IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 2 August 2024, accepted 17 August 2024, date of publication 26 August 2024, date of current version 3 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3450313

RESEARCH ARTICLE

Blockchain-Enhanced Zero Knowledge Proof-Based Privacy-Preserving Mutual Authentication for IoT Networks

ADITYA PATHAK[1], (Graduate Student Member, IEEE),
IRFAN AL-ANBAGI[1,2], (Senior Member, IEEE),
AND HOWARD J. HAMILTON[3]
[1]Faculty of Engineering and Applied Science, University of Regina, Regina, SK S4S 0A2, Canada
[2]Department of Electrical and Computer Engineering, College of Engineering, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada
[3]Department of Computer Science, University of Regina, Regina, SK S4S 0A2, Canada

Corresponding author: Irfan Al-Anbagi (irfan.al-anbagi@uregina.ca)

ABSTRACT Authentication in low-latency Internet of Things (IoT) networks must satisfy three requirements, namely, high security and privacy preservation, high scalability, and low authentication time. These requirements arise because devices in IoT networks must operate in a secure and scalable manner despite being limited in computational resources. Existing authentication mechanisms focus on the security and privacy of IoT networks but neglect the importance of scalability and authentication time. Therefore, existing authentication mechanisms are unscalable and unsuited to low-latency IoT networks. With a focus on increasing scalability and reducing the authentication time while providing high security and privacy preservation in low-latency IoT networks, we propose a mutual authentication mechanism called Zero-Knowledge Proof-based Privacy-Preserving Mutual Authentication (Z-PMA) for IoT networks. The Z-PMA mechanism utilizes a combination of a zero-knowledge proof, an incentive mechanism, and a permissioned blockchain to provide secure, privacy-preserving, scalable, low-latency authentication for IoT networks. We develop a new approach to address the trade-off between the three requirements for authentication mechanisms for low-latency IoT networks that has the potential to improve the overall performance of these networks. A permissioned blockchain is incorporated in the approach to provide secure and immutable data storage using its distributed and unforgeable ledger. Our experimental results show that the Z-PMA mechanism reduces authentication time than existing state-of-the-art authentication mechanisms, while providing high security and privacy preservation as well as high scalability.

INDEX TERMS Authentication, blockchain, IoT networks, privacy-preserving, zero-knowledge proof.

I. INTRODUCTION

The use of Internet of Things (IoT) technologies has expanded tremendously in recent years, with a wide range of applications that leverage the connectivity and data exchange capabilities of IoT devices. As IoT networks become increasingly decentralized with the help of blockchains, the need for robust and efficient authentication mechanisms increases. Authentication plays a vital role in ensuring the security and integrity of IoT networks by verifying the identities of devices, protecting against identity forgery, and preventing unauthorized access. However, existing blockchain-based authentication mechanisms in decentralized IoT networks often have limitations with respect to privacy preservation, scalability, and authentication time [1], [2], [3], [4], [5], [6]. Based on these limitations, the three main requirements for future authentication mechanisms in low-latency IoT networks are high security and privacy preservation, high scalability, and low authentication time. Authentication mechanisms should be secure and privacy-preserving because decentralization of IoT networks increases the attack surface, allowing adversaries to exploit multiple nodes at different

The associate editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero.

layers of the network. Since IoT networks are sometimes large and are expected to grow larger [7], the performance of authentication mechanisms should also scale well as the number of IoT devices increases. Authentication mechanisms should be designed to provide low authentication time to meet the requirements of various low-latency IoT applications, including healthcare, telemedicine, industrial IoT, and smart grids [7].

Neither traditional authentication mechanisms nor recent provide privacy preservation in terms of anonymity, unlinkability, and traceability. *Anonymity* ensures the identities and communication keys of IoT devices remain concealed. *Unlinkability* prevents the authentication messages from being traced back to the real identities of IoT devices. *Traceability* enables the identification and tracking of IoT devices, which is crucial for removing malicious devices from networks. Although traditional authentication mechanisms [8], [9], [10], [11] provide authentication, they may not preserve anonymity, unlinkability, and traceability [2], [4]. Some recent authentication mechanisms solve the anonymity problem by authenticating devices anonymously. Anonymous-based authentication mechanisms [12], [13], [14] provide strong anonymity, unlinkability, and traceability at the price of inefficiency. The existing mechanisms use bilinear mapping, group signatures, and other complex algorithms, which can impose heavy computational burdens on IoT devices. Other recent approaches, named pseudonym-based authentication mechanisms [15], [16], provide anonymity and traceability, but cannot guarantee the unlinkability of IoT devices, which may reveal their real identities if an adversary tracks long-term pseudonyms. Besides, these mechanisms must frequently change their pseudonyms to improve security, which increase the computational burden on IoT devices.

Zero-knowledge proof (ZKP) [17] is a cryptography tool that allows the provers to show their identity without providing any valuable information to the verifier. ZKP-based mechanisms are used in anonymous payments [18] and supply chains to keep consumption and transportation information private [19]. Current ZKP-based authentication mechanisms for IoT networks [1], [2], [4] provide security and privacy preservation, while not imposing a large computational burden on the IoT device. However, these mechanisms do not simultaneously fulfill two of the main requirements of authentication mechanism for a low-latency IoT network, which are high scalability and low authentication time.

Therefore, this paper proposes a novel Zero Knowledge Proof-based Privacy-Preserving Mutual Authentication (Z-PMA) mechanism to simultaneously achieve the three main requirements of authentication mechanisms for low-latency IoT networks. The Z-PMA mechanism uses ZKP and a permissioned blockchain to provide high security and privacy preservation in authentication, along with high scalability. The main reason that we use a blockchain network in the Z-PMA mechanism is to provide secure storage for the parameters that are used for mutual authentication. In addition, we use the permissioned type of blockchain to increase the scalability of the IoT network, because permissioned blockchain networks are more scalable than permissionless blockchain networks [20]. Furthermore, an incentive mechanism is provided to select additional authenticators (AAs) from the base station (BS) devices, which provide low authentication time and contribute to high scalability. The main operations of the Z-PMA mechanism are authentication and base station incentive operations. In the authentication operation, we use the quadratic residue (QR) technique to implement ZKP-based authentication between edge devices and IoT devices. This operation makes the IoT network secure, privacy-preserving, and scalable. In the base station incentive operation, we use concepts from contract theory to implement an incentive mechanism to offload the computation of authentication operations from edge devices to additional authenticators (selected BSs). This operation decreases authentication time and increases scalability. Overall, both operations in the Z-PMA mechanism make it suitable for low-latency IoT networks.

The main contributions of this paper can be summarized as follows:

1) We develop a novel Zero Knowledge Proof-based Privacy-Preserving Mutual Authentication (Z-PMA) mechanism, which uses a ZKP-based mutual authentication mechanism, a permissioned blockchain network, and an incentive operation to achieve the three main requirement of low-latency IoT network, which are high security and privacy preservation, high scalability, and low authentication time. A blockchain network is used to provide secure storage of the parameters used in the Z-PMA mechanism.
2) We develop a ZKP-based mutual authentication mechanism using the quadratic residue (QR) technique and a permissioned blockchain network to provide high security and privacy preservation as well as high scalability. This authentication mechanism is implemented as a smart contract in a blockchain network to secure the authentication process.
3) We develop a base station incentive operation using contract theory to provide low authentication time and contribute to high scalability.
4) We present a four-part analysis of the effectiveness of Z-PMA that includes an analysis of security requirements, a formal security analysis using Proverif, a simulation-based analysis, and an IoT testbed-based analysis.

The remainder of this paper is organized as follows. Section II discusses related work. Section III provides background knowledge. Section IV explains the network architecture and assumptions. Section V describes the design of the Z-PMA mechanism. Section VI presents experimental results and analysis. Finally, Section VII provides conclusions and describes future work.

II. RELATED WORK

In this section, we review existing research on authentication to determine its suitability for low-latency IoT networks. We divide the related work into privacy-preserving security mechanisms, authentication mechanisms, and privacy-preserving authentication mechanisms. To evaluate existing work, we focus on five key criteria: security, privacy preservation, scalability, authentication time, and suitability for low-latency IoT applications. A detailed comparison of the related work in terms of the security criterion is shown in Table 1. The security criterion is evaluated based on six subcriteria: mutual authentication (MA), secure incentive operation (SIO), secure key agreement (SKA), attack resistance (AR), immunity to malicious AAs (IMA), and non-repudiation (NR). Here, ✔ means that subcriteria is fulfilled and ✕ means that subcriteria is not fulfilled. In addition, a comparison of related work in terms of privacy preservation, scalability, authentication time, and suitability is shown in Table 2. The criteria for the parameters shown in Table 2 are evaluated based on the following information. Privacy preservation is evaluated based on four subcriteria: privacy (P), anonymity (A), traceability (T), and unlinkability (U). The authentication time is high or low depending on whether computationally expensive operations are performed on IoT devices or not because if the IoT device performs such operations, the authentication time will be high. Scalability is high or low, depending on whether the mechanism supports a large number of IoT devices or not. Suitability is high or low, depending on whether the cryptographic operations performed by IoT devices are lightweight (computationally inexpensive) or not. The remainder of this section describes the related work and justifies the ratings shown in the tables. For a quantitative analysis of some of these mechanisms, see Section VI, where the Z-PMA mechanism is compared to existing mechanisms.

Previous research provides a variety of privacy-preserving security mechanisms for IoT networks. Zöscher et al. [23] propose a security-based automatic fare collection system that anonymizes unique identifications for smart cards to conceal linkages with cardholder tagging and location patterns. Although the mechanism preserves the privacy of the location patterns between the smart card and the RFID reader, the data transmission between the public transport system and the RFID readers is in raw format. Furthermore, no encryption algorithm is used to keep stored RFID data secure and private, resulting in low privacy preservation. Kumar et al. [25] propose a lightweight Advanced Encryption Standard (AES) and a Secure Hash Algorithm 1 (SHA1)-based mechanism to maintain the privacy of content in smart homes. However, implementing AES operations on sensor devices leads to a high computation and memory overhead. Additionally, the use of a centralized service provider leads to a single point of failure and reduces the scalability of the mechanism. Dorri et al. [22] propose a decentralized privacy-preserving mechanism using Diffie-Hellman key exchange and hashing to overcome the single point of failure problem due to centralized processing and storage. However, the mechanism does not scale well because it requires a high-resource, always-online device known as a *miner*, which is responsible for all communication within the network. Ivacscu et al. [21] propose a security mechanism in a multi-agent architecture for privacy preservation in a healthcare system; in this mechanism, the data are transmitted to a central medical server for long-term storage. The use of a central server creates a Distributed Denial-of-service (DDoS) attack vulnerability and could lead to data loss caused by a single point of failure. The above-discussed privacy-preserving security mechanisms mainly use centralized servers, which lead to scalability problems and increase the chances of single point of failure problems in low-latency IoT networks. To tackle these problems, the work presented in this paper uses a distributed edge computing layer and a permissioned blockchain to create a secure, distributed, and decentralized mechanism that provides high privacy preservation and high scalability for low-latency IoT networks.

Recent papers propose a variety of authentication-based security mechanisms for IoT networks. Shivraj et al. [24] propose a one-time password (OTP) authentication mechanism for IoT networks. The mechanism employs Identity-Based Elliptic Curve Cryptography (IBE-ECC) to provide lightweight end-to-end authentication between IoT devices. However, IoT devices face increased computational complexity with increased OTP size. Kumar et al. [26] propose a lightweight, authentication-based session-key establishment mechanism for smart home-based IoT applications. The mechanism requires a security service provider, which is a trusted centralized server, to assign parameters, generate tokens, and distribute the tokens to the IoT devices. This mechanism provides low scalability because it uses a trusted centralized server, which leads to single point of failure. Gope et al. [27] propose a non-traceable authentication mechanism in a IoT network using hash functions and bitwise XOR operations. This mechanism provides low security and privacy preservation, as well as low scalability. Ying et al. [28] propose an anonymous, lightweight authentication mechanism for vehicle ad-hoc networks (VANET). The mechanism uses a hash function and bitwise XOR operations to verify the legitimacy of the vehicle and the data messages. Chen et al. [29] propose a patch to the mechanism of Ying et al. [28]. The patched mechanism provides better security than the original mechanism [28]. However, no security evaluation of the patched mechanism [29] is reported. Lansky et al. [30] propose BCmECC, an Elliptic Curve Cryptography-(ECC) based lightweight authentication mechanism, that relies on a public blockchain to validate the users' public keys. However, BCmECC has high computational complexity and resource consumption, which increase authentication time and thus make it unsuitable for low-latency IoT networks. The above-discussed mechanisms exhibit low privacy

TABLE 1. A comparison of the related work on security criterion.

Related Work	MA	SIO	SKA	AR	IMA	NR
[21]	×	×	×	×	×	×
[22]	×	×	×	✔	×	×
[23], [24]	✔	×	✔	×	×	×
[2], [4], [6], [25]–[29]	✔	×	✔	✔	×	×
[1], [3], [30]–[32]	✔	×	✔	✔	×	✔
Z-PMA (this paper)	✔	✔	✔	✔	✔	✔

TABLE 2. A comparison of the related work on privacy preservation, scalability, authentication time, and suitability criteria.

Related Work	Privacy Preservation	Scalability	Authentication Time	Suitability
[23]	-	Low	-	Low
[21]	P	Low	-	Low
[22]	P, A	High	-	High
[25]	P, A, U	Low	-	Low
[24], [26]	-	Low	Low	Low
[31]	P	Low	High	Low
[6]	A	Low	High	Low
[29]	P, A	Low	Low	Low
[1], [3], [32]	P, A	Low	High	Low
[4]	P, A, T	Low	High	Low
[2], [28]	P, A, U	Low	High	Low
[27]	P, A, T, U	Low	High	Low
[30]	P, A, T, U	Low	Low	Low
Z-PMA (this paper)	P, A, T, U	High	Low	High

preservation, low scalability, and high authentication time. Therefore, to address these problems, our proposed Z-PMA mechanism uses distributed edge computing, a permissioned blockchain network, and a contract theory-based incentive mechanism to provide high security and privacy preservation, high scalability, and low authentication time.

Blockchain-based authentication mechanisms address some of the security problems observed in IoT networks. Syed et al. [6] propose a lightweight, continuous device-to-device authentication (LCDA) mechanism that uses dynamically changing session keys for continuous device-to-device authentication. However, the mechanism does not perform cost and scalability analyses, which makes it difficult to analyze the suitability for IoT networks. Rasheed et al. [4] propose a blockchain-based ZKP authentication mechanism, which uses a blockchain network to protect authentication records from tampering. However, the blockchain structure is relatively simple, causing it to be unsecure and unscalable. Song et al. [2] propose a ZKP-based authentication mechanism for Radio Frequency Identification (RFID) and describe a polynomial time simulator that proves the security of the mechanism. However, the mechanism is centralized, which leads to a lack of scalability. Dwivedi et al. [5] propose a ZKP-based authentication mechanism that provides privacy preservation in IoT networks. They also propose a ZKNimble-based data encryption technique that can be used for encryption and decryption by users for communications after the authentication process. The idea seems promising but they do not provide a performance analysis of the mechanism. Ramezan et al. [3] propose a ZKP-based authentication and key agreement mechanism to mitigate DDoS attacks in IoT networks. However, the mechanism has a single-point-of-failure and scalability problems due to its centralized design. Wang et al. [31] propose a ZKP-based authentication mechanism for IoT-embedded devices, aiming to address the challenges associated with resource limitations in computation, communication, and storage. Their mechanism employs hash commitments during the registration phase to associate identity information with a unique identifier, which simplifies the verification process and reduces resource usage. They also introduce direct communication between the embedded device and the authentication server for key management, which incorporates Chebyshev Polynomial chaotic maps. However, while the mechanism decentralizes certain aspects of the authentication process, it still relies on central components for key management and registration, potentially introducing single points of failure and scalability problems. Sharma et al. [32] propose a secure authentication and privacy-preserving blockchain framework for the Industrial Internet of Things (IIoT). They develop a Fully Homomorphic Encryption Neural

Network (AFHENN) to ensure secure user authentication and optimal blockchain node selection. The proposed system includes cryptographic measures like Transient Key Congruential Generator-based Elliptic Curve Cryptography (TKCG-ECC) and Dual Keyed Cipolla's Extended Euclidean Algorithm-based Lattice Cryptosystem (DKCEED-LC) to safeguard registered user data. Additionally, the blockchain network employs Keyed-ZKP (k-ZKP) and a Approximation Fully Homomorphic encryption neural network for data authentication, enhancing security against common cyber threats. Although the system achieves higher throughput and Packet Delivery Ratio (PDR) with reduced computing time than other methods, it still relies on centralized components for key management and registration, introducing potential single points of failure and scalability problems. Yang et al. [1] propose a secure and lightweight ZKP-based authentication mechanism for IoT networks. The mechanism achieves mutual authentication between IoT devices and servers, and maintains IoT data privacy using the modular square root (MSR) technique. A public blockchain network is used to store the smart contract that registers the devices on the network to prevent malicious devices from transmitting data to the network. However, the mechanism has scalability problems due to the use of a public blockchain and the IoT devices in the mechanism perform computationally-intensive operations, which makes it unsuitable for large-scale, low-latency, resource-constrained IoT applications. The above-discussed mechanisms typically have low scalability and high authentication time, which make them unsuitable for low-latency IoT applications. Our Z-PMA mechanism is included in Tables 1 and 2 for comparison purposes. This new mechanism uses a ZKP-based authentication mechanism based on the Quadratic Residue (QR) technique to provide secure and privacy-preserving authentication between IoT devices and edge devices. In addition, Z-PMA uses an incentive mechanism to increase scalability and decrease the authentication time in the network, which allows it to support low-latency IoT applications. Furthermore, the authentication mechanism in Z-PMA is implemented as a smart contract, which is an immutable self-executing program in a blockchain network. To further increase the security of the Z-PMA mechanism, edge devices host the main blockchain network and AAs host the sidechain network, in both cases to store information in a distributed and unforgeable manner.

III. BACKGROUND KNOWLEDGE

This section overviews the quadratic residue (QR) technique and the blockchain technology.

A. QUADRATIC RESIDUE TECHNIQUE

To understand the QR technique, we first need to understand the modular square root (MSR) problem. The MSR problem is to determine a square root of y mod n for given integers y, n, and x, such that

$$x^2 = y(\text{mod } n) \tag{1}$$

No polynomial time algorithm is known to solve the MSR problem for an arbitrary value of n. However, if n is a prime or a product of two odd primes, such a polynomial time algorithm exists. Two examples of such values of n are 13 and 15:

$$\begin{aligned} x|x^2 &= 1(\text{mod } 15) = \{1, 4, 11, 14\} \\ x|x^2 &= 1(\text{mod } 13) = \{1, 12\} \end{aligned} \tag{2}$$

where 13 is a prime and 15 is a product of two odd primes (i.e., 3 and 5).

Definition: Let n be a prime number. The integer a is said to be a **quadratic residue (QR)** of modulo n if the congruence $a \equiv b^2 \bmod n$ has a solution. Otherwise, a is called a **quadratic non-residue (QNR)**. Here, $a, b \in \mathbb{Z}_n^*$ and $\mathbb{Z}_n^* = \{a|1 \leq a \leq n, \gcd(a, n) = 1\}$ is the set of integers between 1 and n that are relatively prime to n (i.e., they do not share any factors other than 1). If n is prime, then $\mathbb{Z}_n^*$ is the values from 1 up to $(n - 1)$.

For any prime n, the set $QR(n)$ has $(n - 1)/2$ elements. For $n = p \cdot q$, where p and q are odd prime numbers, the set $QR(n)$ has $(p - 1)(q - 1)/4$ elements. To find all $QR(n)$ of $\mathbb{Z}_n^*$, we can compute the squares of all elements in $\mathbb{Z}_n^*$. For example, $n = 13$ is an odd prime and $QR(13)$ is calculated as:

$$\begin{gathered} \text{Since} \\ n = 13\ , \mathbb{Z}_{13}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\} \\ 1 \equiv 1^2(\text{mod } 13),\ 4 \equiv 2^2(\text{mod } 13),\ 9 \equiv 3^2(\text{mod } 13) \\ 3 \equiv 4^2(\text{mod } 13),\ 12 \equiv 5^2(\text{mod } 13),\ 10 \equiv 6^2(\text{mod } 13) \\ 10 \equiv 7^2(\text{mod } 13),\ 12 \equiv 8^2(\text{mod } 13),\ 3 \equiv 9^2(\text{mod } 13) \\ 9 \equiv 10^2(\text{mod } 13),\ 4 \equiv 11^2(\text{mod } 13),\ 1 \equiv 12^2(\text{mod } 13) \\ \therefore QR(13) = \{1, 3, 4, 9, 10, 12\} \end{gathered} \tag{3}$$

According to *Euler's Criterion*, a is a QR modulo n, such that $\gcd(a, n) = 1$, iff $a^{(n-1)/2} \equiv 1(\text{mod } n)$. Similarly, if $n = p \cdot q$ and $\gcd(a, n) = 1$, where p and q are odd primes that satisfies $p = q = 3(\text{mod } 4)$, then a is a QR iff $a^{(p-1)/2} \equiv 1(\text{mod } p)$ and $a^{(q-1)/2} \equiv 1(\text{mod } q)$.

We use the Chinese Remainder Theorem (CRT) to find the solutions for a QR congruence $a \equiv b^2(\text{mod } n)$, given $n = p \cdot q$ (where $p = q = 3(\text{mod } 4)$), such that

$$S_{1,2,3,4} = (\pm a_1.q.y_1 \pm a_2.p.y_2) \bmod n \tag{4}$$

where $a_1 = a^{(p+1)/4}(\text{mod } p)$, $y_1 = q^{-1}(\text{mod } p)$, $a_2 = a^{(q+1)/4}(\text{mod } q)$, $y_2 = p^{-1}(\text{mod } q)$, and $S_{1,2,3,4}$ are the four solutions of the QR congruence. For example, the QR congruence of $x^2 \equiv 11 \bmod 133$, where $133 = 7 \cdot 19$, has four solutions, using Equation (4), i.e., $121, 12, 107, 26(\text{mod } 133)$.

A problem is called *tractable* iff an efficient (i.e., polynomial-time) algorithm is known that solves it.

Otherwise, it is called *intractable*. Given $n = p \cdot q$, where p and q are unknown, there is no known polynomial-time algorithm to find a solution to $QR(n)$ [33], which means that finding the QR set is an intractable problem. Since the QR technique, where the factors of n are unknown, is based on an intractable problem, this technique is secure. Also, if p and q are large odd prime numbers, it is difficult to factor n in a given polynomial time. In addition, p and q should be updated in each round to prevent their disclosure, which would make the problem tractable.

B. BLOCKCHAIN TECHNOLOGY

A *blockchain* is a decentralized and secure digital ledger that records transactions across a network of computers. A blockchain consists of blocks that are linked together using cryptography. Each block contains a unique digital signature and a record of multiple transactions. Once a block is added to the blockchain, its information is verified and then becomes permanent and immutable. This immutability makes blockchain technology particularly useful for maintaining secure and trustworthy records of transactions. Although blockchains were introduced as the underlying technology behind the Bitcoin cryptocurrency, their potential has since been recognized in other industries as well. For example, in an IoT network, a blockchain can be used to provide secure and trustworthy communication for data exchange between IoT devices. Blockchain technology provides the following features relevant to IoT networks:

- Decentralization: The decentralized nature of a blockchain ensures that there is no single point of failure in the network and eliminates the need for intermediaries, resulting in a secure and efficient system.
- Security: Blockchains use cryptographic techniques, such as hash functions, to secure the data in the network, making it immutable.
- Transparency: Transactions on a blockchain are transparent and can be easily audited, making it easier to detect any suspicious activity.
- Scalability: The distributed nature of a blockchain enables it to scale effectively to accommodate large numbers of connected devices.
- Trust: The use of smart contracts in blockchain technology enables the automated execution of transactions, reducing the risk of fraud and increasing trust between devices.

IV. NETWORK ARCHITECTURE AND ASSUMPTIONS

This section explains the network architecture of the IoT network, the assumptions about the network, the adversary model, and the security requirements.

A. NETWORK ARCHITECTURE

Our proposed network architecture includes three layers: Edge Layer, Base Station (BS) Layer, and IoT Device Layer, as shown in Figure 1.

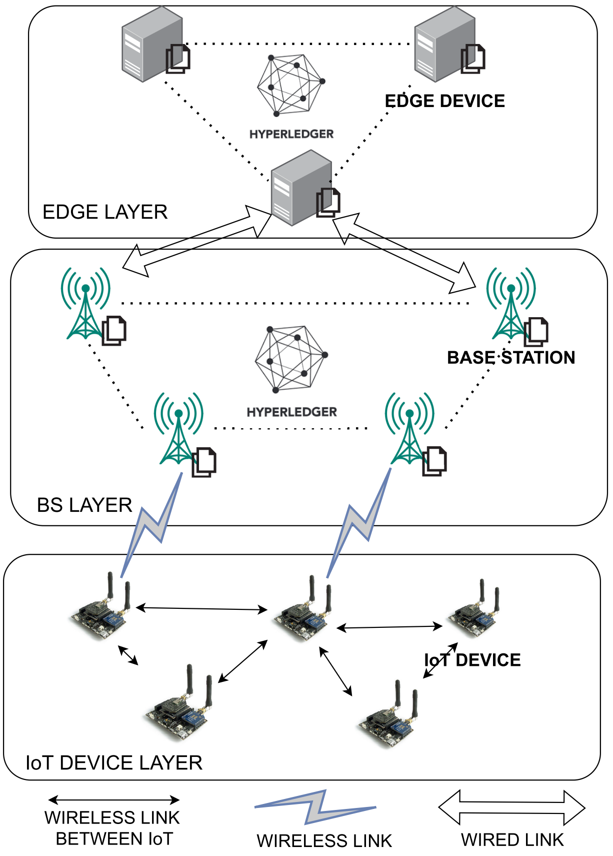


FIGURE 1. Network architecture for Z-PMA mechanism.

1) Edge Layer: consists of edge devices, which runs the main blockchain network. Edge devices are equipped with higher computational resources than IoT devices. Each edge device maintains a ledger that stores certain parameters (discussed in Section IV-A) and the transactions of the authentication phase.
2) BS Layer: consists of base stations (BSs), which run a sidechain blockchain network and can be used as Additional Authenticators (AAs). AAs are used in the authentication phase to decrease the authentication time. A *sidechain network* is a separate blockchain network running in parallel to the main blockchain network [34]. Adding a sidechain network can increase the data privacy and scalability of the blockchain network [35]. The communication between the sidechain and the main blockchain is carried out using cross-chain smart contracts. These smart contracts are installed on both chains and they transfer values between the main blockchain and the sidechain [36]. A simple example of cross-chain communication is shown in Figure 2. If there is a cross-chain communication transaction from the sidechain to the main blockchain, the sidechain generates a timestamp and sends the transaction to the main blockchain using TCP. Later, both chains update their ledgers. Each BS maintains a ledger that stores

certain parameters (discussed in Section IV-A) and the transactions between BSs and other devices. The BS layer and edge layer are connected via a wired link.

3) IoT Device Layer: consists of IoT devices that are authenticated using the Z-PMA mechanism. The IoT device layer and BS layer are connected via wireless links. Also, the IoT devices in the network are connected via wireless links.

B. NETWORK ASSUMPTIONS

In the Z-PMA mechanism, we make the following assumptions:

- Edge devices are trustworthy during the authentication process. In addition, edge devices do not disclose private information about IoT devices.
- BSs may be trustworthy, curious, or malicious. A curious BS follows the authentication mechanism but eavesdrops on the communication channel to gather information. A malicious BS intentionally disrupts the authentication phase.
- When a BS or an IoT device is compromised and is detected by the blockchain network, its credentials are revoked and removed from the blockchain network.
- In the blockchain network, most nodes are honest and follow the authentication process. Specifically, if the total number of nodes in the blockchain network is N, the risk that a malicious node can disrupt the authentication process is $1/N$.

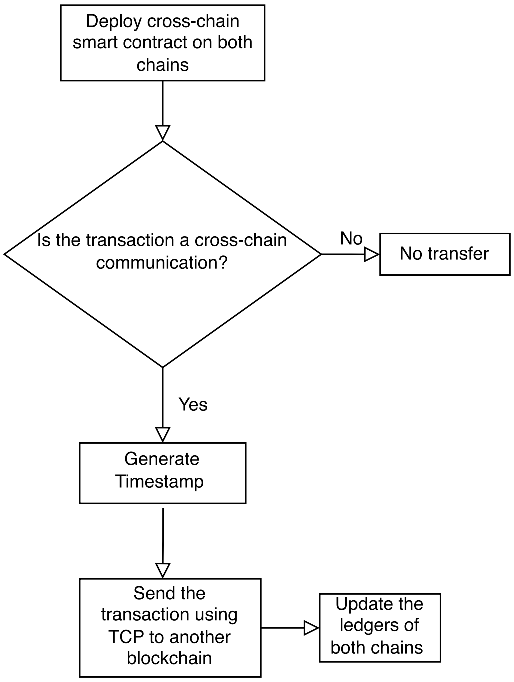

Deploy cross-chain smart contract on both chains
Is the transaction a cross-chain communication?
No
No transfer
Yes
Generate Timestamp
Send the transaction using TCP to another blockchain
Update the ledgers of both chains

FIGURE 2. Flowchart of simple cross-chain communication.

C. ADVERSARY MODEL

In this paper, we use the classical Dolev-Yao model [37] to evaluate the security of the Z-PMA mechanism. The model assumes that the cryptographic primitives used in the protocol are secure and that an adversary can intercept, tamper with, delete, store, and replay any message from the open channel. An adversary can only decrypt or sign a message if it has the correct key. Adversaries can only forge new messages from the keys and messages in their possession [38].

In the Z-PMA mechanism, an adversary $\mathcal{A}$ has the following abilities:

1) $\mathcal{A}$ can eavesdrop on the communication channel to obtain private information.
2) $\mathcal{A}$ can impersonate a legitimate device (a BS or an IoT device) to modify, delete, or replay the messages to affect the performance of the network.
3) $\mathcal{A}$ can collude with a BS to disrupt the authentication phase.
4) $\mathcal{A}$ can try to obtain the private information of the IoT devices.

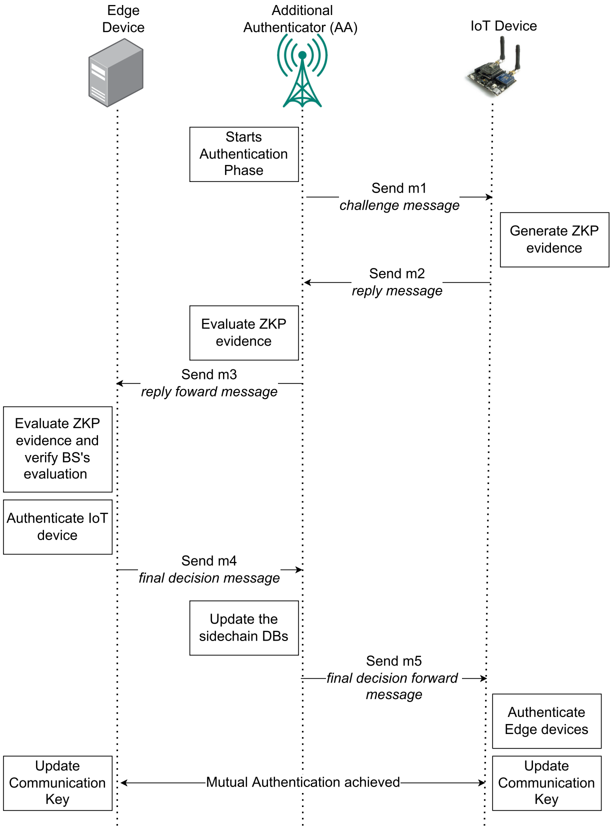


FIGURE 3. Authentication phase for Z-PMA mechanism.

D. SECURITY REQUIREMENTS

We list the security requirements that the Z-PMA mechanism must achieve to provide secure, scalable and privacy-preserving authentication [1], [2], [3], [4], [5], [6], [7]:

1) Mutual Authentication: Any devices in the IoT network must be registered and capable of identifying and authenticating other devices.

2) Privacy Preservation: Any private information about an IoT device's *ID* and communication key, and a base station's idle time must either not be disclosed to the devices in the network or disclosed in such a manner that an attacker cannot obtain this information.
3) Secure Incentive Operation: The security of the base station incentive operation must be ensured to keep the network secure and privacy-preserving.
4) Secure Key Agreement: After successful mutual authentication between the IoT devices and edge devices, a session key must be established to facilitate future secure communication and maintain the confidentiality and integrity of the transmitted data.
5) Attack Resistance: Any malicious or unauthorized IoT device that tries to access the network must be identified and excluded from the network, and any security attacks, such as eavesdropping, replay, and Man-in-the-Middle (MITM) attacks, must be mitigated.
6) Immunity to Malicious AAs: Any malicious AA must be identified and removed during the authentication phase.
7) Non-repudiation: Any device in the IoT network that transmits a message must not later deny that it transmitted the message.

We use these requirements to perform our security analysis in Sections VI-A and VI-B.

V. THE OPERATION OF THE Z-PMA MECHANISM

The Z-PMA mechanism performs two operations: (1) the privacy-preserving mutual authentication (PMA) operation, which mutually authenticates edge devices and IoT devices using QR-based ZKP techniques in such a way that IoT device are authenticated without revealing valuable information (*ID* and *sk*) to edge devices and AAs [39], and (2) the base station incentive (BSI) operation, which encourages BSs to provide their extra computing resources to achieve fast and secure authentication.

A. PRIVACY-PRESERVING MUTUAL AUTHENTICATION (PMA) OPERATION

This subsection describes the PMA operation, which achieves mutual authentication between edge and IoT devices. The PMA operation consists of two phases.

1) NETWORK INITIALIZATION PHASE

The main blockchain assigns each joining IoT device a device ID and a communication key, expressed as $\langle ID, sk \rangle$. The main blockchain also generates two pairs of prime numbers (p_1, q_1) and (p_2, q_2), and calculates $N_1 = p_1 \times q_1$ and $N_2 = p_2 \times q_2$. Here, p_1, q_1, p_2, and q_2 are considered to be private parameters, whereas N_1 and N_2 are considered to be public parameters. The main blockchain, hosted on the edge devices, stores the $ID, sk, N_1, p_1, q_1, N_2, p_2$, and q_2 parameters in its ledger to provide distributed and unforgeable storage of these parameters. In addition, a sidechain hosted on the AAs, stores the N_1, N_2, p_2, and q_2 parameters in its ledger. When an IoT device wants to leave the network, it sends a deregistration request (which is a nonce-size message) to the main blockchain. Later, the main blockchain invalidates the corresponding IoT device's ID and sk in its ledger and terminates the IoT device's authentication status. The edge devices and AAs use these parameters to authenticate the IoT devices. Later, the edge devices select multiple BSs based on the base station incentive mechanism (explained in Section IV-B) to act as AAs to decrease authentication time. The AAs use their redundant computing power to authenticate IoT and edge devices and receive incentives for successful authentications.

2) AUTHENTICATION PHASE

To understand the authentication phase, let the total number of AAs be α, such that $i \in \{1, 2, 3, \ldots, \alpha\}$ and α is always less than or equal to the total number of BSs. Before the authentication phase begins, the AAs check the authentication status of an IoT device d. If the authentication status has expired, the IoT device d must be re-authenticated. The authentication phase consists of six steps and is shown in Algorithm 1 and Figure 3.

In Step 1 of Algorithm 1, each AA performs the following tasks. First, each AA generates a random number r_i selected from $\mathbb{Z}^*_{N_2}$. Here, $\mathbb{Z}^*_{N_2}$ represents a set of integers in a multiplicative group of N_2. Secondly, each AA produces a vector $\overrightarrow{v_i}$ with j elements, which are randomly selected from $\{0, 1\}$ using a uniform distribution, such that $\overrightarrow{v_i} = [v_{i1}, v_{i2}, \ldots, v_{ij}]$. Then, each AA sends a ZKP-based challenge to the IoT device as the *challenge message* $m_1 := \langle r_i, \overrightarrow{v_i} \rangle$. The role of $\overrightarrow{v_i}$ is to reduce the probability of unauthorized IoT devices bypassing the authentication phase to $1/2^j$.

In Step 2, the IoT device d performs the following tasks. First, it aggregates all the received messages m_1 into $\langle R, \overrightarrow{V} \rangle$, where $R = \{r_1, r_2, \ldots, r_\alpha\}$ and $V = \{\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots, \overrightarrow{v_\alpha}\}$. Second, the IoT device generates a random number r_d selected from $\mathbb{Z}^*_{N_2}$ and calculates $y = sk \oplus r_d$, where $\oplus$ is the xor operation. This process is repeated in a loop until $y \in \mathbb{Z}^*_{N_1}$ is satisfied. Then, r_d and y are encrypted using the QR technique as $R_d \leftarrow QR_E(r_d) = r_d^2 \bmod N_2$ and $Y \leftarrow QR_E(y) = y^2 \bmod N_1$, respectively. Note that R_d and Y cannot be decrypted without knowing p_1, q_1, p_2 and q_2. Third, the IoT device calculates $k_d = H(ID||sk)$, where $H()$ is a hash function and $||$ is a concatenation function. The value of k_d is encrypted as $K = QR_E(k_d) \oplus r_d$, where $QR_E(k_d) = k_d^2 \bmod N_1$. Fourth, in response to the vector $\overrightarrow{v_i}$ sent by AAs, the IoT device generates a corresponding vector $\overrightarrow{u_i} = [u_{i1}, u_{i2}, \ldots, u_{ij}]$, such that each element $u_{ij} = r_d k_d^{v_{ij}} \bmod N_1$, where $i \in \{1, 2, 3, \ldots, \alpha\}$. All $\overrightarrow{u_i}$ values are aggregated into $U = \{\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_\alpha}\}$. Additionally, hash values of y and r_d are calculated as $H(y)$ and $H(r_d)$, respectively. Finally, the IoT device sends a ZKP-based reply to AAs as *reply message* $m_2 := \langle R_d, Y, K, \overrightarrow{U}, H(y), H(r_d) \rangle$.

In Step 3, each AA verifies the ZKP-based reply in the following ways. First, since AAs know private parameters like p_2 and q_2, and also know R_d and N_2, each AA decrypts R_d using CRT and obtain four solutions of $r_d \leftarrow QR_D(R_d)$, such as $(r_{d_1}, r_{d_2}, r_{d_3}, r_{d_4})$. Then, the original value of r_d is identified by comparing $H(r_d)$ and $H(r_{d_i})$ for $1 \leq i \leq 4$. Second, each AA calculates the authentication result using $x_{ij} = (u_{ij})^2 \bmod N_1 - r_d^2(K \oplus r_d)^{v_{ij}} \bmod N_1$, where $i \in \{1, 2, 3, \ldots, \alpha\}$. Finally, each AA sends *reply forward message* $m_3 := \langle X_i, m_1, m_2 \rangle$ to the edge devices, where $X_i = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ij}\}$.

In Step 4, each edge device aggregates all the received X_i into a set X, such that $X = \{X_1, X_2, X_3, \ldots, X_\alpha\}$. Then, each edge device performs two tests. In the first test, if the IoT device is legitimate, all the elements in each X_i for $1 \leq i \leq \alpha$ of the set X must be equal to 0. Otherwise, the IoT device is illegitimate. Each edge device decrypts R_d using CRT and obtains four solutions for $r_d \leftarrow QR_D(R_d)$, such as $(r_{d_1}, r_{d_2}, r_{d_3}, r_{d_4})$. Then, the original value of r_d is identified by comparing $H(r_d)$ with $H(r_{d_i})$ for $1 \leq i \leq 4$. In the second test, each edge device decrypts Y to obtain y using CRT (same as step 3) to mitigate the possibility of malicious AAs in the network. Using the decrypted y, each edge device calculates sk' such that $sk' = y \oplus r_d$ and matches sk' with the sk stored in the main blockchain with the ID. If it matches, the IoT device is legitimate; otherwise, it is illegitimate. If both tests yield the same result, no AAs in the network are malicious; otherwise, malicious AAs are detected and identified using set X. Next, if the IoT device is legitimate, each edge device calculates $E_p = ID \oplus r_d \oplus y$ as evidence to prove the edge device's identity to the IoT device. Also, each edge device updates sk to $sk_n = H(sk||r_d)$ for future communication with IoT devices. Finally, each edge device sends *final decision message* $m_4 := \langle Res, H(E_p) \rangle$ to AAs, where Res represents the final decision of the edge devices, i.e., whether the IoT device is authenticated (legitimate) or not.

In Step 5, each AA records the Res value into the sidechain and forwards $H(E_p)$ to the IoT device as *final decision forward message* $m_5 := \langle H(E_p) \rangle$.

In Step 6, the IoT device calculates $H(sk \oplus r_d \oplus y)$ and compares it with $H(E_p)$. If both are equal, the edge devices are legitimate, and mutual authentication has been achieved between IoT devices and edge devices. Finally, the IoT devices update sk to $sk_n = H(sk||r_d)$ for future communication.

B. BASE STATION INCENTIVE (BSI) OPERATION

The BSI operation is designed to incentivize BSs acting as AAs to provide their spare computing resources for use in the authentication phase. Incentivization is reasonable given the energy cost incurred by the base stations to execute authentication tasks offloaded by the edge devices. One problem to be faced when designing a BSI operation is lack of easy access to information about a base station's availability for offloading. The resources for the authentication task may not be available at a BS if it has allocated its resources to IoT devices. The authentication process may be interrupted if

Algorithm 1 Authentication Phase

```
Input: $r_i$, $\overrightarrow{v_i}$, $r_d$
Output: $sk$ or REJECT
Set $status_{IoT}$ = illegitimate
Set $status_{Edge}$ = illegal
Set $Test_1$ = 0
Set $Test_2$ = 0
/* Step 1: Each AA does the following */
for i from 1 to $\alpha$ do
    Generate $r_i \in \mathbb{Z}^*_{N_2}$
    Generate $\overrightarrow{v_i} = [v_{i1}, v_{i2}, \ldots, v_{ij}]$ with $j$ elements
    Send $m_1 := \langle r_i, \overrightarrow{v_i} \rangle$ to IoT device
end for
/* Step 2: IoT device does the following */
Aggregate all received $m_1$ into $\langle R, \overrightarrow{V} \rangle$
do
    Generate $r_d \in \mathbb{Z}^*_{N_2}$
    Compute $y = sk \oplus r_d$
while $y \notin \mathbb{Z}^*_{N_1}$
Compute $R_d \leftarrow QR_E(r_d) = r_d^2 \bmod N_2$
Compute $Y \leftarrow QR_E(y) = y^2 \bmod N_1$
Compute $k_d = H(ID||sk)$
Compute $QR_E(k_d) = k_d^2 \bmod N_1$
Compute $K = QR_E(k_d) \oplus r_d$
for i from 1 to $\alpha$ do
    Compute $\overrightarrow{u_i} = [u_{i1}, u_{i2}, \ldots, u_{ij}]$, where $u_{ij} = r_d k_d^{v_{ij}} \bmod N_1$
end for
Aggregate all $u_{ij}$ into $U = \{\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_\alpha}\}$
Compute $H(y)$ and $H(r_d)$
Send $m_2 := \langle R_d, Y, K, \overrightarrow{U}, H(y), H(r_d) \rangle$ to AAs
/* Step 3: Each AA does the following */
Compute $(r_{d_1}, r_{d_2}, r_{d_3}, r_{d_4}) \leftarrow QR_D(R_d)$ using CRT
Compute $r_d \leftarrow r_{d_i}$ s.t. $H(r_d) = H(r_{d_i})$ for $1 \leq i \leq 4$
for i from 1 to $\alpha$ do
    Compute $x_{ij} = (u_{ij})^2 \bmod N_1 - r_d^2(K \oplus r_d)^{v_{ij}} \bmod N_1$
    Aggregate $X_i = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ij}\}$
    Send $m_3 := \langle X_i, m_1, m_2 \rangle$ to edge devices
end for
/* Step 4: Each edge device does the following */
Aggregate all received $X_i$ into $X = \{X_1, X_2, \ldots, X_\alpha\}$
if $X == 0$ then
    Set $Test_1$ = 1
end if
Compute $(r_{d_1}, r_{d_2}, r_{d_3}, r_{d_4}) \leftarrow QR_D(R_d)$ using CRT
Compute $r_d \leftarrow r_{d_i}$ s.t. $H(r_d) = H(r_{d_i})$ for $1 \leq i \leq 4$
Compute $(y_1, y_2, y_3, y_4) \leftarrow QR_D(Y)$ using CRT
Compute $y \leftarrow y_i$ s.t. $H(y) = H(y_i)$ for $1 \leq i \leq 4$
Compute $sk' = y \oplus r_d$
if $sk' == sk$ then
    Set $Test_2$ = 1
end if
if $Test_1 == Test_2$ then
    if $Test_1 == 1$ then
```

```
        Set $status_{IoT}$ = legal
    else
        return REJECT
    end if
else
    Find Malicious AAs using $X_i$
    return REJECT
end if
    Compute $E_p = ID \oplus r_d \oplus y$
    Compute $sk \leftarrow sk_n = H(sk||r_d)$
    Send $m_4 := \langle Res, H(E_p)\rangle$ to AAs
/* Step 5: Each AA does the following */
    Record $Res$ into sidechain
    Send $m_5 := \langle H(E_p)\rangle$ to IoT device
/* Step 6: IoT Device does the following */
    Compute $H(ID \oplus r_d \oplus y)$
if $H(ID \oplus r_d \oplus y) == H(E_p)$ then
    Set $status_{Edge}$ = legal
    Compute $sk \leftarrow sk_n = H(sk||r_d)$
else
    return REJECT
end if
```

the authentication task is deployed on a BS without knowing whether its resources are currently available. Resource availability in BSs is private information and is not disclosed to edge devices, which leads to information asymmetry between edge devices and BSs. To avoid delays during authentication, the BSI operation should be designed to overcome this information asymmetry.

We use contract theory [40], a powerful tool from economics, to handle the problem of information asymmetry in the BSI operation. Using contract theory, the group of edge devices is modeled as an employer who offers a work contract to each BS. Each work contract consists of a reward-resource pair, i.e., $(\$, CC)$, where \$ represents the reward and CC represents the required computing capacity. Also, \$ is an increasing function of CC to ensure that rewards increase as required computing capacity increases. The reward can be in the form of monetary value or any privileged access, such as an amount of free computing power to offloading data, high traffic processing priority [41], [42]. Due to information asymmetry, edge devices are unaware of the idle times of the BSs. Therefore, the edge devices partition all the BSs into N discrete *types* using statistical distributions of the BSs' behaviors from historical data to improve the efficiency of offering work contracts. For clarity, we refer to the type i as t_i. Let $T = \{t_1, t_2, \ldots, t_N\}$ represent the set of N discrete *types*, such that

$$t_1 < \ldots < t_i < \ldots < t_N, i \in \{1, \ldots, N\} \text{ and } N \leq \alpha \tag{5}$$

Here, BSs with higher types are more likely to be selected as AAs and each type t_i is associated with a work contract denoted $\mathcal{WC}_i = (\$_i, CC_i)$, $\forall t_i \in T$. The type of BS b, represented as type (x), is calculated based on two factors, which are the credibility c_b of the computing capacity of the BS b evaluated by the edge devices based on historical data about interactions, and the probability p_b that the BS b is idle for at least τ time. In other words, type$(b) = c_b \cdot p_b$, where $\cdot$ denotes multiplication.

The expected utility function of a BS of type t_i under the work contract offered by edge devices is the expected reward minus the energy cost of task execution.

$$\begin{aligned} E_{BS}(i) &= R_i - EC_i \\ &= (t_i\$_i) - (ew\mu(CC_i)^2) \end{aligned} \tag{6}$$

where R_i is the expected reward for offloaded computation based on $\mathcal{WC}_i$, $\$_i$ is the reward earned by a BS of type t_i from edge devices for the offloaded computation, EC_i is the energy cost incurred by task execution based on $\mathcal{WC}_i$, e is the energy cost per unit workload, w is the the given workload, μ is the energy co-efficient, and CC_i is the computing capacity of a BS of type t_i.

To decrease the authentication time, the BS must meet the latency requirements of the edge devices. Therefore, the expected utility function of the edge devices for a BS of type t_i is given by:

$$E_{ED}(i) = (\tau_i - \$_i) \tag{7}$$

where τ_i is the amount of time saved by offloading computation to a BS of type t_i. The factor τ_i is calculated as:

$$\tau_i = \eta\left(\frac{w}{CC_{edge}} - \frac{w}{CC_i} - \frac{w}{TR_i}\right) \tag{8}$$

where η is the profit co-efficient for the unit time saved, CC_{edge} is the computing capacity of the edge device, and TR_i is the transmission rate between a BS of type t_i and the edge devices. A BS with a higher value for CC_i can provide greater benefit to the edge devices.

Here, the BSI operation aims to maximize the expected utility function of the edge devices (E_{ED}), the rewards earned by the BS, and the expected utility function of the base station (E_{BS}). Such a scenario leads to an optimization problem, which is formulated as follows:

$$\begin{aligned} \max_{\langle \$_i, CC_i\rangle} \; & \sum_{i=1}^{N} \psi_i\left[\eta\left(\frac{w}{CC_{edge}} - \frac{w}{CC_i} - \frac{w}{TR_i}\right) - \$_i\right] \\ s.t. \quad & \text{(9a) } (t_i\$_i) - (ew\mu(CC_i)^2) \geq 0, \;\; \forall i \in \{1, .., N\} \\ & \text{(9b) } (t_i\$_i) - (ew\mu(CC_i)^2) \geq (t_i\$_j) - (ew\mu \\ & (CC_j)^2), \;\; \forall i \neq j, \;\; i, j \in \{1, .., N\} \\ & \text{(9c) } 0 \leq \$_1 \leq \ldots \leq \$_i \leq \ldots \leq \$_N \end{aligned} \tag{9}$$

where ψ_i is the probability that edge devices select a base station of type t_i and $\sum_{i=1}^{N} \psi_i = 1$. The individual rationality (IR) constraint in (9a) guarantees that the utility function of each BS is non-negative. Otherwise, it is ineffectual for the BS to participate in the BSI operation. The incentive compatibility (IC) constraint in (9b) guarantees that a BS of type t_i can receive a maximum reward by selecting the work contract $\mathcal{WC}_i = (\$_i, CC_i)$. In other words, IC shows that BSs receive work contracts according to their types. The constraint in (9c) shows the monotonicity of the contract, which means that BS of higher types receive higher rewards for performing computations.

VI. RESULTS AND ANALYSIS

In the previous sections, we highlighted three main requirements of an authentication mechanism, such as security and privacy preservation, scalability, and authentication time. We provide a detailed experimental evaluation of the performance of the Z-PMA authentication mechanism to show that the above requirements are satisfied. We divide our evaluation into four parts, namely, informal analysis of security requirements, formal analysis of security requirement using the Proverif tool, simulation-based analysis, and IoT testbed-based analysis. In the first part, we provide a detailed security analysis of the Z-PMA mechanism showing that all the security requirements listed in Section IV-D are satisfied. In the second part, we formally evaluate the security requirements of the Z-PMA mechanism using the Proverif tool. In the simulation-based analysis, we simulate Z-PMA to evaluate its performance in terms of scalability and authentication time. We also simulate various existing mechanisms, namely Yang et al. [1], Wang et al. [31], and Sharma et al. [32] and compare their performance with the Z-PMA mechanism. We selected these mechanisms because our survey of previous work shows that they are the most similar mechanisms to the Z-PMA mechanism. Table 3 shows the parameter values used to implement the mechanisms. In the IoT testbed-based analysis, we use a Raspberry Pi as an IoT device, a PC as an edge device, and another PC as a BS device to analyze the efficiency of the Z-PMA mechanism for overall authentication time and time taken to perform specific operations in the authentication phase.

TABLE 3. Defaults parameter values of constants.

Parameters	Values
p_1, q_1, p_2, q_2	512 bits
N_1, N_2	1024 bits
ID	32 bits
sk	160 bits
r_d, r_i	1024 bits
timestamp	32 bits
hash function	160 bits

A. INFORMAL ANALYSIS OF SECURITY REQUIREMENTS

In this section, we show how the security requirements listed in Section IV-D are satisfied by the Z-PMA mechanism.

1) MUTUAL AUTHENTICATION

In the Z-PMA mechanism, mutual authentication takes place between IoT devices and edge devices. Also, Z-PMA takes advantage of base stations as AAs using the BSI operation to decrease the authentication time. In the PMA operation, mutual authentication takes place using QR. Therefore, it is infeasible for an attacker to decrypt the encrypted messages without knowing private parameters, such as p_1, q_1, p_2, and q_2. Furthermore, due to the intractability of QR (explained in Section III), the attacker cannot factor public parameters, such as N_1 and N_2. All of these parameters are stored on the blockchain, which makes them immutable and decentralized (i.e., the same parameters are stored at different peer nodes). Overall, the Z-PMA mechanism achieves mutual authentication between IoT and edge devices in a secure manner.

2) PRIVACY PRESERVATION

In the Z-PMA mechanism, the private information of IoT devices, such as ID and sk, are hashed as $k_d = H(ID||sk)$ and encrypted as $K = QR_E(k_d) \oplus r_d$, where $QR_E(k_d) = k_d^2 \bmod N_1$. Also, sk is transmitted after hashing and encrypting it as $y = sk \oplus r_d$ and $Y \leftarrow QR_E(y) = y^2 \bmod N_1$. Therefore, an attacker who obtains K and Y during communication cannot decrypt it because the private parameters (i.e., p_1 and q_1) to decrypt the encrypted data are unavailable to the attacker. Also they are not stored on the sidechain to mitigate privacy leakage in case of malicious AAs. Similarly, under an asymmetric information scenario, private information, such as the idle time of base stations, is not disclosed on the communication channel, and edge devices use the BSI operation to select base stations as AAs.

3) SECURE INCENTIVE OPERATION

The BSI operation is secure because it is written as a smart contract and cannot be modified. Hence, the execution of the BSI operation is automatic and cannot be disturbed. Also, the output of the BSI operation is written in the blockchain directly. Since it cannot be revised by any devices, the impartiality of the AA selection in the authentication phase is ensured. Therefore, the BSI operation is secure.

4) SECURE KEY AGREEMENT

In the Z-PMA mechanism, the communication key sk is updated between IoT devices and edge devices after each successful mutual authentication. Due to the intractability of QR, the attacker cannot decrypt K without the knowledge of p_1 and q_1 and ultimately cannot obtain sk, which is required for communication. Overall, the key agreement is achieved between IoT and edge devices after the successful mutual authentication. The attacker cannot obtain the key, given that decrypting K is infeasible if p_1 and q_1 are unknown. This proves that the attacker cannot obtain the key sk without the knowledge of p_1 and q_1, which proves the anonymity properties of the Z-PMA mechanism.

5) ATTACK RESISTANCE

Z-PMA identifies malicious IoT devices and is guaranteed to resist the following security attacks in the described manners:

a) Eavesdropping attack: In an eavesdropping attack, the attacker obtains information by capturing the messages transmitted over the communication channel. However, the communication in the authentication phase is encrypted using QR, which reduces the chance of the attacker decrypting the message and obtaining private information, given the intractability of QR. Therefore, the attacker cannot obtain private information by capturing the messages communicated between the devices.

b) Replay attack: In a replay attack, the attacker captures a message (encrypted or not) by eavesdropping and resends the entire message to misdirect the receiver into doing what the attacker wants. There are three ways to prevent replay attacks, which are to use random session keys, timestamps, and one-time passwords for each transaction [43]. Therefore, to mitigate reply attacks, Z-PMA uses a random session key as the communication key sk (it changes after every authentication phase) and timestamps every message the devices send. The freshness of the timestamp is checked by all devices receiving the message, and if it is not satisfactory, the message is discarded. Overall, the attacker cannot pass the authentication phase using the reply attack.

c) MITM attacks: An MITM attack is a type of impersonation attack in which attackers intercept the communication between two devices and relay a forged/altered message in an attempt to steal important information. In Z-PMA, all devices communicate using encrypted messages and timestamps, and each IoT device is assigned one ID, which is stored in the main blockchain. Therefore, the attacker cannot impersonate any legitimate IoT device and launch a MITM attack (because the attacker cannot decrypt the message given the intractability of QR).

6) IMMUNITY TO MALICIOUS AAs

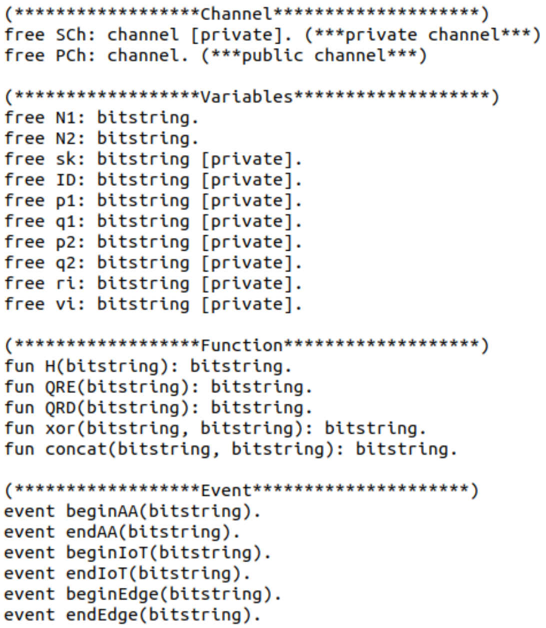
Using BSI operation, edge devices select AAs after evaluating the base station's credibility based on historical data and the probability that the base station is idle for at least a time period of τ. However, a base station can turn malicious after it is selected as an AA for the authentication phase. Z-PMA mitigates such a scenario in two ways. First, Z-PMA performs two tests (shown in lines 37-56 of Algorithm 1) to detect malicious AAs in the authentication phase. This ensures that malicious AAs are detected and removed from the network, which proves the traceability properties of the Z-PMA mechanism. Secondly, to mitigate privacy leakage due to malicious AAs, certain private parameters such as p_1, q_1, ID and sk are not stored in the AA's sidechain network. To explain, AAs receive Y along with $H(y)$ in m_2 from an IoT device, but AAs cannot decrypt Y to learn sk because p_1 and q_1 are not known by AAs. In addition, edge devices send $H(E_p)$ where $E_p = ID \oplus r_d \oplus y$ to AAs. However, AAs cannot know ID without knowing y. This shows that AAs (whether legitimate or malicious) cannot decrypt the Y parameter to learn sk and ID from the messages, which proves the anonymity and unlinkability properties of the Z-PMA mechanism. Overall, Z-PMA can detect malicious AAs in the authentication phase and can also mitigate privacy leakage due to malicious AAs.

7) NON-REPUDIATION

The property of non-repudiation is that any device cannot deny the message on the network after sending it. Here, the IoT device sends m_2 to AAs, which cannot be altered without knowing private parameters. Also, Z-PMA stores ID and sk in the main blockchain, where data cannot be altered. Therefore, Z-PMA guarantees non-repudiation, such that m_2 is sent by a particular IoT device, and ID and sk stored in the main blockchain belong to a particular IoT device.

B. FORMAL SECURITY ANALYSIS USING PROVERIF

In this section, we perform formal security analysis using the Proverif tool to evaluate the security of the Z-PMA mechanism.

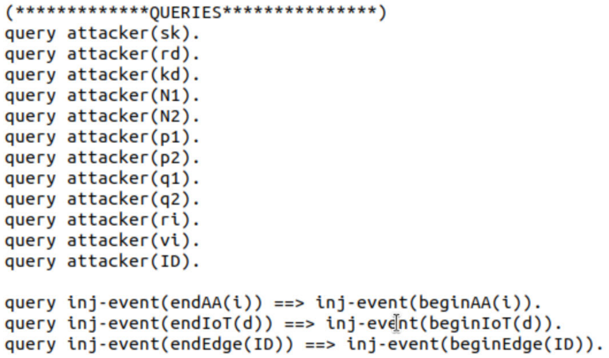


```
(******************Channel********************)
free SCh: channel [private]. (***private channel***)
free PCh: channel. (***public channel***)

(******************Variables*******************)
free N1: bitstring.
free N2: bitstring.
free sk: bitstring [private].
free ID: bitstring [private].
free p1: bitstring [private].
free q1: bitstring [private].
free p2: bitstring [private].
free q2: bitstring [private].
free ri: bitstring [private].
free vi: bitstring [private].

(******************Function*******************)
fun H(bitstring): bitstring.
fun QRE(bitstring): bitstring.
fun QRD(bitstring): bitstring.
fun xor(bitstring, bitstring): bitstring.
fun concat(bitstring, bitstring): bitstring.

(******************Event*********************)
event beginAA(bitstring).
event endAA(bitstring).
event beginIoT(bitstring).
event endIoT(bitstring).
event beginEdge(bitstring).
event endEdge(bitstring).
```

FIGURE 4. Declaration statement in Proverif code.

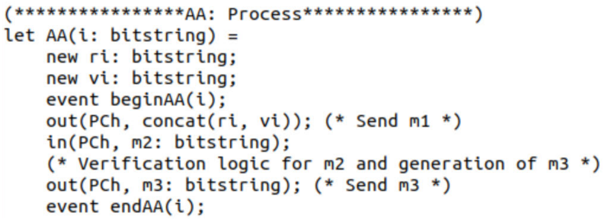
Proverif [44] is a π-algorithm based automated cryptographic protocol verification tool developed by Bruno Blanchet using the Prolog language. ProVerif has been widely used for the formal verification of cryptographic protocols [45], [46], [47], [48], [49], [50].

Our ProVerif validation code is divided into three sections, which are declaration, queries, and process for each participant. The declaration section defines the name and type for each variable. We use two channel types, Private Channel SCh for communicating participants to pass sensitive messages during the registration phase, and Public Channel PCh for communicating participants to pass messages



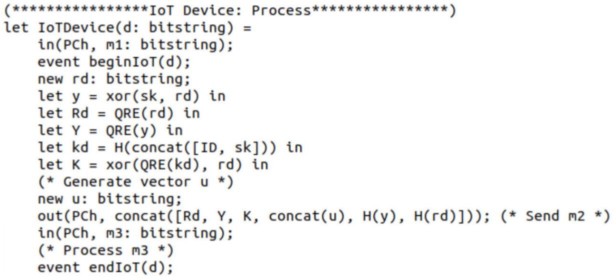
```
(************QUERIES**************)
query attacker(sk).
query attacker(rd).
query attacker(kd).
query attacker(N1).
query attacker(N2).
query attacker(p1).
query attacker(p2).
query attacker(q1).
query attacker(q2).
query attacker(ri).
query attacker(vi).
query attacker(ID).

query inj-event(endAA(i)) ==> inj-event(beginAA(i)).
query inj-event(endIoT(d)) ==> inj-event(beginIoT(d)).
query inj-event(endEdge(ID)) ==> inj-event(beginEdge(ID)).
```

FIGURE 5. Queries in Proverif code.



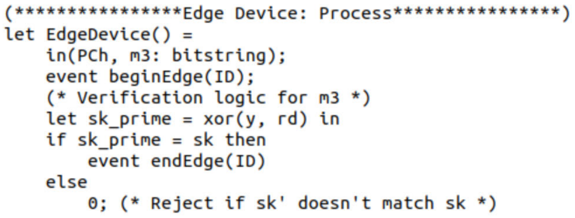
```
(***************AA: Process***************)
let AA(i: bitstring) =
    new ri: bitstring;
    new vi: bitstring;
    event beginAA(i);
    out(PCh, concat(ri, vi)); (* Send m1 *)
    in(PCh, m2: bitstring);
    (* Verification logic for m2 and generation of m3 *)
    out(PCh, m3: bitstring); (* Send m3 *)
    event endAA(i);
```

FIGURE 6. Process of AA in Proverif code.



```
(***************IoT Device: Process***************)
let IoTDevice(d: bitstring) =
    in(PCh, m1: bitstring);
    event beginIoT(d);
    new rd: bitstring;
    let y = xor(sk, rd) in
    let Rd = QRE(rd) in
    let Y = QRE(y) in
    let kd = H(concat([ID, sk])) in
    let K = xor(QRE(kd), rd) in
    (* Generate vector u *)
    new u: bitstring;
    out(PCh, concat([Rd, Y, K, concat(u), H(y), H(rd)])); (* Send m2 *)
    in(PCh, m3: bitstring);
    (* Process m3 *)
    event endIoT(d);
```

FIGURE 7. Process of IoT device in Proverif code.



```
(***************Edge Device: Process***************)
let EdgeDevice() =
    in(PCh, m3: bitstring);
    event beginEdge(ID);
    (* Verification logic for m3 *)
    let sk_prime = xor(y, rd) in
    if sk_prime = sk then
        event endEdge(ID)
    else
        0; (* Reject if sk' doesn't match sk *)
```

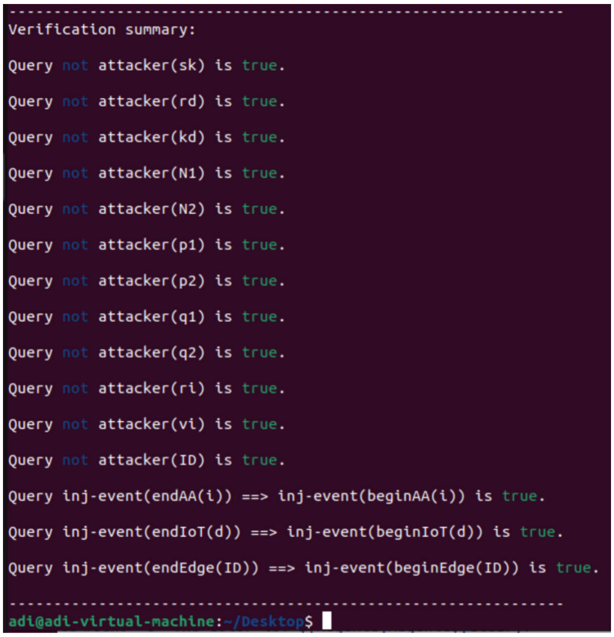
FIGURE 8. Process of edge device in Proverif code.

publicly. The variables, functions, and events required for the authentication process are also defined. The Proverif code of the declaration section is shown in Figure 4.

As shown in Figure 5, we define a set of query statements to verify the key security and mutual authenticity of each participant, and the security of the Z-PMA mechanism. We define 15 queries–three for each participant and 12 for the attacker–to indicate that the parameters are secure and the attacker cannot intercept the communication process. The attacker in Proverif is defined based on our Adversary model (Section IV-C). In 12 queries, we assess whether an attacker can deduce critical information such as sk, r_d, k_d, N_1, N_2, p_1, p_2, q_1, q_2, r_i, v_i, and ID because ensuring the secrecy of these parameters is crucial for the security of the Z-PMA mechanism. In the remaining three queries, the *inj-event* tests the event injection for each participant process to check whether the connection is successfully opened and closed.

In ProVerif, processes are used to model the behavior of participants in a security protocol. In the Z-PMA mechanism, we define three participants, which are AA, IoT device, and Edge device. We explain the processes of AA, IoT device, and Edge device in Figure 6, 7, 8, respectively.

The verification summary for our Proverif validation is shown in Figure 9. In the summary, the first 12 lines indicate that all 12 parameter queries are secure, i.e., the result is "Query not attacker() is true" for all 12 queries. Therefore, the parameters are deemed secure, and an attacker cannot intercept any of them from the public channel. For the remaining three queries, the result indicates that the event injection from *end* to *begin* is true, meaning that the participant's communication channel is functioning correctly.



```
--------------------------------------------------------------
Verification summary:

Query not attacker(sk) is true.

Query not attacker(rd) is true.

Query not attacker(kd) is true.

Query not attacker(N1) is true.

Query not attacker(N2) is true.

Query not attacker(p1) is true.

Query not attacker(p2) is true.

Query not attacker(q1) is true.

Query not attacker(q2) is true.

Query not attacker(ri) is true.

Query not attacker(vi) is true.

Query not attacker(ID) is true.

Query inj-event(endAA(i)) ==> inj-event(beginAA(i)) is true.

Query inj-event(endIoT(d)) ==> inj-event(beginIoT(d)) is true.

Query inj-event(endEdge(ID)) ==> inj-event(beginEdge(ID)) is true.

--------------------------------------------------------------
adi@adi-virtual-machine:~/Desktop$
```

FIGURE 9. Verification summary for our Proverif validation.

To understand the verification result according to the security requirement (listed in Section IV-D and analyzed in Section VI-A), we observe that all the security requirements rely on the security of parameters, namely, sk, N_1, N_2, p_1, p_2, q_1, q_2, and ID, except the secure incentive mechanism requirement, which relies on the security of the permissioned blockchain and smart contract. Also, Figure 9 shows that all the parameters are secured from the attacker. Since the security of the permissioned blockchain and smart contract is ensured by cryptographic techniques, such as hash functions, the security requirements for the Z-PMA mechanism are satisfied using formal security analysis.

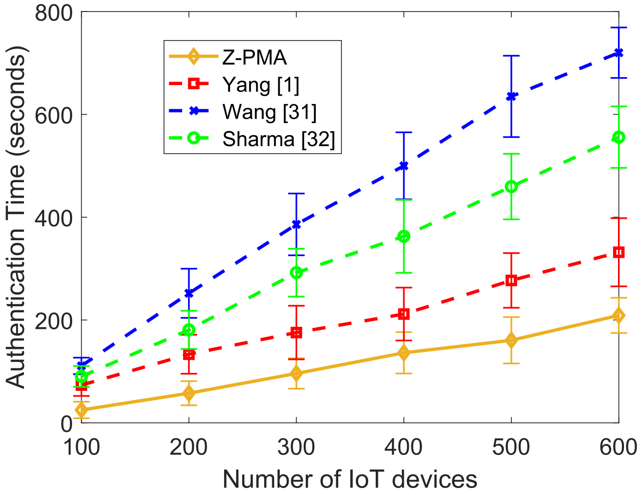


FIGURE 10. Elapsed time to authenticate IoT devices in the network.

C. SIMULATION RESULTS AND ANALYSIS

We use docker containers [51] and Hyperledger Composer [52] to simulate IoT devices, BS, and edge devices in the Z-PMA mechanism. We use an Ubuntu-based operating system on a desktop configured with a 3.5 GHz i9-10900KF, 10 cores & 64 GB RAM. The consensus algorithm used is Kafka (which performs better than Raft and Practical Byzantine Fault Tolerance (PBFT) in terms of throughput and latency as the size of the network is increased [53]), the number of peer nodes is 25 and the number of orderer nodes is 8 [20]. The authentication algorithm (Algorithm 1) and BSI operation are implemented as smart contracts and written in Golang. In this analysis, we vary the number of IoT devices. By doing so, we can compare our mechanism to existing authentication mechanisms (Yang et al. [1], Wang et al. [31], and Sharma et al. [32]) with respect to scalability.

TABLE 4. Device configuration for IoT testbed.

Device	Specification	Operating System
IoT device (Raspberry Pi 2 Model B 2015)	1GB RAM & 900MHz quad-core ARM Cortex-A7 CPU	Ubuntu 32-bit
BS device (Macbook Pro 2012)	4GB RAM & 2.5 GHz dual-core i5 processor	Ubuntu
Edge device (Dell Desktop)	16 GB RAM & 1.8 GHz Intel Core i7 8565U	Ubuntu

We perform scalability analysis to evaluate the authentication time by varying the number of IoT devices in the network for four authentication mechanisms, namely Z-PMA, Yang et al. [1], Wang et al. [31], and Sharma et al. [32]. Figure 10(a) shows the authentication time for a varying number of IoT devices. The number of edge devices and BSs is constant (25 each). We observe that, despite having more steps in the authentication phase, the Z-PMA mechanism requires less time to mutually authenticate IoT devices and edge devices than the existing mechanisms. Z-PMA uses two layers (i.e., BS layer and edge layer) to achieve mutual authentication. Computationally expensive processes, like decryption using CRT, are carried out on BSs. Since existing mechanisms require IoT devices to perform expensive operations such as quadratic decryption using CRT, symmetric encryption, elliptic curve scalar multiplication, and finding Chebyshev Polynomial points, their authentication time is higher than for the Z-PMA mechanism.

TABLE 5. Average execution time (in milliseconds) for cryptographic operations using MIRACL and Chebyshev libraries.

Primitives	IoT device	BS device	Edge device
T_{RN}	0.120	0.070	0.010
T_{EXP}	0.228	0.093	0.039
T_{ME}	210.691	10.614	6.851
T_{MA}	0.435	0.123	0.007
T_{ESM}	143.590	5.610	2.010
T_{MI}	1.526	0.794	0.002
T_{MO}	1.150	0.228	0.001
T_{HF}	1.012	0.120	0.014
T_{MM}	2.988	0.674	0.061
T_{CP}	250.780	7.610	3.926
T_{MAC}	1.012	0.120	0.014
T_{AESE}	7.820	3.470	1.903
T_{AESD}	5.616	2.891	1.110
T_{QRE}	3.051	1.806	0.913
T_{ORD}	10.724	5.400	2.325

D. IoT TESTBED RESULTS AND ANALYSIS

The configurations of all devices are shown in Table 4. The BSs and edge devices run a Hyperledger Fabric (HLF) blockchain as docker images [20]. This blockchain stores the parameters listed in Table 3 in its ledger. The HLF blockchain is a permissioned blockchain and provides higher transaction throughput and lower confirmation latency than the Ethereum blockchain [20]. Table 5 shows the time taken to perform the cryptographic operations using the devices specified in Table 4. The cryptographic operations included in this analysis are Random Number generation (T_{RN}), Exponentiation (T_{EXP}), Modular Exponentiation (T_{ME}), Modular Addition (T_{MA}), ECC Scalar Multiplication (T_{ESM}), Modular Inverse (T_{MI}), Modular Reduction (T_{MO}), Hash Function (T_{HF}), Modular Multiplication (T_{MM}), Chebyshev Polynomial (T_{CP}), Message Authentication Code (T_{MAC}), AES Encryption/Decryption (T_{AESE}/T_{AESD}), and

TABLE 6. Comparison of authentication time (in milliseconds).

Mechanisms	IoT device	BS device	Edge device	Total
Wang *et al.* (2023) [31] (if no key generated)	$7T_{RN} + 4T_{HF} + 4T_{CP} + 2T_{ME} + 2T_{MO} + \big[(3T_{MM} + 1T_{MA})$ or $(5T_{MM})\big]$ = **1441.090 or 1446.630**	-	$4T_{HF} + 2T_{RN} + \big[(6T_{CP})$ or $(7T_{CP} + 2T_{MM})\big]$ = **23.630 or 27.680**	**1464.720 or 1474.310**
Wang *et al.* (2023) [31] (if key generated and about to expire)	$6T_{RN} + 4T_{HF} + 4T_{CP} + 1T_{MO} + 1T_{MA} + \big[(1T_{MM} + 1T_{MA})$ or $(3T_{MM})\big]$ = **1012.896 or 1018.437**	-	$4T_{HF} + 2T_{RN} + \big[(6T_{CP})$ or $(7T_{CP} + 2T_{MM})\big]$ = **23.630 or 27.680**	**1036.520 or 1046.117**
Sharma *et al.* (2023) [32]	$6T_{ESM} + 1T_{MA}$ = **861.975**	$5T_{ESM} + 2T_{MM} + 2T_{MI} + 1T_{MA} + 1T_{MO}$ = **31.337**	$2T_{MM} + 1T_{RN} + \big[(2T_{ME})$ or $(2T_{ME} + 1T_{MA})\big]$ = **13.834 or 13.841**	**907.146 or 907.153**
Yang *et al.* (2021) [1]	$2T_{HF} + 2T_{MAC} + 1T_{RN} + 1T_{QRE} + 1T_{QRD} + 1T_{AESE} + \big[(2T_{ME})$ or $(8T_{ME})\big]$ = **447.145 or 1711.291**	-	$5T_{HF} + 5T_{MAC} + 1T_{MO} + 1T_{QRD} + 1T_{AESD}$ = **3.576**	**450.721 or 1714.867**
Z-PMA (2024)	$5T_{HF} + 3T_{QRE} + 1T_{RN} + 1T_{MM} + 1T_{EXP}$ = **17.549**	$4T_{HF} + 2T_{RN} + 2T_{EXP} + 1T_{QRD} + 1T_{QRE} + 1T_{MM}$ = **8.686**	$4T_{HF} + 1T_{QRD}$ = **2.381**	**28.616**

QR Encryption/Decryption (T_{QRE}/T_{QRD}). We use the MIRACL library[1] and the Chebyshev library[2] to implement cryptographic operations for this analysis.

In Table 6, we compare the cryptographic operations of the Z-PMA mechanism with existing mechanisms, namely Yang et al. [1], Wang et al. [31], and Sharma et al. [32]. We divide the Wang et al. [31] mechanism into two phases, namely, if no key is generated, and if the key is generated and about to expire, because different cryptographic operations are used for each phase. We observe that the Z-PMA mechanism has the lowest authentication time than the existing mechanisms because Z-PMA performs lightweight, secure operations on IoT devices and computationally expensive operations such as T_{QRD} on BS devices and edge devices. In contrast, existing mechanisms perform computationally expensive operations such as T_{CP}, T_{ME}, and T_{ESM} on IoT devices.

E. NASH EQUILIBRIUM ANALYSIS

In this section, we analyze the interactive behavior between the IoT, edge, and BS devices in the Z-PMA mechanism. We also include a table of different scenarios to demonstrate the existence of a Nash equilibrium in which all three types of devices collaborate in the Z-PMA mechanism. We state a several assumptions and key-points before performing the analysis. From Equations (5) and (9a), we observe that the utility value of a BS device is a non-negative value. We assume that the utility value of an edge device is also non-negative as it is maximized in the optimization problem (Equation (9)) and obviously $\tau_i > \$_i$ in Equation (7). In addition, we also assume that the operations performed by the BS device other than the required mutual authentication process are not relevant to this analysis and are not rewarded by the edge devices.

In Table 7, we show the interactive behavior of the three types of devices along with the utility values for an edge device (E_{ED}) and a base station (E_{BS}). In the Z-PMA mechanism, each device can choose one of two actions. An IoT device can choose to authenticate or not authenticate. An edge device can choose to participate or not participate in the authentication process. A BS device can choose to collaborate or not collaborate as AA for the authentication process. Therefore, a total of eight scenarios are generated, as shown in Table 7. Furthermore, we also give the utility values for the edge device and base station to show the Nash equilibrium. Among all scenarios, we only observe non-zero positive utility values for the edge device and the BS device when the three devices of the Z-PMA mechanism choose to collaborate (i.e., the IoT device chooses to authenticate,

[1] https://github.com/miracl/MIRACL
[2] https://github.com/mlazaric/Chebyshev

TABLE 7. Interactive behavior table with specific utility values.

IoT Device Action	Edge Device Action	BS Action	Utility for Edge Device	Utility for BS
Authenticate	Participate	Collaborate	E_{ED}	E_{BS}
Authenticate	Participate	Not Collaborate	0	0
Authenticate	Not Participate	Collaborate	0	0
Authenticate	Not Participate	Not Collaborate	0	0
Not Authenticate	Participate	Collaborate	0	0
Not Authenticate	Participate	Not Collaborate	0	0
Not Authenticate	Not Participate	Collaborate	0	0
Not Authenticate	Not Participate	Not Collaborate	0	0

TABLE 8. Average execution time (in milliseconds) for cryptographic operations using the MIRACL and Chebyshev libraries for P2P energy trading scenario.

Primitives	SM device	PMU device	Data Control Center
T_{RN}	0.099	0.070	0.010
T_{EXP}	0.103	0.093	0.039
T_{ME}	198.328	10.614	6.851
T_{MO}	1.098	0.228	0.001
T_{HF}	0.971	0.120	0.014
T_{MM}	2.136	0.674	0.061
T_{MAC}	0.992	0.120	0.014
T_{AESE}	6.550	3.470	1.903
T_{AESD}	5.233	2.891	1.110
T_{QRE}	2.547	1.806	0.913
T_{ORD}	9.837	5.400	2.325

the edge device chooses to participate, and the BS device chooses to collaborate). Why? When an IoT device chooses not to authenticate, there is no work contract between edge devices and BS devices, resulting in zero utility values for both. Likewise, when the edge device chooses not to participate or the BS device chooses not to collaborate, there is no collaboration between the edge device and BS device, resulting in zero utility values for both.

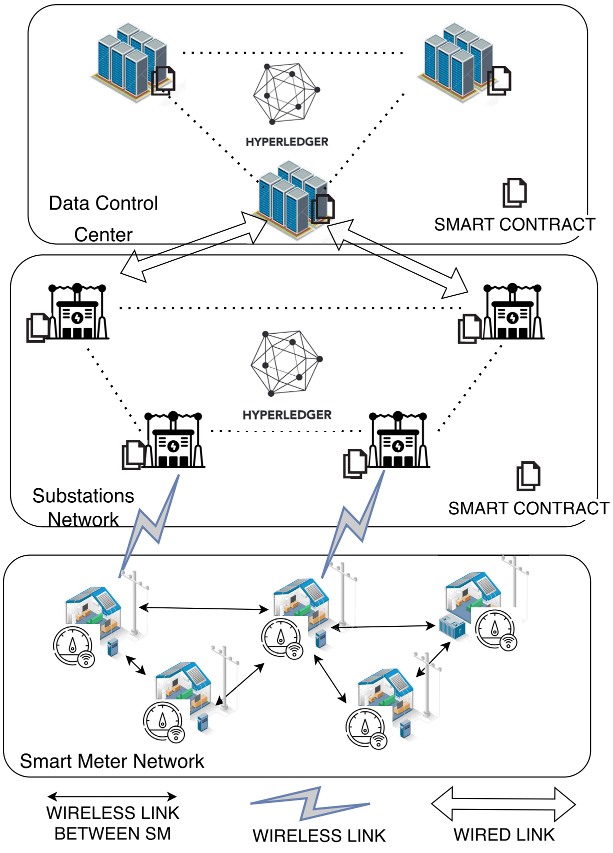


FIGURE 11. Z-PMA implementation in P2P energy trading scenario.

F. USE CASE: IMPLEMENTATION OF THE Z-PMA MECHANISM IN A PEER-TO-PEER ENERGY TRADING SCENARIO

To illustrate the practicality of the Z-PMA mechanism, we provide a use case for a decentralized peer-to-peer (P2P) energy trading scenario [54]. The Z-PMA mechanism architecture for this context is illustrated in Fig. 11. It includes the following components:

- Prosumers: Consist of energy purchasers (EPs) and energy sellers (ESs). These are typically electric vehicles (EVs), smart homes, and smart buildings engaged in P2P energy trading. Each prosumer has three energy trading options: purchasing energy, selling energy, or remaining idle, chosen based on their current energy levels and anticipated future demands.
- Smart Meters (SMs) (identical to IoT devices): Embedded devices in each prosumer setup that monitor and record the volume of energy transacted and the identity of the counterparties. The records maintained by smart meters are designed to be tamper-proof to ensure the integrity and reliability of transaction data.
- Substations (identical to BS devices): Act as local aggregators within the network, facilitating communication between smart meters and edge devices. They also serve as AAs in the Z-PMA mechanism, enhancing

TABLE 9. Authentication time (in milliseconds) for the P2P energy trading scenario.

Mechanisms	SM device	PMU device	Data Control Center	Total
Yang *et al.* (2021) [1]	**418.298 or 1608.266**	-	**3.576**	**421.874 or 1611.842**
Z-PMA (2024)	**14.834**	**8.686**	**2.381**	**25.901**

the scalability and reducing latency in authentication processes.

- Data Control Centers (identical to Edge Devices): More powerful computational hubs located strategically within the network, such as at utility company facilities or community centers. They manage the permissioned blockchain and handle the bulk of computational tasks, including the execution of smart contracts, and authentication of transactions using the Z-PMA mechanism.
- Blockchain Network: A permissioned blockchain operated by the edge devices, which securely logs all transactions, manages smart contracts, and ensures privacy and security through mutual authentication protocols. This blockchain effectively supports the scalability and privacy requirements of the P2P energy trading market.

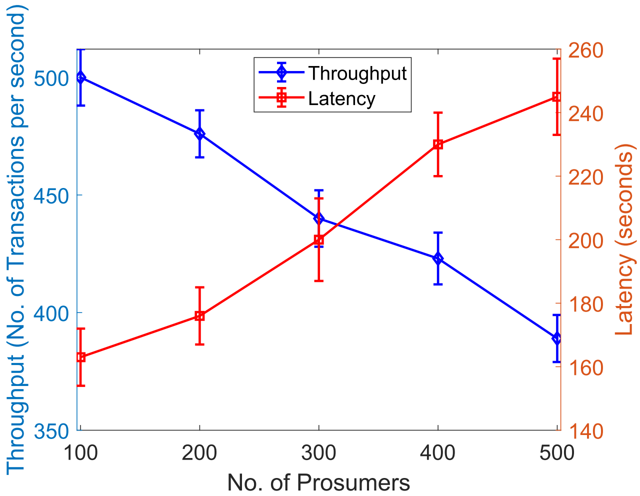


FIGURE 12. Scalability analysis by varying no. of prosumers for Z-PMA implementation in a P2P energy trading scenario.

The above architecture supports a secure, scalable, and efficient platform for P2P energy trading, showing how the Z-PMA mechanism can provide low latency, security, and privacy preservation in the practical context of energy trading. To further validate our claims, we provide a scalability analysis of the Z-PMA mechanism for a varying number of prosumers in terms of throughput and latency in Fig. 12. The throughput is defined as the number of transactions confirmed per second by the blockchain network and the latency is defined as the time taken for the authentication between prosumers for energy trading. In this analysis, we assume 25 substations and 25 data control centers and vary the number of prosumers (100 to 500). We use HLF blockchain network with Kafka consensus mechanism (similar to Section VI-C). In Fig. 12, we observe that the throughput decreases and the latency increases as the number of prosumers increases. This is because the number of transactions in the network increases with the number of prosumers. Consequently, the blockchain network requires more time to process the transactions needed to achieve authentication.

We analyze the authentication time of the Z-PMA mechanism to demonstrate its cost-effectiveness in a smart grid application scenario. In this analysis, a Raspberry Pi 3 model B+ with a 1 GB RAM and a 1.4 GHz quad-core ARM CPU is used as the central processing unit for a smart meter, a personal computer with a 4 GB RAM and a 2.5 GHz dual-core i5 CPU is used as the central processing unit for a Phase Measurement Unit (PMU), and a personal computer with a 16 GB RAM and a 1.8 GHz quad-core i7 8565U CPU is used as the Data Control Center node. The PMU is a component of the substation network layer that has computational processing capabilities [55], [56]. All devices run the Ubuntu operating system. Table 8 shows the execution time (in milliseconds) of the cryptographic operation on each device. Table 9 shows a comparison between the Yang et al. [1] and Z-PMA mechanisms in terms of authentication time (in milliseconds) to achieve authentication for the P2P energy trading scenario. We compare the Z-PMA mechanism with the Yang et al. [1] mechanism because both mechanisms employ ZKP-based technique for authentication in the blockchain-based IoT network. We observe that the time taken to achieve authentication is lower in the Z-PMA mechanism than the Yang et al. mechanism because the Z-PMA mechanism performs computational operations in the substation and data control center layers rather than in the smart meters layer. Therefore, we observe that the Z-PMA mechanism is cost-effective in this smart grid application scenario despite using the resource-intensive ZKP technique.

VII. CONCLUSION

In this article, we highlighted three main requirements for authentication mechanisms in low-latency IoT networks. The requirements are high security and privacy preservation, high scalability, and low authentication time. To fulfill these requirements, we proposed a novel, secure and privacy-preserving mutual authentication mechanism, named Z-PMA, for low-latency IoT networks. The Z-PMA mechanism uses a combination of zero-knowledge proof, an incentive mechanism, and a permissioned blockchain to provide secure, privacy-preserving, and scalable authentication for IoT networks. We performed four analyzes, namely informal security analysis, formal security analysis, simulation-based analysis, and IoT-based testbed analysis, to show that the requirements for high security and privacy preservation, high scalability, and low authentication time are satisfied. In our analysis, we compared the Z-PMA mechanism with an existing authentication mechanism to evaluate scalability and authentication time. The evaluation shows that the Z-PMA mechanism provides fast and secure authentication for low-latency IoT networks. For future work, we plan to investigate the use of Self-Sovereign Identities (SSI) based on ZKP and implement them in a blockchain network for performing mutual authentication in an IoT network. In addition, we plan to investigate a scalable solution of the PBFT consensus mechanism, which helps to overcome the 51% attack on the blockchain network.

REFERENCES

[1] X. Yang, X. Yang, X. Yi, I. Khalil, X. Zhou, D. He, X. Huang, and S. Nepal, "Blockchain-based secure and lightweight authentication for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3321–3332, Mar. 2022.

[2] J. Song, P.-W. Harn, K. Sakai, M.-T. Sun, and W.-S. Ku, "An RFID zero-knowledge authentication protocol based on quadratic residues," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12813–12824, Jul. 2022.

[3] G. Ramezan, A. Abdelnasser, B. Liu, W. Jiang, and F. Yang, "EAP-ZKP: A zero-knowledge proof based authentication protocol to prevent DDoS attacks at the edge in beyond 5G," in *Proc. IEEE 4th 5G World Forum (5GWF)*, Oct. 2021, pp. 259–264.

[4] A. Rasheed, R. N. Mahapatra, C. Varol, and K. Narashimha, "Exploiting zero knowledge proof and blockchains towards the enforcement of anonymity, data integrity and privacy (ADIP) in the IoT," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 3, pp. 1476–1491, Jul. 2022.

[5] A. D. Dwivedi, R. Singh, U. Ghosh, R. R. Mukkamala, A. Tolba, and O. Said, "Privacy preserving authentication system based on non-interactive zero-knowledge proof suitable for Internet of Things," *J. Ambient Intell. Humanized Comput.*, vol. 13, pp. 4639–4649, Sep. 2021.

[6] S. W. Shah, N. F. Syed, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "LCDA: Lightweight continuous device-to-device authentication for a zero trust architecture (ZTA)," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102351.

[7] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, "A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 525–552, 1st Quart., 2016.

[8] S. H. Islam, M. S. Obaidat, P. Vijayakumar, E. Abdulhay, F. Li, and M. K. C. Reddy, "A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs," *Future Gener. Comput. Syst.*, vol. 84, pp. 216–227, Jul. 2018.

[9] C. Wang, J. Shen, Q. Liu, Y. Ren, and T. Li, "A novel security scheme based on instant encrypted transmission for Internet of Things," *Secur. Commun. Netw.*, vol. 2018, pp. 1–7, Jan. 2018.

[10] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[11] I. Ali, Y. Chen, N. Ullah, R. Kumar, and W. He, "An efficient and provably secure ECC-based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1278–1291, Feb. 2021.

[12] S.-Y. Tan and T. Groß, "MoniPoly—An expressive Q-SDH-based anonymous attribute-based credential system," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Daejeon, South Korea. Cham, Switzerland: Springer, Dec. 2020, pp. 498–526.

[13] R. Casanova-Marqués, P. Pascacio, J. Hajny, and J. Torres-Sospedra, "Anonymous attribute-based credentials in collaborative indoor positioning systems," in *Proc. 18th Int. Conf. Secur. Cryptogr.*, 2021, pp. 1–7.

[14] J. L. C. Sanchez, J. B. Bernabe, and A. F. Skarmeta, "Integration of anonymous credential systems in IoT constrained environments," *IEEE Access*, vol. 6, pp. 4767–4778, 2018.

[15] L. Benarous, B. Kadri, S. Bitam, and A. Mellouk, "Privacy-preserving authentication scheme for on-road on-demand refilling of pseudonym in VANET," *Int. J. Commun. Syst.*, vol. 33, no. 10, p. 4087, 2020.

[16] F. Armknecht, L. Chen, A.-R. Sadeghi, and C. Wachsmann, "Anonymous authentication for RFID systems," in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*, Istanbul, Turkey. Cham, Switzerland: Springer, Jun. 2010, pp. 158–175.

[17] Q. Hu, Y. Dai, S. Li, and T. Jiang, "Enhancing account privacy in blockchain-based IoT access control via zero knowledge proof," *IEEE Netw.*, vol. 37, no. 6, pp. 117–123, Nov. 2023.

[18] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," *IEEE Netw.*, vol. 35, no. 4, pp. 198–205, Jul. 2021.

[19] S. Sahai, N. Singh, and P. Dayama, "Enabling privacy and traceability in supply chains using blockchain and zero knowledge proofs," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Nov. 2020, pp. 134–143.

[20] A. Pathak, I. Al-Anbagi, and H. J. Hamilton, "TABI: Trust-based ABAC mechanism for edge-IoT using blockchain technology," *IEEE Access*, vol. 11, pp. 36379–36398, 2023.

[21] T. Ivăscu, M. Frâtcu, and V. Negru, "Considerations towards security and privacy in the Internet of Things based eHealth applications," in *Proc. IEEE 14th Int. Symp. Intell. Syst. Inform. (SISY)*, 2016, pp. 275–280.

[22] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2017, pp. 618–623.

[23] L. Zöscher, J. Grosinger, R. Spreitzer, U. Muehlmann, H. Gross, and W. Bösch, "Concept for a security aware automatic fare collection system using HF/UHF dual band RFID transponders," in *Proc. 45th Eur. Solid-State Device Res. Conf. (ESSDERC)*, 2015, pp. 194–197.

[24] V. Shivraj, M. Rajan, M. Singh, and P. Balamuralidhar, "One time password authentication scheme based on elliptic curves for IoT," in *Proc. 5th Nat. Symp. Inf. Technol., Towards New Smart World (NSITNSW)*, 2015, pp. 1–6.

[25] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, "Anonymous secure framework in connected smart home environments," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 968–979, 2017.

[26] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors J.*, vol. 16, no. 1, pp. 254–264, 2015.

[27] P. Gope and T. Hwang, "Untraceable sensor movement in distributed IoT infrastructure," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5340–5348, Sep. 2015.

[28] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10626–10636, Dec. 2017.

[29] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of Vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[30] J. Lansky, A. M. Rahmani, S. Ali, N. Bagheri, M. Safkhani, O. H. Ahmed, and M. Hosseinzadeh, "BCmECC: A lightweight blockchain-based authentication and key agreement protocol for Internet of Things," *Mathematics*, vol. 9, no. 24, p. 3241, Dec. 2021.

[31] Z. Wang, J. Huang, K. Miao, X. Lv, Y. Chen, B. Su, L. Liu, and M. Han, "Lightweight zero-knowledge authentication scheme for IoT embedded devices," *Comput. Netw.*, vol. 236, 2023, Art. no. 110021.

[32] P. C. Sharma, M. R. Mahmood, H. Raja, N. S. Yadav, B. B. Gupta, and V. Arya, "Secure authentication and privacy-preserving blockchain for industrial Internet of Things," *Comput. Elect. Eng.*, vol. 108, 2023, Art. no. 108703.

[33] D. Hofheinz and E. Kiltz, "The group of signed quadratic residues and applications," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 2009, pp. 637–653.

[34] X. Jia, N. Hu, S. Yin, Y. Zhao, C. Zhang, and X. Cheng, "A2 chain: A blockchain-based decentralized authentication scheme for 5G-enabled IoT," *Mobile Inf. Syst.*, vol. 2020, pp. 1–19, Dec. 2020.

[35] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[36] I. A. Qasse, M. Abu Talib, and Q. Nasir, "Inter blockchain communication: A survey," in *Proc. ArabWIC 6th Annu. Int. Conf. Res. Track*, Mar. 2019, pp. 1–6.

[37] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[38] Q. Chen, K. Su, C. Liu, and Y. Xiao, "Automatic verification of web service protocols for epistemic specifications under Dolev–Yao model," in *Proc. Int. Conf. Service Sci.*, May 2010, pp. 49–54.

[39] W. Liu, J. Weng, B. Zhang, K. He, and J. Huang, "Improvements on non-interactive zero-knowledge proof systems related to quadratic residuosity languages," *Inf. Sci.*, vol. 613, pp. 324–343, Oct. 2022.

[40] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2004.

[41] Z. Hou, H. Chen, Y. Li, and B. Vucetic, "Incentive mechanism design for wireless energy harvesting-based Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2620–2632, Aug. 2018.

[42] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2144–2155, Oct. 2015.

[43] A. A. Elsaeidy, N. Jagannath, A. G. Sanchis, A. Jamalipour, and K. S. Munasinghe, "Replay attack detection in smart cities using deep learning," *IEEE Access*, vol. 8, pp. 137825–137837, 2020.

[44] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial," in *Version From*, 2018, pp. 5–16.

[45] N. Kobeissi, K. Bhargavan, and B. Blanchet, "Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Apr. 2017, pp. 435–450.

[46] S. Bussa, R. Sisto, and F. Valenza, "Formal verification of a V2X privacy preserving scheme using proverif," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2023, pp. 341–346.

[47] M. A. Khan, S. Ullah, T. Ahmad, K. Jawad, and A. Buriro, "Enhancing security and privacy in healthcare systems using a lightweight RFID protocol," *Sensors*, vol. 23, no. 12, p. 5518, Jun. 2023.

[48] X. Zhou, S. Wang, K. Wen, B. Hu, X. Tan, and Q. Xie, "Security-enhanced lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9599–9609, Mar. 2024.

[49] S. Kardas, S. Çelik, M. Sariyüce, and A. Levi, "A secure and private RFID authentication protocol based on quadratic residue," in *Proc. SoftCOM, 20th Int. Conf. Softw., Telecommun. Comput. Netw.*, Sep. 2012, pp. 1–6.

[50] X. Yang, X. Yi, I. Khalil, Y. Zeng, X. Huang, S. Nepal, X. Yang, and H. Cui, "A lightweight authentication scheme for vehicular ad hoc networks based on MSR," *Veh. Commun.*, vol. 15, pp. 16–27, Jan. 2019.

[51] *Docker*. Accessed: Oct. 11, 2022. [Online]. Available: https://docs.docker.com/get-docker/

[52] *Hyperledger Composer*. Accessed: Oct. 11, 2022. [Online]. Available: https://hyperledger.github.io/composer/latest/

[53] G. Yang, K. Lee, K. Lee, Y. Yoo, H. Lee, and C. Yoo, "Resource analysis of blockchain consensus algorithms in hyperledger fabric," *IEEE Access*, vol. 10, pp. 74902–74920, 2022.

[54] A. Pathak, I. Al-Anbagi, and H. J. Hamilton, "Privacy-preserving authentication mechanism for P2P energy trading in smart grid networks," in *Proc. IEEE Conf. Commun. (ICC)*, Aug. 2024, pp. 1–6.

[55] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke, "Synchronized phasor measurement applications in power systems," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 20–27, Jun. 2010.

[56] A. Bose, "Smart transmission grid applications and their supporting infrastructure," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 11–19, Jun. 2010.

ADITYA PATHAK (Graduate Student Member, IEEE) received the B.E. degree in electronic and communication engineering from Gujarat Technological University, India, in 2017, and the M.A.Sc. degree in electronic systems engineering from the University of Regina, Canada, in 2021. He is currently pursuing the Ph.D. degree in electronic systems engineering with the University of Regina. His research interests include blockchain, the Internet of Things (IoT), edge computing, trust-based security, joint optimization of security, quality of service (QoS), and energy consumption. He was the Past Student Activities Chair of the IEEE South Saskatchewan Section.

IRFAN AL-ANBAGI (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Ottawa, in October 2013. From 2013 to 2015, he was a Postdoctoral Fellow and the Product Development Manager of the SecCharge Project with the University of Ottawa. He is currently an Associate Professor with the Faculty of Engineering and Applied Science, University of Regina, and an Adjunct Professor with the Department of Electrical and Computer Engineering, College of Engineering, University of Saskatchewan. He is registered as a Professional Engineer with the Association of Professional Engineers and Geoscientists of Saskatchewan (APEGS) and a Professional Engineers Ontario (PEO). His research interests include security and reliability in cyber-physical systems, the Internet of Things (IoT) systems, and edge computing.

HOWARD J. HAMILTON received the B.Sc. and M.Sc. degrees in computational science from the University of Saskatchewan and the Ph.D. degree in computing science from Simon Fraser University. He is currently a Professor with the Department of Computer Science, University of Regina, Regina, Canada, where he has working, since 1991. He is also the Director of the University of Regina's Laboratory for Computational Discovery. He is the co-author of *Knowledge Discovery and Measures of Interest* and co-editor of four other books, including *Quality Measures for Data Mining*. His research interests include machine learning, data mining, blockchains, applying artificial intelligence to computer animation and computer games, and spatio-temporal data mining.