to prior approaches. Oleksandr Kuznetsov *et al.* [25] proposed a system designed to reduce the proof size and computational demands of blockchain data verification, providing a more efficient and secure framework. Through comprehensive testing with real Ethereum data, the proposed solution demonstrates significant improvements in proof compactness and processing efficiency, surpassing traditional methods. The research contributes a scalable and robust verification mechanism that enhances blockchain applications across multiple sectors, including finance and supply chain management.

VIII. Challenges of ZKP

Zero-knowledge proofs (ZKPs) hold significant promise for enhancing privacy and security in digital applications, yet they face several challenges that affect their practical deployment:

- *Computational Complexity and Hardware Requirements:* ZKPs require complex calculations that necessitate the use of advanced, specialized hardware, making them expensive and less accessible for widespread use. These high costs often get passed down to consumers, reducing affordability and limiting adoption. Furthermore, the demanding hardware requirements make it difficult to implement ZKPs on mobile devices. Companies like Ingonyama are working on specialized hardware solutions to mitigate these issues, but the success of such initiatives is yet to be fully determined.
- *Verification Expenses:* Beyond the costs of generating proofs, verifying them also involves intensive computations, adding to the overall expense of ZKP applications. Even with the most efficient zk-SNARKs, the verification process is resource-heavy. For example, verifying a single zk-SNARK proof on Ethereum can cost about 500,000 gas, impacting the cost-effectiveness of solutions like zk-rollups.
- *Limited Consumer Applications:* Despite their potential, ZKPs have yet to find widespread application in consumer-facing products. While they are used in Ethereum scalability solutions such as ZK-rollups, popular applications like Tornado Cash have faced legal challenges due to their association with illicit activities, which stymies broader consumer adoption.
- *Trust and Setup Concerns:* zk-SNARKs require a "trusted setup," where participants create public parameters using secret inputs that must then be discarded to prevent misuse. This setup makes it necessary for users to trust these participants, as there is no way to independently verify the disposal of the inputs. Efforts are ongoing to develop zkSNARKs that do not require a trusted setup, aiming to enhance trustlessness.

- *Quantum Vulnerability:* zk-SNARKs rely on elliptic curve cryptography, vulnerable to potential quantum computing advances. This presents a security risk, although zk-STARKs, which use collision-resistant hashing, offer a quantum-resistant alternative, ensuring more robust long-term security.
- *Development Barriers:* The ecosystem for ZKPs lacks comprehensive developer tools and educational resources, making it challenging for developers, especially those without a strong background in mathematics or cryptography, to create ZKP-based applications. This shortage of tools and training materials hinders wider adoption and slows the advancement of ZKP technologies.

IX. Application of ZKP

- *Blockchain Technology:* Zero-Knowledge Proofs (ZKPs) are a powerful cryptographic tool with significant potential in enhancing blockchain technology. Blockchain systems can leverage ZKPs to meet specific needs for protecting sensitive information or ensuring data privacy. ZKPs can validate transactions while keeping details like the sender, recipient, and other transaction-related data confidential. Conceptually, the blockchain is seen as a decentralized network of miners who maintain a secure consensus protocol known as the Global State. While the blockchain is trusted for its accuracy and availability, it is not inherently private [29]. A notable application of ZKP in blockchain is ZCash, which was the first cryptocurrency to implement zk-SNARKs, demonstrating the technology's capability to enhance transaction privacy.
- *Verifiable Anonymous Voting:* A country or corporate organization must allow voting to maintain democracy. Voters' identities may, however, be compromised throughout the voting process. Implementing verifiable, anonymous voting systems is made possible by Zero-Knowledge Proofs (ZKPs) [16]. By employing ZKPs, voters can prove their eligibility and cast votes without revealing their identities. Additionally, ZKPs enable voters to obtain proof that their votes have been counted in the results, ensuring both privacy and verifiability of the election process.
- *Encrypted Remote Biometric Verification*: Utilizing distinct biological characteristics such as fingerprints, face recognition, or iris patterns, remote biometric authentication may expose sensitive information to unapproved parties. ZKPs may protect this procedure by making sure that no private biometric information is revealed during the authentication process. Additionally, ZKPs give a proof of authentication, proving the authenticity of the access without compromising security.
- *Secure Exchange of Digital Assets:* Digital assets, defined by unique, valuable binary data, often require privacy-sensitive exchanges. Utilizing ZKPs, two parties can swap digital assets without exposing their identities or the details of the assets being exchanged. ZKPs also provide a verifiable record of the transaction process, enhancing the security of digital asset exchanges.
- *Safe-Trading:* The bidding process integrity is critical in government auctions, as many vendors secretly submit offers in a competitive setting. Bids are kept private for the first part of the normal two-phase procedure and then made public for assessment. ZKPs may stop the disclosure of supplier names and unsuccessful bids, offering verifiable evidence that protects participant privacy and validates the legitimacy of the auction results.

X. Conclusion

Zero-knowledge proofs (ZKPs) are a groundbreaking development in cryptography that enable secure information verification without exposing the actual data. Since their introduction in the 1980s, ZKPs have transformed various applications, particularly in authentication and privacy protection. By allowing a prover to demonstrate the truth of a statement to a verifier without revealing any additional information,

ZKPs maintain the confidentiality of sensitive data. Various forms of ZKPs, such as interactive and non-interactive proofs, address different security and efficiency requirements. Despite their potential, ZKPs face challenges including high computational complexity, verification costs, and the need for advanced hardware. Ongoing research aims to overcome these hurdles, enhancing the practical application and scalability of ZKPs in securing digital communications and protecting privacy in an increasingly interconnected world.

References

[1] S. M. Shlaka, and H. B. A. Wahab, "The zero-knowledge proof technique: Limitations and challenges," *2023 Second International Conference on Advanced Computer Applications (ACA)*, Misan, Iraq, 2023, pp. 90-95, doi: https://doi.org/10.1109/ACA57612.2023.10346931.

[2] E. Morais, T. Koens, C. van Wijk, and A. Koren, "A survey on zero knowledge range proofs and applications," *SN Appl. Sci.*, vol. 1, p. 946, 2019, doi: https://doi.org/10.1007/s42452-019-0989-z.

[3] C. Rackoff, and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in J. Feigenbaum (Ed.), *Advances in Cryptology — CRYPTO '91. CRYPTO 1991. Lecture Notes in Computer Science*, vol 576. Springer, Berlin, Heidelberg, 1992, doi: https://doi.org/10.1007/3-540-46766-1_35.

[4] S. M. Shlaka, and H. B. A. Wahab, "The zero-knowledge proof technique: Limitations and challenges," *2023 Second International Conference on Advanced Computer Applications (ACA)*, Misan, Iraq, 2023, pp. 90-95, doi: https://doi.org/10.1109/ACA57612.2023.10346931.

[5] B. A. Bryanton, "Introduction to privacy-enhancing cryptographic techniques: Zero knowledge proof – Proving something without exchanging evidence," Canada Revenue Agency, (n.d.).

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[7] U. Feige, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity," *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 1987, pp. 210-217, doi: https://doi.org/10.1145/28395.28419.

[8] F. Hamila, M. Hamad, D. C. Salgado et al., "Enhancing security in Fiat–Shamir transformation-based non-interactive zero-knowledge protocols for IoT authentication," *Int. J. Inf. Secur.*, vol. 23, pp. 1131-1148, 2024, doi: https://doi.org/10.1007/s10207-023-00779-8.

[9] M. Agal, K. P. Kishan, R. Shashidhar, S. S. Vantmuri, and P. Honnavalli, "Non-interactive zero-knowledge proof based authentication," *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, Hassan, India, 2021, pp. 837-843, doi: https://doi.org/10.1109/MysuruCon52639.2021.9641514.

[10] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, ACM, 1988, pp. 103-112.

[11] K. Yang, and X. Wang, "Non-interactive zero-knowledge proofs to multiple verifiers," *Cryptology*, ePrint Archive, Paper 2022/063, 2022.

[12] I. Santoso, and Y. Christyono, "zk-SNARKs as a cryptographic solution for data privacy and security in the digital era," *International Journal of Mechanical Computational and Manufacturing Research*, vol. 12, pp. 53-58, 2023, doi: https://doi.org/10.35335/computational.v12i2.122.

[13] A. Berentsen, J. Lenzi, and R. Nyffenegger, "A walk-through of a simple zk-STARK proof," Dec. 21, 2022.

[14] B. Bunz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs:

Short proofs for confidential transactions and more," *2018 IEEE Symposium on Security and Privacy (SP),* 2018, pp. 315-334, doi: https://doi.org/10.1109/SP.2018.00020.

[15] A. Gabizon, Z. J. Williamson, and O.-M. Ciobotaru, "PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge," *IACR Cryptology ePrint Archive*, 2019, 953.

[16] V. Botta, and I. Visconti, "Doubly adaptive zero-knowledge proofs," *Theoretical Computer Science*, vol. 968, p. 114014, 2023, doi: https://doi.org/10.1016/j.tcs.2023.114014.

[17] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, "Interactive proofs and the hardness of approximating cliques," *J. ACM*, vol. 43, no. 2, pp. 268-292, Mar. 1996, doi: https://doi.org/10.1145/226643.226652.

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

[19] M. V. G. da Silva., L. M. Zatesko, A. F. B. Costa, and H. Hepp, "The hidden subgroup problem and non-interactive perfect zero-knowledge proofs," 2023, doi: https://doi.org/10.5753/etc.2023.230017.

[20] X. Yang, and W. Li, "A zero-knowledge-proof-based digital identity management scheme in blockchain," *Computers & Security*, vol. 99, p. 102050, 2020, ISSN 0167-4048.

[21] S. Xie, W. Yao, F. Wu, and Z. Zheng, "Non-interactive zero-knowledge proof scheme from RLWE-based key exchange," *PLOS ONE*, vol. 16, pp. 1-19, 2021, doi: https://doi.org/10.1371/JOURNAL.PONE.0256372.

[22] M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu, "Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications," in A. Boldyreva, and D. Micciancio (Eds.), *Advances in Cryptology – CRYPTO 2019. CRYPTO 2019. Lecture Notes in Computer Science*, vol. 11692, Springer, Cham, 2019, doi: https://doi.org/10.1007/978-3-030-26948-7_5.

[23] T. Bai, Y. Hu, J. He, H. Fan, and Z. An, "Health-zkIDM: A healthcare identity system based on fabric blockchain and zero-knowledge proof," *Sensors*, vol. 22, no. 20, p. 7716, 2022, doi: https://doi.org/10.3390/s22207716.

[24] H. Zheng, L. You, and G. Hu, "A novel insurance claim blockchain scheme based on zero-knowledge proof technology," *Computer Communications,* vol. 195, pp. 207-216, 2022, ISSN: 0140-3664.

[25] O. Kuznetsov, A. Rusnak, A. Yezhov, D. Kanonik, K. Kuznetsova, and S. Karashchuk, "Enhanced security and efficiency in blockchain with aggregated zero-knowledge proof mechanisms," *IEEE Access*, pp. 1-1, 2024.

[26] J. Hasan, "Overview and applications of zero knowledge proof (ZKP)," *International Journal of Computer Science and Network*, vol. 8, no. 5, pp. 436-440, 2019, ISSN: 2277-5420.

[27] B. A. Bryanton, "Introduction to privacy-enhancing cryptographic techniques: Zero knowledge proof – Proving something without exchanging evidence," Canada Revenue Agency, (n.d.).

[28] 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.

[29] J. Kurmi, and A. Sodhi, "A survey of zero-knowledge proof for authentication," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, no. 1, 2015.

[30] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," *2016 IEEE Symposium on Security and Privacy,* 2016.

[31] J. Burleson, M. Korver, and D. Boneh, "Privacy-protecting regulatory solutions using zero-knowledge proofs," 2022.

[32] zk-SNARKs. [Online]. Available: https://z.cash/technology/zksnarks/