A Technology Review of Zero Knowledge Proof Techniques

Seyed Mohsen Rostamkolaei Motlagh[1], Claus Pahl[1][a], Hamid R. Barzegar[1] and Nabil El Ioini[2][b]

[1] *Free University of Bozen-Bolzano, 39100 Bolzano, Italy*

[2] *University of Nottingham, 43500 Semenyih, Malaysia*

Keywords: ZKP, Authentication, Distributed Systems, Technology Review.

Abstract: Distributed systems, particularly in IoT, require robust privacy-preserving authentication mechanisms to address increasing concerns about data security and integrity. Zero-Knowledge Proofs (ZKPs) have emerged as a promising solution to balance security, privacy, and efficiency. This paper reviews and compares state-of-the-art ZKP protocols, focusing on their suitability for decentralized, resource-constrained environments. We propose a comprehensive evaluation framework and apply it to zk-SNARK, zk-STARK, and Bulletproof protocols, analyzing metrics such as scalability, efficiency, and proof size. Our findings provide actionable insights into the trade-offs between these protocols, offering guidance for their application in IoT systems.

1 INTRODUCTION

Zero-knowledge proofs (ZKPs) are a cryptographic tool that can enhance the security of IoT devices. ZKPs enable one party (the prover) to prove to another party (the verifier) that a given statement is true without revealing any additional information beyond the truth of the statement itself. This characteristic is particularly valuable in distributed applications, where it is often necessary to verify identities and transactions while preserving the privacy of the underlying data. Traditional authentication mechanisms often require sharing or exposing some level of information that could potentially be intercepted or misused. ZKPs mitigate this risk by proving the validity of a claim without disclosing any additional data.

Scalability Issues: distributed systems can have larger numbers of nodes and devices, each requiring authentication. Centralized systems may struggle to handle the high volume of authentication requests, leading to latency issues (Yang et al., 2017).

Single Point of Failure: Centralized authentication systems create a single point of failure (Roman et al., 2013).

Intermittent Connectivity: Many devices and nodes, e.g., in mobility scenarios, may experience intermittent connectivity (Atzori et al., 2010).

Resource Constraints: Many devices and nodes have limited resources. Traditional authentication may require extensive cryptographic operations and can be resource-intensive (Mukherjee et al., 2017).

To address these challenges, ZKPs offer secure authentication without revealing any sensitive information, facilitating the following in distributed, decentralised contexts:

Scalable Authentication: ZKPs support decentralized authentication mechanisms, reducing the reliance on central servers and enhancing scalability.

Resilience to Connectivity Issues: ZKPs can enable authentication protocols that do not require constant connectivity to a central server, making them suitable for mobile and intermittently connected devices.

Efficiency for Resource-Constrained Devices: Certain ZKP protocols are designed to be computationally efficient, making them suitable for devices with limited resources.

Enhanced Privacy: ZKPs do not disclose sensitive information during the authentication, addressing privacy concerns in open environments (Goldwasser et al., 2019).

We evaluate different ZKP techniques to determine their suitability for the above challenges, considering factors such as efficiency, scalability, and security (Werth et al., 2023a; Werth et al., 2023b).

We evaluate different ZKP techniques based on a comprehensive assessment framework to cover various ZKP protocols. The evaluation will consider factors such as efficiency, scalability, and security.

[a] https://orcid.org/0000-0002-9049-212X

[b] https://orcid.org/0000-0002-1288-1082

Rostamkolaei Motlagh, S. M., Pahl, C., Barzegar, H. R. and El Ioini, N.
A Technology Review of Zero Knowledge Proof Techniques.
DOI: 10.5220/0013269600003944

In *Proceedings of the 10th International Conference on Internet of Things, Big Data and Security (IoTBDS 2025)*, pages 245-253
ISBN: 978-989-758-750-4; ISSN: 2184-4976

2 ZKP TECHNOLOGY BACKGROUND

A *trusted setup* refers to a phase in initialising a cryptographic protocol where certain parameters (often random values) are generated. These parameters are crucial for the security of the system. The key characteristic of a trusted setup is that the security of the entire system depends on the secrecy and proper disposal of initial data used during this setup phase (El Ioini and Pahl, 2018). In protocols that use a trusted setup, such as zk-SNARKs, the setup phase involves generating a set of public parameters that are used to construct and verify zero-knowledge proofs. Parameter Generation: During the trusted setup, The generation of initial cryptographic material might happen. This is used to derive public parameters for the ZKP system. Public and Private Data: Public parameters can be safely shared and are necessary for users to generate and verify proofs. Private data, if leaked, could be used to forge proofs, and must be destroyed or kept secret. Proof Generation and Verification: Once the parameters are set up, users can generate proofs that demonstrate their knowledge of certain information without revealing the information itself. Verifiers use the public parameters to check proof validity. The main security concern with a trusted setup is the *trust assumption* itself. Participants must trust that the setup ceremony was conducted honestly and that all copies of the toxic waste were destroyed. A reliance on a few during the setup creates a central point of failure. Corruption or coercion of participants in the trusted setup phase can lead to security failures.

ZKPs can be classified into *interactive and non-interactive*. *Interactive zero-knowledge proofs (IZKPs)* involve a series of back-and-forth communications where the verifier sends challenges to the prover, who responds with answers that demonstrate knowledge of a secret, without revealing the secret itself. *Non-interactive zero-knowledge proofs (NIZKPs)* require only a single message from the prover to the verifier if interaction is not feasible.

2.1 Interactive ZKPs

Fiat-Shamir Protocol: Conceptualized as an interactive proof system, the protocol involved a dynamic exchange between the prover and the verifier where the verifier would send random challenges to the prover, who, in turn, would respond in a way that convincingly demonstrated their knowledge of a secret without revealing the secret itself. Fiat and Shamir devised an adaptation known as the Fiat-Shamir heuristic. This non-interactive version used a cryptographic hash function to simulate the verifier's random challenges. It is used for digital signatures and secure authentication systems (Fiat and Shamir, 1986).

Schnorr Protocol: builds on proving the possession of a discrete logarithm, being effective in digital signatures and identity verification. The protocol operates in an interactive setting where the verifier sends a random challenge to the prover. The prover must demonstrate knowledge of a secret discrete logarithm in response to this challenge (Schnorr, 1990).

Guillou-Quisquater Protocol: is an interactive ZKP system designed for RSA-like cryptographic settings. This protocol is used to prove knowledge of k-th roots modulo a composite number. Specifically, it allows a prover to demonstrate that they possess a secret value. It serves as a foundation for RSA-based interactive identification schemes, providing a secure method for identity verification by ensuring that the verifier can be convinced of the prover's knowledge without gaining any additional information about the secret (Guillou and Quisquater, 1988).

Feige-Fiat-Shamir Protocol: Building upon the Fiat-Shamir protocol, this improves security by utilizing multiple secret values, thus strengthening authentication. It allows a prover to demonstrate their identity without revealing their secrets. This approach provides a more secure and robust method for identity verification, using principles of ZKP to ensure that the verifier is convinced of the prover's identity without any additional information (Fiege et al., 1987).

Graph Isomorphism Protocol: can demonstrate that two graphs are isomorphic without revealing the isomorphism itself. This protocol involves multiple rounds in which the verifier challenges the prover to demonstrate the isomorphism of randomly permuted graphs. The verifier sends a randomly permuted version of one graph, and the prover must respond by showing the isomorphism between permuted and original graphs (Goldwasser et al., 2019).

2.2 Non-Interactive ZKPs

zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge): represent a cryptographic tool that allows a prover to demonstrate possession of certain information without the need for interactive verification. One of the key features of zk-SNARKs is conciseness, i.e., proofs are both small in size and quick to verify. This makes zk-SNARKs well-suited for applications in blockchain technologies (Ben-Sasson et al., 2014; Pahl and El Ioini, 2019; Berenjestanaki et al., 2023).

zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge): are similar to zk-SNARKs with some differences. One of the main features of zk-STARKs is their transparency, i.e., not requiring a trusted setup. Additionally, zk-STARKs are designed to be post-quantum secure, making them resistant to potential future attacks by quantum computers. These characteristics make zk-STARKs an emerging technology in the blockchain space, where they are used to enhance scalability and security without compromising transparency.

Bulletproofs: are a type of non-interactive zero-knowledge proof that stand out for their compactness and efficiency. Unlike many other zero-knowledge proofs, Bulletproofs do not require a trusted setup phase, making them more practical and secure. They are particularly noted for their effectiveness in range proofs, which are essential in verifying that a secret value lies within a certain range without revealing the value itself. This feature is crucial in the context of cryptocurrencies citebunz2018bulletproofs.

Groth-Sahai Proofs: provide an efficient way to construct non-interactive zero-knowledge proofs for statements involving bilinear maps. These proofs are particularly valuable in cryptographic protocols that require the verification of complex relationships between elements, such as ciphertexts in attribute-based encryption schemes. By leveraging the properties of bilinear maps, Groth-Sahai proofs enable the secure and efficient verification of cryptographic operations (Groth and Sahai, 2008).

ZKBoo/ZKB++: is a framework designed to construct non-interactive zero-knowledge proofs that are both efficient and scalable. It achieves this by allowing the verification of computations without revealing the underlying data or software. ZKB++ builds upon the ZKBoo framework, introducing optimizations that reduce proof size and computational overhead. This makes ZKB++ particularly suitable for applications that require the secure outsourcing of computations (Giacomelli et al., 2016).

Ligero: is a lightweight zero-knowledge proof protocol designed to be both scalable and efficient, reducing the computational and communication overhead compared to other protocols like SNARKs or STARKs. Its design focuses on minimizing proof size and verification complexity, making it a strong candidate for applications requiring efficient cryptographic proofs citeames2017ligero.

PLONK: (Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge) is a SNARK that simplifies proof creation and verification using a single reference string for any computation. This eliminates the need for multiple trusted setups for different computations, enhancing versatility and efficiency. PLONK can improve scalability and privacy (Gabizon et al., 2019).

2.3 Other Variants

One advancement is auxiliary-input zero-knowledge (Goldreich and Oren, 1994). This form of zero-knowledge proof addresses scenarios where the verifier may possess prior knowledge related to the assertion. The definition ensures that the proof protocol safeguards against any additional information leakage, even if the verifier starts with auxiliary information. Another advancement is the blackbox-simulation zero-knowledge, which further refines the security model of zero-knowledge proofs. In this approach, the proof's security is tested by simulating the interaction between the prover and the verifier using a 'black box' method, ensuring the verifier cannot distinguish between the simulated interaction and the actual proof process. Boyar, Friedl, and Lund (Boyar et al., 1991) introduce new techniques for constructing ZKPs that are not only efficient but also reduce the computational demands on the prover. Their work addresses a key challenge in the practical implementation of zero-knowledge proofs by reducing the complexity and power requirements, making these proofs accessible for everyday cryptographic applications.

2.4 Integrating Blockchain and ZKPs

To address distributed systems challenges, integrating blockchain and zero-knowledge proofs (ZKP) into access control systems has been proposed. Blockchain technology offers a decentralized framework that enhances data integrity and security by distributing the control and verification of transactions across multiple nodes. This eliminates the need for a central authority and makes the system more resilient to attacks. Every transaction or access request is logged on an immutable distributed ledger, ensuring transparency and reducing data tampering (Song et al., 2021).(Alkhamisi and Alboraei, 2019) further support this by illustrating how decentralized systems can alleviate the credibility problem posed by third-party information concentration.

ZKPs allow a party to prove they have certain knowledge without revealing the knowledge itself. This method is particularly beneficial for maintaining privacy in access control systems. For instance, Song et al. (2021) proposed a model where access permissions are managed using encrypted tokens based on ZKP. This ensures that user identities remain hidden, and the attributes required for access are not exposed, thereby enhancing privacy. Jedlicka and Grant

(2022) also highlight the potential of ZKP in maintaining data privacy, noting that it allows verification without data exposure (Jedlicka and Grant, 2022).

Smart contracts, which are self-executing contracts with the terms of the agreement directly written into code, play a vital role in improving the efficiency of IoT access control systems. By implementing access control policies as smart contracts on the blockchain, these systems can automate policy enforcement, thus reducing the computational burden on IoT devices and optimizing resource usage (Song et al., 2021). (Lin et al., 2023) add that smart contracts can be particularly effective in handling high-traffic environments by batching authorization requests, thereby improving overall system efficiency.

3 COMPARATIVE ZKP ANALYSIS

We first define the comparison criteria, then describe the selection of protocols and the comparison process, before presenting the results of the comparison.

3.1 ZKP Comparison Criteria

The selection of ZKP techniques was guided by several key criteria to ensure their suitability for distributed systems environments in general and constrained, decentralised ones in particular. We introduce *criteria* together with suitable *metrics*.

Scalability: The ability of the ZKP technique to manage a large number of devices and high volumes of data efficiently. Scalability is important in where a large number of devices or nodes may be connected. In practice, scalability is measured by evaluating the *proof generation and verification times* as the number of devices and data size increase. We also assess the system's ability to maintain performance under different network conditions and loads.

Efficiency: The computational and communication efficiency of ZKP, especially in terms of proof generation and verification times. This is important for real-time applications and for devices with limited processing power and energy resources. Efficiency is measured by the *time taken to generate and verify proofs*, the *computational resources are required*, as well as the *communication overhead* associated with sending proofs. Lower values show higher efficiency.

Security Robustness: The robustness of ZKP against various types of attacks, including quantum attacks, is necessary to maintain integrity and confidentiality of the data. Ensuring the security of the proofs is important to protect against unauthorized access and tampering. Security is assessed based on the *theoretical guarantees* provided by the ZKP technique, such as resistance to known attack vectors.

No Trusted Setup: Techniques that do not require a trusted setup phase are preferred to eliminate the need for a trusted third party and reduce the risk of compromised security. Setup requirements are evaluated based on the *complexity, duration, and necessity of a trusted setup phase*. Techniques without trusted setup requirements are considered more favourable.

Compact Proof Size: Smaller proof sizes are advantageous for IoT devices with limited storage capabilities and for reducing the communication overhead in constrained network environments. *Proof size* is measured in bytes. Smaller proof sizes are preferable as they minimize storage and bandwidth needs.

Privacy-Preserving Capabilities: The ability of ZKP to maintain privacy by not revealing any information beyond the validity of the statement being proved. *Privacy-preserving capabilities* are assessed based on the *theoretical foundations* of the ZKP technique and *practical evaluations of the information leakage* during proof generation and verification.

Applicability to Mobility Scenarios: Assessing how well the ZKP technique can be integrated into mobile scenarios, such as connected vehicles and wearable health monitors. Mobility scenarios introduce additional challenges such as dynamic network environments and frequent handovers. Techniques that perform well in these scenarios ensure reliable and secure operation. Applicability is evaluated based on the *performance* of the ZKP technique in simulated mobility scenarios, considering factors such as *proof generation and verification times, communication overhead, and resilience to network changes*.

3.2 Selection and Comparison Process

The process of selecting the ZKP techniques involved a comprehensive review of existing ZKP protocols and evaluating them against the aforementioned criteria. The selection process included the following:

Literature Review: An extensive review of the literature on various ZKP techniques was conducted to identify potential candidates. This included reviewing academic papers, technical reports, and industry publications.

Evaluation of Features: The identified ZKP techniques were evaluated based on their features, such as proof generation and verification times, security properties, and whether they required a trusted setup.

Comparison and Analysis: The techniques were compared against each other based on the selection criteria. Techniques that best met the criteria were shortlisted for evaluation.

Two tables provide a comparative analysis of several ZKP techniques, highlighting their key features and suitability for IoT applications. Table 1 focuses on technical aspects, including type of ZKP, communication cost, proof size, and setup requirements (Ben-Sasson et al., 2018). These attributes determine efficiency and practicality of each ZKP technique in applications where bandwidth and storage are limited. Table 2 details potential applications, advantages, and disadvantages of each ZKP technique (Bünz et al., 2018; Sun et al., 2021; Ames et al., 2017; Gabizon et al., 2019; Groth and Sahai, 2008). This information provides a comprehensive overview of the suitability of each technique for specific IoT use cases and highlights the trade-offs involved in using each method.

3.3 Selected ZKP Protocol Analysis

We now detail three significant ZKP protocols: zk-SNARKs, zk-STARKs, and Bulletproofs. These protocols have substantial applications in areas such as blockchain technology, privacy-preserving computations, and secure communications. We explore each protocol covering the following: *Working Mechanism*: in-depth explanation of how the protocol operates, detailing the main phases involved. *Proof Generation and Verification Times*: discussion on the efficiency of the protocol in terms of proof generation and verification times. *Comparative Metrics*: comparison highlighting strengths and trade-offs of the protocol in terms of proof size, generation, and verification times. *Advantages and Disadvantages*: analysis of the protocol's main benefits and limitations.

3.3.1 ZK-SNARKs

zk-SNARKs allow a prover to convince a verifier that a particular statement is true without revealing any additional information, thereby maintaining the confidentiality of the data involved.
Working Mechanism of zk-SNARKs: In the setup phase, a trusted party generates a Common Reference String (CRS), a set of cryptographic parameters essential for both the proving and verification processes. This CRS is critical as it ensures the integrity and security of the subsequent proofs. During the proving phase, the prover uses the CRS to create a proof that a given statement is true. This process involves encoding the statement and the computation that verifies it into a succinct proof. The challenge here is to transform potentially large and complex data into a compressed form that retains its validity and can be efficiently verified. In the verification phase, the verifier uses the CRS and the proof to check the validity of the statement. This phase is designed to be extremely efficient, enabling quick verification of the proof without requiring the verifier to repeat the original computation. This efficiency is crucial for applications requiring real-time verification, such as blockchain transactions.
Proof Generation and Verification Times: The time required to generate a zk-SNARK proof can vary based on several factors, including the complexity of the computation, the optimization of the implementation, and the hardware used. More complex computations naturally require more time to generate proofs.
Comparative Metrics: When considering the *proof generation time, verification time, and proof size* of these implementations, the differences highlight the strengths and trade-offs of each approach.

3.3.2 ZK-STARK

zk-STARKs offer an advancement by eliminating the need for a trusted setup and providing a proof system that scales efficiently with large data sets.
Working Mechanism of zk-STARKs: The zk-STARK protocol consists of several key phases: setup, proving, and verification. These phases are designed to ensure that the system is both scalable and secure, without the need for any initial trusted setup. 1) Transparent Setup. Unlike zk-SNARKs, zk-STARKs do not require a trusted setup. Instead, they utilize publicly verifiable randomness, ensuring transparency and eliminating the risks associated with a trusted setup. This is achieved through the use of cryptographic primitives that are publicly known and verifiable. 2) Interactive Oracle Proofs (IOPs). zk-STARKs leverage IOPs to achieve scalability and efficiency. IOPs allow the prover to interact with the verifier through a series of queries to an oracle, which provides the necessary information to validate the proof. This interaction can be structured to ensure that the proof remains succinct and easy to verify. 3) Low-Degree Testing and Error-Correcting Codes. The protocol employs low-degree testing and error-correcting codes to ensure that the proofs are both correct and resistant to errors. These techniques allow zk-STARKs to maintain integrity even in the presence of noisy data or computational errors.
Steps in zk-STARK Protocol: The setup generates public parameters using publicly verifiable randomness. During proving, the prover constructs a proof by encoding computation and data for it to be efficiently verified in several rounds of interaction with the verifier, during which the prover responds to queries from the verifier's oracle. Verification involves the verifier checking the proof using the information provided by

Table 1: Comparative Analysis of ZKP Techniques – Part 1 Technical Focus.

ZKP	Type	Communication Cost	Proof Size	Setup Requirements
Groth's zkSNARK	Non-interactive	Low	Moderate	Requires a trusted setup
PLONK	Non-interactive	Low	Very Small	Does not require a trusted setup
FRI	Non-interactive	Very Low	Medium	Does not require a trusted setup
ZKBoo	Non-interactive	Very Low	Very Small	Does not require a trusted setup
Halo	General framework	Varies	Varies	Varies
Bulletproofs	Non-interactive	Low	Very Small	Does not require a trusted setup
zk-STARKs	Non-interactive	Medium	Large	Does not require a trusted setup
zk-SNARK	Non-interactive	Low	Moderate	Requires a trusted setup
Ligero	Non-interactive	Medium	Medium	Does not require a trusted setup

Table 2: Comparative Analysis of ZKP Techniques – Part 2 Applicability Focus.

ZKP	Potential Applications	Advantages	Disadvantages
Groth's zk-SNARK	Privacy-preserving cryptocurrencies, anonymous credentials	Efficient, proofs are relatively small	requires trusted setup, vulnerable to attacks if setup compromised
PLONK	Privacy-preserving cryptocurrencies, secure voting systems	Very efficient, proofs are very small	Can be less versatile
FRI	Privacy-preserving applications	Very efficient can handle complex computations	Can be difficult to implement
ZKBoo	Privacy-preserving applications	Very efficient, proofs are very small	Can be less versatile
Halo	Privacy-preserving cryptocurrencies, anonymous credentials	Flexible, versatile	Complex, requires expertise to implement
Bulletproofs	Confidential transactions, privacy-preserving applications	No trusted setup, compact proofs	Higher computational cost for proof generation
zk-STARKs	Large-scale computations, blockchain	Scalable, post-quantum secure	Larger proof sizes
zk-SNARK	Privacy-preserving cryptocurrencies, anonymous authent., secure voting	Efficient, proofs are relatively small	Requires a trusted setup, can be vulnerable to trust attacks
Ligero	Privacy-preserving applications	Efficient, medium-sized proofs	Can be less efficient in some use cases

the prover and the publicly known parameters. This is designed to be efficient, allowing for rapid verification of large computations.

Comparative Metrics: When comparing zk-STARKs to other cryptographic proof systems, several key metrics highlight their advantages: *Proof Generation Time:* zk-STARKs are designed to generate proofs efficiently, even for large data sets. The time required for proof generation scales logarithmically with the size of the data, making it feasible for real-world applications involving large amounts of data. *Verification Time:* The verification process is highly optimized, allowing for rapid verification of proofs. This efficiency is crucial for applications requiring real-time validation, such as blockchain transactions and large-scale data analysis. *Proof Size:* zk-STARKs produce much smaller proofs than the data they represent by using advanced cryptographic techniques, making proofs easy to store and transmit.

Advantages of zk-STARKs: *Scalability:* zk-STARKs are designed to handle large-scale data efficiently. The proof generation and verification processes are optimized to scale logarithmically with the size of the data, making zk-STARKs suitable for applications involving massive data sets. *Transparency:* By eliminating the need for a trusted setup, zk-STARKs enhance transparency and trustworthiness. The use of publicly verifiable randomness ensures that all parties can trust the setup process without relying on a single entity. *Post-Quantum Security:* zk-STARKs are designed to be secure against quantum computing attacks. This post-quantum security is achieved through the use of cryptographic primitives that are resistant to the capabilities of quantum computers. *Efficiency:* The efficiency of proof generation and verification makes zk-STARKs practical for real-world applications. The compact proofs and rapid verification are crucial for real-time validation.

3.3.3 Bulletproofs

Bulletproofs are a cryptographic protocol designed to facilitate range proofs, a mechanism that verifies whether a hidden numerical value falls within a predetermined interval without revealing the value itself. Introduced to enhance privacy and security, Bulletproofs are particularly important in financial systems and secure credential verification.

Working Mechanism of Bulletproofs: Bulletproofs operate without the need for a trusted setup phase, which contrasts with other ZKP systems like zk-SNARKs, using the Fiat-Shamir heuristic to achieve non-interactive zero-knowledge proofs.

Proof Construction: The prover generates vectors and scalars that encode the binary representation of the secret value x. A series of commitments and polynomials obfuscate the actual value while proving its legitimacy within the claimed range. This involves commitments to auxiliary vectors that mask the original input's structure.

Proof Generation: The prover uses Pedersen commitments, a cryptographic commitment scheme, to commit to the secret value x without revealing it. The commitment C is typically calculated as $C = g^x h^r$, where g and h are public generator points of a cyclic group, x is the secret value, and r is a random nonce.

Verification: The verifier checks these commitments against the public parameters to confirm the proof's validity. This involves ensuring that the inner product argument holds, proving that the committed value lies.

Proof Generation and Verification Times: Bulletproofs offer significant efficiency in proof generation and verification:

Proof Generation Time: The time to generate Bulletproofs scales logarithmically with the number of bits of the value. This efficiency is particularly beneficial in environments requiring rapid proof generation.

Verification Time: Although individual verification can be computationally intensive compared to some alternatives, batch verification is highly efficient, making Bulletproofs practical for systems processing large transaction volumes.

Proof Size: The proof sizes in Bulletproofs are significantly smaller, scaling logarithmically with the witness size, which reduces storage and bandwidth requirements.

Comparative Metrics: When compared to other cryptographic proof systems, Bulletproofs demonstrate the following advantages and trade-offs. *Efficiency:* Bulletproofs do not require a trusted setup and offer compact proofs, making them efficient for storage and transmission. *Verification Complexity:* While batch verification is efficient, individual verification can be more computationally intensive than protocols like zk-SNARKs. *Scalability:* Bulletproofs scale logarithmically with the witness size, making them suitable for large-scale applications, although the computational burden on the prover can be significant.

Advantages: (i) No trusted setup required, reducing the risk of malicious setup. (ii) Smaller proof sizes, enhancing storage and transmission efficiency. (iii) Efficient batch verification, beneficial for processing large volumes of transactions.

Disadvantages: (i) Individual verification is more complex and computationally intensive. (ii) Significant computational burden on the prover, especially in aggregated proofs or MPC settings. (iii) Potential scalability challenges due to computational complexities.

4 LIMITATIONS AND CHALLENGES

While the integration of blockchain and ZKP)in access control systems is promising, limitations and challenges remain for effective and widespread adoption in challenging contexts and in combination with blockchains.

Scalability and Performance: One of the primary challenges is the scalability of blockchain systems. Blockchain's decentralized nature, while beneficial for security and transparency, often results in high latency and low transactions per second (TPS). This is particularly problematic in high-traffic network environments where rapid and frequent transactions are required. Blockchain-based access control systems struggle with scalability due to low TPS and high latency (Lin et al., 2023).

Computational Overhead: The computational demands of zero-knowledge proofs can be a challenge for devices with limited processing power and energy supply. Implementing ZKP in resource-constrained environments requires efficient algorithms to minimize computational overhead. Lightweight ZKP implementations can ensure that devices can perform necessary computations without excessive energy consumption (Song et al., 2021).

Privacy and Data Integrity: Ensuring privacy and data integrity in a decentralized setting is complex. Although blockchain technology provides an immutable ledger, the public nature of blockchain can lead to potential privacy issues. ZKP can help maintain user privacy by allowing verification without data exposure (Jedlicka and Grant, 2022).

Integration with Existing Systems: Integrating blockchain and ZKP technologies with existing IoT systems is another challenge. Many current IoT devices and infrastructures are not designed to support the high computational and storage requirements of blockchain and ZKP (Alkhamisi and Alboraei, 2019).

Security Concerns: While blockchain and ZKP enhance security, they are not immune to all attacks. Potential vulnerabilities in cryptographic algorithms can be exploited, and ensuring robust security protocols is essential (Sun et al., 2021).
Cost of Implementation: The cost associated with implementing blockchain and ZKP technologies can be prohibitive. This includesfinancial costs but also time and resources required for development, deployment, and maintenance (Derei et al., 2023).

5 CONCLUSIONS

We explored the foundational concepts and developments in zero-knowledge proofs (ZKPs), a cryptographic technique that ensures the validity of a statement without revealing any additional information. We provided 1. Introduction to Zero-Knowledge Proofs as an overview of ZKPs, including their origin and fundamental principles. 2. Types of Zero-Knowledge Proofs providing a differentiation examination of the different types of ZKPs, including interactive and non-interactive proofs. 3. Classification of Common Protocols based on Defined Assessment criteria - following a systematic selection process. 4. Advanced Protocols like zkSNARKs, zkSTARKs, Bulletproofs, comparing their mechanisms, advantages, and limitations.

Our review shows that decentralisation or resource limitations create challenges such as scalability, computational overhead, privacy and data integrity, wider security concerns and cost to be addressed.

REFERENCES

Alkhamisi, A. O. and Alboraei, F. (2019). Privacy-aware decentralized and scalable access control management for iot environment. *Jrnl of King Abdulaziz Univ Comp and Inf Tech Sci*, 8(1):71–84.

Ames, S., Hazay, C., Ishai, Y., and Venkitasubramaniam, M. (2017). Ligero: Lightweight sublinear arguments without a trusted setup. In *Conf Comp & Comm Sec*.

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Comp Netw*, 54(15):2787–2805.

Ben-Sasson, E., Bentov, I., Horesh, Y., and Riabzev, M. (2018). Scalable, transparent, and post-quantum secure computational integrity. *Crypt ePrint Arch*.

Ben-Sasson, E., Chiesa, A., Tromer, E., and Virza, M. (2014). Succinct {Non-Interactive} zero knowledge for a von neumann architecture. In *USENIX*.

Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N., and Pahl, C. (2023). Blockchain-based e-voting systems: a technology review. *Electronics*, 13(1):17.

Boyar, J., Friedl, K., and Lund, C. (1991). Practical zero-knowledge proofs: Giving hints and using deficiencies. *Journal of cryptology*, 4:185–206.

Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., and Maxwell, G. (2018). Bulletproofs: Short proofs for confidential transactions and more. In *Sym Sec&Priv*.

Derei, T., Aulenbach, B., Carolino, V., Geren, C., Kaufman, M., Klein, J., Islam Shanto, R., and Korth, H. F. (2023). Scaling zero-knowledge to verifiable databases. In *Workshop on Verifiable DBS*.

El Ioini, N. and Pahl, C. (2018). Trustworthy orchestration of container based edge computing using permissioned blockchain. In *2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pages 147–154.

Fiat, A. and Shamir, A. (1986). How to prove yourself: Practical solutions to identification and signature problems. In *Theory and Appl of cryptographic Tech*.

Fiege, U., Fiat, A., and Shamir, A. (1987). Zero knowledge proofs of identity. In *Symp on Theory of computing*.

Gabizon, A., Williamson, Z. J., and Ciobotaru, O. (2019). Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *Cryptology ePrint Archive*.

Giacomelli, I., Madsen, J., and Orlandi, C. (2016). {ZKBoo}: Faster {Zero-Knowledge} for boolean circuits. In *USENIX*.

Goldreich, O. and Oren, Y. (1994). Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32.

Goldwasser, S., Micali, S., and Rackoff, C. (2019). The knowledge complexity of interactive proof-systems. In *Providing sound foundations for cryptography*.

Groth, J. and Sahai, A. (2008). Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*.

Guillou, L. C. and Quisquater, J.-J. (1988). A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *EUROCRYPT*, pages 123–128. Springer.

Jedlicka, J. and Grant, E. S. (2022). Data privacy through zero-knowledge proofs. In *ICERECT*.

Lin, X., Zhang, Y., Huang, C., Xing, B., Chen, L., Hu, D., and Chen, Y. (2023). An access control system based on blockchain with zero-knowledge rollups in high-traffic iot environments. *Sensors*, 23(7):3443.

Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., and Kumar, V. (2017). Security and privacy in fog computing: Challenges. *IEEE Access*, 5:19293–19304.

Pahl, C. and El Ioini, N. (2019). Blockchain based service continuity in mobile edge computing. In *IOTSMS*, pages 136–141.

Roman, R., Zhou, J., and Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Comp Netw*, 57(10):2266–2279.

Schnorr, C.-P. (1990). Efficient identification and signatures for smart cards. In *CRYPTO*.

Song, L., Ju, X., Zhu, Z., and Li, M. (2021). An access control model for the internet of things based on zero-knowledge token and blockchain. *Jrnl on Wireless Communications and Networking*, 2021(1):105.

Sun, X., Yu, F. R., Zhang, P., Sun, Z., Xie, W., and Peng, X. (2021). A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205.

Werth, J., Berenjestanaki, M. H., Barzegar, H. R., El Ioini, N., and Pahl, C. (2023a). A review of blockchain platforms based on the scalability, security and decentralization trilemma.

Werth, J., El Ioini, N., Berenjestanaki, M. H., Barzegar, H. R., and Pahl, C. (2023b). A platform selection framework for blockchain-based software systems based on the blockchain trilemma.

Yang, Y., Wu, L., Yin, G., Li, L., and Zhao, H. (2017). A survey on security and privacy issues in internet-of-things. *IEEE Internet of things Jrnl*, 4(5):1250–1258.