zk-IoT: Securing the Internet of Things with Zero-Knowledge Proofs on Blockchain Platforms

Gholamreza Ramezan
FidesInnova
g2n@fidesinnova.io

Ehsan Meamari
FidesInnova
e2i@fidesinnova.io

Abstract—This paper introduces the zk-IoT framework, a novel approach to enhancing the security of Internet of Things (IoT) ecosystems through the use of Zero-Knowledge Proofs (ZKPs) on blockchain platforms. Our framework ensures the integrity of firmware execution and data processing in potentially compromised IoT devices. By leveraging the concept of ZKP, we establish a trust layer that facilitates secure, autonomous communication between IoT devices in environments where devices may not inherently trust each other. The framework includes zk-Devices, which utilize functional commitment to generate proofs for executed programs, and service contracts for encoding interaction logic among devices. It also utilizes a blockchain layer and a relayer as a ZKP storage and data communication protocol, respectively. Our experiments demonstrate that proof generation and verification take approximately 694 and 19 milliseconds in our system setup, respectively. These timings meet the practical requirements for IoT device communication, demonstrating the feasibility and efficiency of our solution. The zk-IoT framework represents a significant advancement in the realm of IoT security, paving the way for reliable and scalable IoT networks across various applications, such as smart city infrastructures, healthcare systems, and industrial automation.

Index Terms—Zero-knowledge proof, Internet of Things, Blockchain.

I. INTRODUCTION

Internet of Things (IoT) refers to a network of interconnected physical devices, objects, or "things" that are embedded with sensors, software, and network connectivity, allowing them to collect and exchange data over the Internet. IoT applications range from items such as wearable devices, smart home appliances, and drones to more specialized industrial equipment and vehicles. IoT technology enables these devices to communicate with each other and with central systems, enabling data collection, monitoring, and automation.

A significant challenge facing IoT is ensuring trust. Without robust security mechanisms to ensure mutual trust among IoT devices, the functionality of each IoT
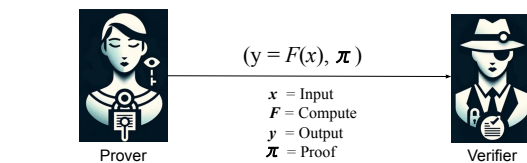

Fig. 1. Key components of zero-knowledge proofs (ZKPs).

device remains isolated, limiting the full potential that could be realized through integrated IoT solutions. An adversary may interfere with the device's firmware or modify its hardware or software to generate incorrect output from a legitimate device, posing a threat to the integrity of the device's computations. Additionally, the adversary may introduce a rogue IoT device to generate and transmit data to other users or IoT actuators, compromising the system's integrity. Hence, the concern about IoT *functions*, which is a result of executing its firmware or program, hinders the ability to fully leverage the interconnected capabilities of IoT device [1].

Addressing the above-mentioned challenge not only ensures the integrity of executed computations but also paves the way for autonomous IoT device communication. Such solution enables mutually distrustful IoT devices, that are from different manufacturers and are in separate IoT ecosystems such as Ring and Ecobee IoT systems, to perform distributed computations that run indefinitely, while ensuring that the correctness of every intermediate state of the computation can be verified efficiently. For instance, in the electric vehicle sector, represented by companies like Tesla, a decentralized trusting IoT framework would enable cars to securely communicate and receive real-time information about road conditions from vehicles traveling in the opposite direction. Additionally, it ensures that various components within a vehicle can authenticate whether commands received from the car's main computer are the
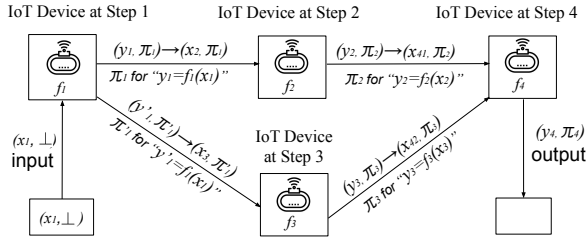
Fig. 2. A sample Internet of Things (IoT) devices setup.

result of legitimate computation (not hacked system), thereby establishing trust in the system's integrity.

The mentioned trust challenge can be addressed through the use of IoT devices built within a single IoT device manufacturer ecosystem. However, IoT devices are engineered across diverse ecosystems by various manufacturers, leading to a situation where the specific functionalities encoded within their firmware are not consistently disclosed. However, the principal features of their operational capabilities are generally recognized. Furthermore, companies may be reluctant to share detailed information about the architecture or firmware of their devices.

Zero-Knowledge Proof (ZKP) is a cryptographic technique enabling one party (the prover) to demonstrate the truth of a statement to another (the verifier) without revealing any information beyond the statement's truth itself. Fig. 1 illustrates the roles of the prover and verifier in a ZKP scenario. Here, the prover shares only the output of a specific function, without disclosing the input (secret), along with a proof. In the ZKP realm, *functional* commitment is a protocol that hides a function f from the verifier, yet allows the verifier to confirm the correctness of the function's execution without needing to rerun it or know the precise function [2], [3].

Tackling the issue of computational integrity within the IoT realm paves the way for the development of solutions that facilitate *autonomous device communication* among devices that may not inherently trust each other. Moreover, various traditional IoT attacks can be addressed such as the injection of malicious code that can alter processing results.

In automatic IoT device communication, trust in data generated by preceding steps is crucial. Proof-carrying data (PCD) allows composing proofs to validate computations, vital for scenarios where mutual trust among multiple parties (e.g., IoT devices) is absent. PCD constructs proofs certifying computation correctness while carrying relevant data, facilitating chained trustworthy computations. This concept is pivotal in decentralized

systems like blockchain, where trust is distributed [4]. Fig. 2 illustrates PCD in IoT. Starting with input x_1, IoT devices execute functions f_i, producing outputs and proofs. This ensures correct execution of IoT functions, maintaining trust throughout the chain of devices, and enabling a reliable data chain.

This paper proposes a framework designed to automate communication among untrusted IoT devices, thereby expanding IoT functionalities and enabling innovative IoT services that are not achievable by an isolated IoT device or a collection of IoT devices controlled by a central authority. The main contributions of this paper are as follows.

- We introduce the zk-IoT framework, designed for automatic communication between untrusted IoT devices. It comprises components including zk-Devices, blockchain nodes, and service contracts, as well as a blockchain protocol and a ZKP scheme.
- We introduce a ZKP scheme designed to ensure the integrity of IoT firmware execution through functional commitment (FC) within a distributed IoT setup.
- We also introduce several key concepts: a service contract, which encodes the interaction logic among IoT devices; zk-Devices, which utilize ZKP to generate proofs of IoT firmware execution; and a blockchain protocol, designed for automatic communication between untrusted IoT devices.
- We include a performance evaluation within a blockchain context, demonstrating the feasibility of implementing the zk-IoT framework.

This paper is structured as follows. In Section 2, we review the related works. Section 3 will introduce the proposed zk-IoT framework. In Section 4, we perform a performance analysis. Section 5 concludes.

II. RELATED WORK

This section provides an overview of existing ideas on how ZKP can empower IoT and introduces our proposal to further enhance IoT using ZKP.

Traditional authentication methods use robust encryption protocols to secure data transfer, ensuring protection from unauthorized access and interception. However, ZKPs revolutionize this approach. Chen et al. [5] review ZKP-based IoT authentication, highlighting its benefits over traditional methods in terms of anonymity and reduced computational complexity. Walsh et al. [6], proposed non-interactive ZKPs in a Merkle tree structure for authentication, along with methods combining ring signatures with ZKPs for enhanced privacy [7]. Wöhnert et al. [8] proposed

ZKP-based algorithms for verifiable identity in IoT, emphasizing accountability and integrity. Access control mechanisms in IoT using ZKPs have also been explored, leveraging Ethereum smart contracts and the Groth16 proof generation mechanism [9]. Verifiable anonymous identity management (VAIM) models have been proposed, focusing on privacy-preserving user-centric identity verification [10].

In the realm of blockchain-based IoT, challenges related to data immutability and General Data Protection Regulation (GDPR) compliance have prompted research into blockchain erasure models, combining ring signatures and ZKPs for enhanced privacy [11], [12]. Furthermore, balancing privacy with accountability in blockchain identity management is essential, as asserted by studies advocating for traceability services in ZKP-based systems [13], [14]. These systems combine authentication with privacy and traceability, using shared secrets among verification nodes and central authority oversight to detect and manage malicious behavior.

ZKPs can also be instrumental in proving the validity of the executed process without revealing the actual software code or detailed information about the process. This method improves security as it eliminates the risk of exposing software integrity mechanisms that hackers could potentially exploit to circumvent the attestation process. In [15], a novel approach for attested cameras is proposed to verify that images were taken by a camera. Zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) allows verification of image transformations non-interactively with only standard cryptographic hardness assumptions. Unfortunately, this work does not preserve input privacy, is impractically slow, and/or requires custom cryptographic arguments [16].

In addition to its previous applications, ZKP can be utilized to enable IoT devices to demonstrate that they have executed a particular function (firmware) on specific data, thereby ensuring the integrity of both the data and the function. This application, when combined with blockchain technology, enhances the capabilities of IoT for future applications. The focus of this paper is to propose an architecture that facilitates this enhancement.

III. zk-IoT Framework

The Zero-Knowledge Internet of Things (zk-IoT) framework ensures the integrity of firmware execution in IoT devices and facilitates automatic communication between untrusted IoT devices. Hence, it enables IoT devices to collaboratively offer innovative services to
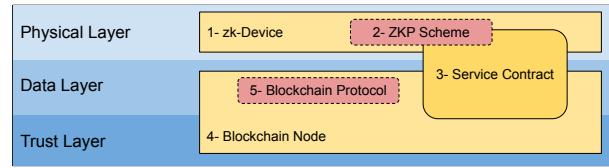

Fig. 3. The main components of the zk-IoT framework.

users by integrating functionalities and processed data across all devices. As it is shown in Fig. 3, the zk-IoT framework consists of five main elements: zk-Devices, ZKP scheme, service contracts, blockchain nodes, and blockchain protocol. The architecture is structured across three tiers. The top tier, known as the physical layer, houses the zk-Devices. The middle tier, or the data layer, includes the data relay mechanism, which facilitates data transfer and bridges the physical world with the blockchain realm. The bottom tier, the blockchain layer, serves as the trust foundation for the framework. It incorporates blockchain nodes to store ZKPs and uses smart contracts to facilitate the data transfer process.

A. zk-Device

In the zk-IoT framework, every IoT device is equipped with a ZKP generator to prove the integrity of its executed programs. Fig. 4, shows the inputs for the computation originating from IoT sensors, while the program is stored within the IoT device's memory.

The zk-Device proves that its output originates from a circuit designed by the manufacturer, certifying that the IoT device operates exactly as intended by its maker. It also guarantees that both the program and its inputs remain unchanged. As a result, the device's output is reliable and trustworthy. Upon completion of the computation, the output, along with the generated proof, is transmitted from the IoT device to the external physical world. It is important to note that the proof guarantees the execution of the program and the integrity of data inputted into the device either after analog-to-digital converters (ADC) or before digital-to-analog converters (DAC). This means that the boundary of the computation is confined to the digital part of the system where computations can be performed.

B. ZKP Scheme

zk-Devices employ a ZKP scheme using FC to verify a chain of computations across a series of IoT devices interacting during an automation process. As discussed in section I, the scheme enables mutually distrustful IoT devices to perform distributed computations while ensuring that the correctness of every intermediate state of the computation can be verified efficiently. The
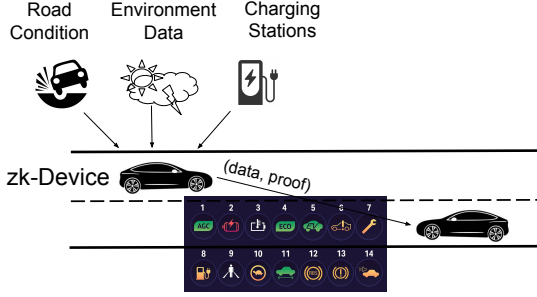
Fig. 4. zk-Device

scheme validates claims like $f^T(x_{11}, \ldots, x_{1t}) = y_T$ where t is the number of IoT devices and f^T denotes a computation of an automation process with T steps. Also, $(x_{11}, \ldots, x_{1t})$ is the input to the first IoT device in the automation process. y_T is the final process result. Without loss of the generality, the scheme is described on one IoT device at Step i of the computation with a private function f_i and public inputs $(x_{i1}, \ldots, x_{it})$ as shown in Fig. 5. This scheme has two phases:

- Phase 1: The prover seeks to establish that her computation is a function and is not a relation in a general form. A function guarantees that the rank one constraint system (R1CS) program generates a unique output for each input. However, in general, the set of matrices A, B, C in R_{R1CS} is not necessarily a functional set.

$$R_{R1CS}(n, h) = \{(A, B, C) \in (\mathbb{F}^{n \times n})^3, x \in \mathbb{F}^h,$$
$$w \in \mathbb{F}^{n-h} : (Az)o(Bz) = Cz,$$
$$z = (x, w)\}$$

where n is the number of constraints and $h \leq n$ is the number of instance variables. $\mathbb{F}$ is a field to which wires of the circuit carry their values. Also, operation o denotes the component-wise product of two vectors of equal size. We restrict R1CS to bounded sized arithmetic circuits and excludes non-functions. The relation of restricted R1CS is denoted as R_{R1CS-f} as following:

$$R_{R1CS-f}(n, t, s) = \{(A, B, C) \in (\mathbb{F}^{n \times n})^3, x \in \mathbb{F}^t,$$
$$y \in \mathbb{F}^s,$$
$$w \in \mathbb{F}^{n-t-s} : (Az)o(Bz) = Cz,$$
$$z = (x, w, y)\}$$

where t and s denote the input size and output size, respectively, with $h = s + t$. R_{R1CS-f} is a functional set $t-$Functional Triple ($t-$FT) that includes triples (A, B, C) so that A and B are strictly lower triangular matrices with t top zero rows (i.e., $t-$strictly lower triangular matrices). Also, C is a diagonal matrix with t top zero rows (i.e., $t-$diagonal matrix). The prover proves
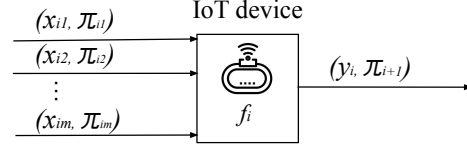

Fig. 5. A sample IoT device at step i in an automation process.

that matrices A_i, B_i and C_i corresponding to computation in Step i (f_i) are in the $t-$FT.

- Phase 2: The prover reliably opens function f_i without revealing any thing about f_i. An honest prover can prove to the verifier that the committed function f_i satisfies $f_i(x_{i1}, \ldots, x_{it}) = y_i$ for a given public $(x_{i1}, \ldots, x_{it})$ and y_i, without revealing any thing about f_i.

Combining the above two phases yields a secure functional commitment scheme structured as a tuple (SETUP, COMMIT, EVAL) as follows:

- SETUP$(1^\lambda) \to pp$: For a security parameter λ, produces the public parameter pp.
- COMMIT$(pp, A_i, B_i, C_i) \to C = (Com_{A_i}, Com_{B_i}, Com_{C_i})$: The Prover represents circuit of function f_i by matrices A_i, B_i, C_i in $\mathbb{F}^{n \times n}$ such that $f_i(x_{i1}, ..., x_{it}) = y_i$ iff there is $(w_{i1}, ..., w_{in-t-1})$ in $\mathbb{F}^{n-t-1}$ such that $(A_i z_i)o(B_i z_i) = C_i z_i$ where $z_i = (x_{i1}, .., x_{it}, w_{i1}, ..., w_{in-t-1}, y_i)$. This function produces commitment $Com_{A_i} = (Com_{Row_{A_i}}, Com_{Col_{A_i}}, Com_{Val_{A_i}})$ where:

  - $Com_{Row_{A_i}}$ is a commitment to polynomial Row_{A_i} that is made by non-zero elements in A_i such that $Row_{A_i}(\gamma^j) = \omega^{r_j}$ where r_j is the row number of jth non-zero element and γ is a generator of a multiplicative subgroup of field $\mathbb{F}$ of order m. Also, m is the maximum number of non-zero elements of matrices A_i, B_i and C_i. Also, ω is a generator of a multiplicative subgroup of field $\mathbb{F}$ of order n.
  - $Com_{Col_{A_i}}$ is a commitment to polynomial Col_{A_i} that is made by non-zero elements in A_i such that $Col_{A_i}(\gamma^j) = \omega^{c_j}$ where c_j is the column number of the jth non-zero element.
  - $Com_{Val_{A_i}}$ is a commitment to polynomial Val_{A_i} that is made by non-zero elements in A_i such that $Val_{A_i}(\gamma^j) = v_j$ where v_j is the value of jth non-zero element. The Com_{B_i} and Com_{C_i} are also made similarly.

- EVAL$(P_E(pp, f_i, (x_{i1}, ..., x_{it}), y_i),$
  $V_E(pp, C, (x_{i1}, ..., x_{it}), y_i)) \to \{0, 1\}$: An interactive protocol for P_E to convince V_E

that committed matrices A_i, B_i and C_i are in $t-$FT based on protocol "t-FT Test" in [3] and $f_i(x_{i1},...,x_{it}) = y_i$ based on construction 1 in [3].

C. Service Contract

Realizing automated IoT device communication requires a mechanism to encode the interaction logic among IoT devices. This is achieved through what we call a Service Contract. A Service Contract specifies the protocol for IoT device communication, detailing how data produced by one IoT device can be utilized by others. Additionally, it ensures the integrity of the data generated at each step by verifying the ZKPs produced by IoT devices engaged in a contract. Service contracts unlock the potential of IoT devices by amalgamating the functions of individual, isolated devices to create new, integrated functionalities. For example, employing a service contract to merge the functionalities of temperature sensors in a wide residential area with a siren in a fire department can create an effective fire detection system. Similarly, integrating a temperature display in a hydro company could lead to an efficient energy management system. These types of integrations serve as prime examples of how service contracts can innovatively augment the capabilities of IoT devices.

Algorithm 1 outlines the basic logic contained in a service contract, which processes a set of inputs including GPS coordinates, a timestamp, a road status (possible car-accident or road problem), the device type, and a proof verifying the integrity of the firmware execution that produced the input data. The process begins by verifying the proof. If the device is identified as a Tesla and the data's location is pinpointed to Seattle, the algorithm then forwards the road condition information and timestamp to the next device in the service chain. The practical implementation of this process for a multi-sensor system will be explored in the performance evaluation section, with examples using JavaScript programming.

Algorithm 1 Service Contract Pseudocode.

Input: Proof P, GPS L, Timestamp T_i, Collision C_i, Device Type D
Output: Temperature T_o, Temperature C_o
Require: $Verify(P) == valid$
Require: $Compliance(P, D,'Tesla') == valid$
  1: **if** city(L)==$'Seattle'$ and type(D)==$'Tesla'$ **then**
  2:     Return T_i as T_o
  3:     Return C_i as C_o

D. Blockchain Node

The blockchain node is a critical component in blockchain technology, responsible for executing blockchain algorithms, including consensus algorithms, smart contract execution, block creation, and transaction verification. It also maintains the blockchain data in what we call 'hot storage,' which is replicated across all nodes in the network.

Within the IoT domain, each device is required to transmit its physical data to a storage system and may receive commands from it. We call it 'cold storage'. This data exchange is secure and exclusive, establishing a direct trust relationship between a node and an IoT device.

In the zk-IoT framework, IoT functionalities are integrated into the blockchain node, creating a node that features two distinct databases for hot and cold storage. Hot storage is dedicated to blockchain data, such as ZKPs, transactions, and smart contracts. Cold storage, on the other hand, houses a local copy of data from zk-Devices and service contracts. Unlike hot storage, cold storage is not automatically shared across the network. Instead, data transfer between source and target nodes activated by a service contract.

E. Blockchain Protocol

In the context of zk-IoT, every automated interaction involves a series of steps designed to facilitate communication and execution among different IoT devices. These interactions are grounded in the fundamental premise of a straightforward exchange between two IoT devices. Without loss of generality, this section will focus on the interaction of only two devices under a service contract. The first device generates data, which the second device then receives, verifies, and uses as input for its own firmware execution, subsequently generating new data along with a ZKP for the next device.

The workflow of the blockchain protocol is detailed in Figure 6. The process is initiated by a service contract but requires sufficient tokens from the second device, which receives the data. To illustrate this, we describe the process beginning with zk-Device B depositing tokens into a smart contract distinct from the service contract. This dedicated smart contract serves solely as a reserve pool specifically for data transactions and is designed to hold and disburse tokens according to a predefined sequence.

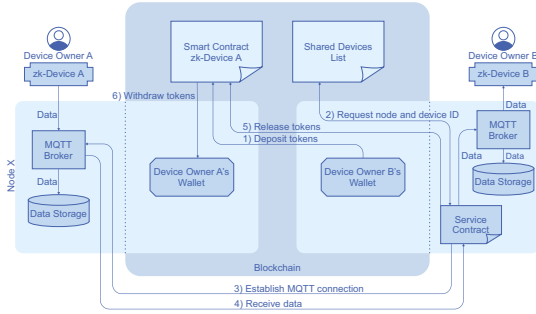 1) **Deposit Tokens:** The Device Owner B deposits sufficient tokens into the zk-Device A's smart contract.

Fig. 6. zk-IoT communication on blockchain layers.

Fig. 7. ZKP-enabled environmental sensor: general architecture.

Fig. 8. ZKP-enabled environmental sensor.

2) **Request node and device ID:** The service contract requests and retrieves the node and zk-Device A's ID from the Shared Devices List on the Blockchain.

3) **Establish MQTT connection:** The Service Contract establishes an MQTT connection between the Service Contract on Node Y and the MQTT broker on Node X.

4) **Receive data:** The service contract receives zk-Device A's data from the MQTT broker. If the set condition of the service contract is correct, it sends the data using MQTT Broker to the zk-Device B.

5) **Release tokens:** The service contract on Node Y sends a release tokens request to the zk-Device A's smart contract.

6) **Withdraw Tokens:** The Device Owner A requests a withdrawal from the smart contract and will receive the payment in their wallet.

IV. Performance Evaluation

We have developed a preliminary version of the zk-IoT system as described below. The service contracts were crafted using JavaScript due to available ZKP libraries like SnarkJS that enable the users to verify the proof while letting the user to encode the complex interactions required for automating processes across multiple IoT devices. The underlying blockchain network utilizes Ethereum technology Proof-of-Authority (PoA) consensus algorithm, supplemented by a specially modified blockchain node designed to run IsolatedVM, enabling it to execute JavaScript-based service contracts. Also, the blockchain node supports the Message Queuing Telemetry Transport (MQTT) protocol for the non-blockchain part to communicate with zk-Devices. To manage the system load, the service contract, through the MQTT broker, can send a command to enable or disable the proof generation function. Additionally, the
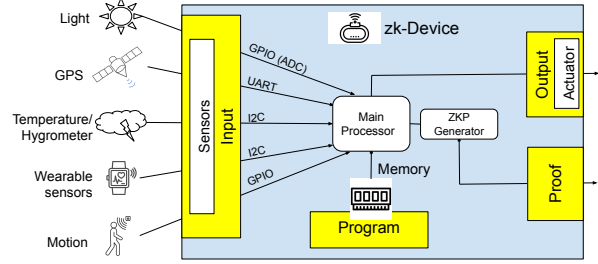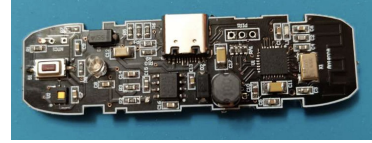
zk-Device is built around ESP32 microcontrollers with a DC power supply, integrating ZKP libraries.

Figures 7 and 8 showcase the schematic and photograph, respectively, of a ZKP-enabled environmental sensor hardware design. This design includes functionalities such as a temperature sensor, hygrometer, motion detector, and smart button. For the purposes of this performance evaluation report, the demonstration of ZKP functionality is confined to the smart button component, which generates a notification upon being pressed. The system has been developed using two ZKP methods: the halo2_proofs library (version 0.3.0) in Rust, and Circom. We implemented a straightforward communication protocol between two IoT devices.

Figure 9 presents the average prover and verifier times (based on the average of 100 experiments) for two distinct ZKP proving systems, Groth16 and Plonk. It is shown that Groth16 requires 694 milliseconds, which is less than Plonk's 776 milliseconds for prover time. Additionally, the verification times for Groth16 and Plonk are 19 milliseconds and 22 milliseconds, respectively. While Groth16 demonstrates faster processing times and a proof size of 856 bytes, it necessitates a trusted party for circuit-specific key generation. In contrast, Plonk benefits from a universal setup, allowing the one-time keys generated by the trusted party to be applied across various circuits, while the proof size is 2,349 bytes. Note that a multi-party computation (MPC) mechanism can be used to set up the keys.

Fig. 10 illustrates the time required for writing and reading proofs on the blockchain within the zk-IoT framework. Specifically, submitting the proof to the smart contract and receiving the first confirmation takes 11,969 milliseconds, and reading the proof re-
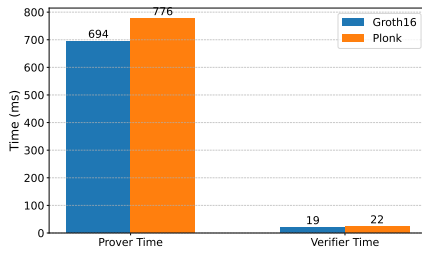
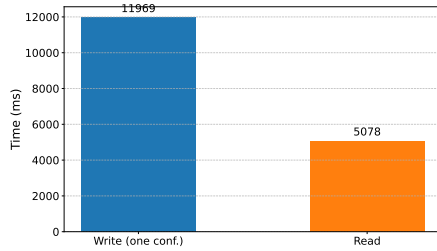Fig. 9. Prover and verifier time: Groth16 vs. Plonk comparison.

Fig. 10. Blockchain proof write and read time.

quires 5,078 milliseconds. These durations highlight that blockchain operations are considerably slower than proof generation and verification, underscoring the need for faster blockchain technology that is more compatible with the zk-IoT framework.

V. CONCLUSION

In this paper, we have introduced the zk-IoT framework as a groundbreaking development in enhancing the trust of IoT ecosystems, utilizing ZKP and blockchain technology. This innovative framework is adept at verifying the correctness of IoT firmware execution and ensuring data integrity, even in scenarios where devices may be compromised. Our approach also fosters new opportunities for dependable and autonomous interactions within IoT networks. These advancements are vital for the progress of IoT systems such as smart city, healthcare systems, and industrial automation.

The practical implementation and preliminary performance analysis of the zk-IoT framework, as demonstrated in our work, confirm its practicality and effectiveness. This signals a shift towards a more secure and trustworthy era of IoT operations. We plan to present the detailed performance evaluation in a subsequent paper for IoT devices featuring more complex codes and functionality, following the completion of a full implementation.

REFERENCES

[1] B. Wu, R. Zhong, Y. Wang, J. Wan, J. Zhang, and W. Shi, "Vpi: Vehicle programming interface for vehicle computing," *Journal of Computer Science and Technology*, 2024.

[2] L. de Castro and C. Peikert, "Functional commitments for all functions, with transparent setup and from sis," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 287–320, Springer, 2023.

[3] D. Boneh, W. Nguyen, and A. Ozdemir, "Efficient functional commitments: How to commit to a private function." Cryptology ePrint Archive, Paper 2021/1342, 2021. https://eprint.iacr.org/2021/1342.

[4] A. Chiesa, "Proof-carrying data," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 2010. Available at MIT DSpace: https://dspace.mit.edu/handle/1721.1/61151.

[5] Z. Chen, Y. Jiang, X. Song, and L. Chen, "A survey on zero-knowledge authentication for internet of things," *Electronics*, vol. 12, no. 5, 2023.

[6] M. Walshe, G. Epiphaniou, H. Al-Khateeb, M. Hammoudeh, V. Katos, and A. Dehghantanha, "Non-interactive zero knowledge proofs for the authentication of iot devices in reduced connectivity environments," *Ad Hoc Networks*, vol. 95, p. 101988, 2019.

[7] Z. Mahmood and J. Vacius, "Privacy-preserving block-chain framework based on ring signatures (rss) and zero-knowledge proofs (zkps)," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pp. 1–6, IEEE, 2020.

[8] K. H. Wöhnert, S.-J. Wöhnert, T. Thiel, R. Weißbach, and V. Skwarek, "Secure cyber-physical object identification in industrial iot-systems," *Procedia Manufacturing*, vol. 51, pp. 1221–1228, 2020.

[9] L. Song, X. Ju, Z. Zhu, and M. Li, "An access control model for the internet of things based on zero-knowledge token and blockchain," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–20, 2021.

[10] G. Ra, T. Kim, and I. Lee, "Vaim: verifiable anonymous identity management for human-centric security and privacy in the internet of things," *IEEE Access*, vol. 9, pp. 75945–75960, 2021.

[11] H. R. Elgabri, M. S. Hassanein, A. A. El-Sawy, and M. Taha, "Blockchain authentication access control framework to secure iot devices in smart cities," *Journal of Southwest Jiaotong University*, vol. 58, no. 4, 2023.

[12] D. C. Rani, S. G. Janakiraman, K. S. Chandra, E. P. Thangavel, G. A. Kothamasu, K. L. Bhaskaran, and G. Jayabalasamy, "A multi-round zero knowledge proof algorithm for secure iot and blockchain environments.," *International Journal of Safety & Security Engineering*, vol. 13, no. 4, 2023.

[13] I. Damgård, C. Ganesh, H. Khoshakhlagh, C. Orlandi, and L. Siniscalchi, "Balancing privacy and accountability in blockchain identity management," in *Topics in Cryptology – CT-RSA 2021* (K. G. Paterson, ed.), (Cham), pp. 552–576, Springer International Publishing, 2021.

[14] D. A. Luong and J. H. Park, "Privacy-preserving identity management system on blockchain using zk-snark," *IEEE Access*, vol. 11, pp. 1840–1853, 2023.

[15] D. Kang, T. Hashimoto, I. Stoica, and Y. Sun, "Zk-img: Attested images via zero-knowledge proofs to fight disinformation," *arXiv preprint arXiv:2211.04775*, 2022.

[16] S. Fang, Q. Liu, F. Zhang, N. Chen, and X. Li, "Application of internet of things and blockchain in information security and privacy protection of global organizations," *Journal of Organizational and End User Computing (JOEUC)*, vol. 35, no. 3, pp. 1–16, 2023.