مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Secure and Trustworthy Smart Water Metering System Based on Blockchain Technology

Tareq Nasser Mahdi
tareq.nmg@uomustansiriyah.edu.iq
Mustansiriyah University, college of pharmacy, Baghdad, Iraq
0000-0001-6093-6765
07901490314

Abstract

With the advancement of Information and Communication Technologies (ICT), and software engineering applications, countries around the world are replacing their old mechanical water meters by smart water meters to avoid manual meter readings. A smart water metering system consists of thousands of constraint devices (such as smart water meters) deployed across the water distribution grid to enable near-real-time monitoring and data collection. Until now, Smart water metering infrastructures have been established utilizing a cloud-centric approach, in which all data obtained by smart water meters (SWMs) are transmitted and managed centrally using cloud services. However, with the continuous growth of connected devices, the cloud paradigm is confronted by security, privacy, and latency issues. The data collected by SWMs are the sensitive data of users and thus require a high level of security. Although several security approaches have been developed over the past years, they cannot be used directly with such constraint devices due to their complicated encryption and decryption operations. Additionally, most of them rely on a third party. To make the data secure and trustworthy in smart water metering infrastructures, we propose in this paper a three-tier, consortium (permissioned) Blockchain system spanning IoT meters, fog nodes, and cloud servers. SHA-256 hashing and Proof-of-Work are used at each layer. No external certificate authority is required. For validation purpose, we create a Blockchain network using the Python programming language. The network is composed of nodes which will represent SWMs, edge gateways, and cloud servers in a deployment. We build a web application based on Flask micro-framework to test the Blockchain and the nodes. Chaining each reading by hash makes later tampering with recorded data detectable. On a single

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

machine, computation time increased from 0.05 seconds at 10 transactions to 0.25 at 50.

Keywords: Blockchain, Smart Water Meter, IoT, Software Engineering, Cloud Computing.

1. Introduction

Due to the ageing of current water infrastructure and issues encountered by such infrastructure such as leakages, water pollution, etc., the concept of smart water grid has been recently introduced. A Smart Water Grid (SWG) builds on the traditional water grid by enabling continuous, on-demand, and bidirectional flow of information among smart water meters, sensing devices, water utilities, and water users (Amaxilatis et al., 2020). SWG deploy to customers' homes or offices, SWMs, in order to gather near-real-time water use data analysis and offer meaningful economic benefit. Data obtained by SWMs is provided to water utilities either upon request or on a regular schedule (Zhang et al., 2016). Based on the information obtained, water utilities may also adopt dynamic pricing strategies, giving customers the opportunity to adjust their water use and manage consumption more efficiently. Also, these data allow water utilities to analyze water demand, which helps better understanding of water end-use and in this way influence the design of urban water distribution grids. Additionally, the availability of SWMs data can also help water utilities to identify leakages and future malfunctions. The knowledge extracted from smart meters' data can also be combined with water consumption demand models in order to determine the factors that contribute to the water peak demand (Amaxilatis et al., 2020).

Until now, the smart water metering infrastructures, which are one of the critical smart city infrastructures, have been developed following a cloud-centric approach where all data collected by SWMs are sent to cloud servers for processing and storage. As smart cities expand, billions of sensors and communication devices become interconnected. Managing this scale of deployment puts pressure on centralized cloud infrastructure, which may no longer handle the growing data traffic efficiently, raising challenges in both end devices and core network domains, namely congestion, latency, privacy, and security (Mehmood et al., 2017; Perera et al., 2017). Basic authentication flaws in IoT devices are still being exploited today

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

(Banoth et al., 2026). Existing solutions have not closed this gap, which is what drives the approach taken in this paper. In this paper, we address the security challenge in the smart water metering system. In (Dogo et al., 2019), the authors identified four specific security threats to smart devices deployed into the water grid. These security threats include: (1) compromised trust in device-to-device communication, (2) physical tampering, (3) network hacking, jamming, device cloning, and eavesdropping, and (4) insecure over-the-air firmware updates for already-deployed devices. Specifically, in this paper, we address the first security threat mentioned by Dogo et al. (2019), which is the most important security threat to address in all IOT applications .

Building a smart water metering system consists of deploying into the water grid, thousands of SWMs to monitor and control the water grid. These metering devices are controlled remotely over a wireless network, and if a hacker gains access to the network it can breach confidentiality and integrity. In the past years, several approaches have been developed in order to address the security challenge in various applications (Hafsa et al., 2019; Yu & Köse, 2017). Unfortunately, these solutions are not directly applicable to water metering devices due to unique requirements and constraints such as limited computing resources and over-the-air communications

(Louw et al., 2016). The challenge is practical: SWMs are constrained devices pushing data through several network hops. Securing that data without a central authority, and without heavy cryptographic schemes that depend on third-party trust, is not straightforward.

In this paper, we propose a system to make secure and trustworthy the data collected by SWMs in smart water metering system by using Blockchain (BC) technology (Nakamoto, 2008) in a consortium (permissioned) configuration. BC is a distributed ledger deployed in a peer-to-peer (P2P) network where nodes participating in the network can create and broadcast continuously transactions. The generic term “data” is often replaced on the internet by the term “transactions.” Therefore, in order to avoid confusion and keep consistency, we will be using the term “transaction” to refer to data. BC is composed of blocks, which are cryptographically linked and time-stamped collections of transactions. It was introduced by Nakamoto (2008).

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

The main goal of this paper is securing data collected by SWMs in smart water metering system which is one important IOT application in smart cities. As the IoT ecosystem suffers from security issue and other technical challenges (Kimani et al., 2019; Wang et al., 2019), by introducing BC, we can make the system more robust. The other objective of our paper is data integrity: to make tampering with the collected data detectable.

BC technology is used here to secure the data collected by SWMs and to make tampering with them detectable. In particular, we aim to detect any modification of the data between collection and the cloud server. The proposed solution provides security services such as data integrity and makes data manipulation detectable. Such manipulation can lead to shocking bills or issues between the water utility and its customers.

This paper makes three contributions: a three-tier consortium blockchain design (meters, gateways, cloud) in which each reading is chained by hash, so later changes to recorded data can be detected; a Python/Flask prototype of the design; and measurements on one machine that compare the cost of PoW, PBFT and a simplified PoA model (Tables 1(a) and 1(b)). The experiments are software simulations and do not cover a deployed network.

In the next section, we review the recent works that focused on the security issue in the IOT ecosystem. In section 3, we provide a general overview of BC technology while section 4 discusses our proposed solution. Section 5 analyses the performance of our solution and Section 6 concludes the paper and provides future research directions.

2. Related Works

In the past years, several research papers have been carried out with the aim of addressing the security challenges in various software engineering applications. Specifically, most of the approaches implemented by researchers are based on public-key cryptography (PKC) which are too computationally expensive for direct implementation in the IoT ecosystem. Ray et al. (2016) addressed security validation for IoT devices to detect vulnerabilities at the design level. Xu et al. (2016) proposed a security enhancement scheme for IoT communications to protect against eavesdroppers with uncertain locations. Ntuli and Abu-Mahfouz (2016) proposed a simple security architecture for smart water management

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

system to ensure secure booting, secure communications and secure firmware updates, relying on a third party called "Authorization and Access Control Servers (AAC)."

Jose and Vijayalakshmi et al. (2018) first reviewed the major security issues in IoT, followed by an investigation of the security algorithms that can address these issues. Then, they proposed an efficient cryptographic method based on Elliptic Curve Cryptography (ECC) for IoT ecosystem. Similarly, Stergiou et al. (2018) presented a survey on the security challenges raised by the integration of IoT and Cloud Computing and then proposed a security model based on Rivest, Shamir and Adelman (RSA) algorithm. Yan et al. (2019) presented a function-based access control scheme using identity-based encryption for Internet of Things ecosystem. However, the above-mentioned approaches provide only authentication in a centralized manner and use the third party in various applications.

In this paper, we have employed the BC technology in a distributed manner that provides authentication as well as validation. Our solution is independent of third party, and is a peer-to-peer network based. Alsheakh et al. (2024) built a trust-by-design architecture for constrained IoT devices, targeting authentication and data integrity at the device level. The system here covers both of those, plus decentralization, across three layers, without a central authority.

Blockchain has also been applied more widely in water management. Xia et al. (2022) built a framework for intelligent water management. Satilmisoglu et al. (2024) looked at where Blockchain fits in water resources and where scalability gets difficult. Pahontu et al. (2023) put water monitoring on a decentralized Blockchain. Nododile and Nyirenda (2025) added IPFS to handle data storage at meter scale.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

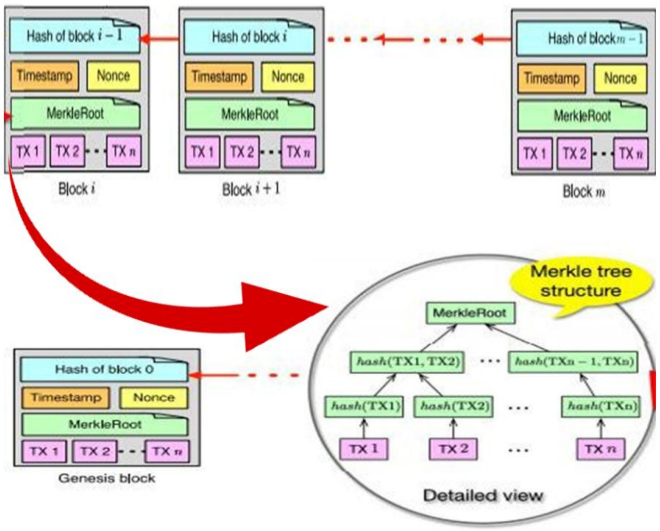

Figure - 1. Blockchain with a detailed view of the Merkle tree structure (Where TX denotes transaction)

3. Blockchain Technology Overview

Blockchain is a peer-to-peer distributed ledger technology originally proposed for cryptocurrency transactions, and has since found broader applications in security and IoT systems (Mohanta et al., 2019).

It is a chain of blocks structured in a chronological manner, where transactions contained in each block are hashed in a Merkle Tree to ensure data integrity (Nakamoto, 2008). Each block of the chain contains: (1) the block version (indicating the validation rules to follow), (2) the hash of the previous block, (3) timestamp recording the current time in seconds, (4) a nonce starting from 0 and increasing for every hash calculation and (5) the Merkle root (obtained by hashing the transaction hashes pairwise, level by level, until one hash remains). In Fig. 1, we illustrate the BC architecture where we show all the components of the BC. As each block contains the hash of the previous block, if someone in the peer-2-peer network attempts to alter a block, he/she will need to modify that block and every block that follows it, which makes tampering with the chain easy to detect. The block of our prototype is simpler: it hashes the transactions directly and has no Merkle root (Section 4.4).

Certain nodes in the Blockchain network, known as mining nodes, work at each moment to generate a valid block by solving a difficult

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

mathematical puzzle, where the node that finds the solution first broadcasts the new block to the network for validation. In Bitcoin, this puzzle is known as Proof-of-Work (PoW). There are other consensus algorithms such as proof of stake (PoS), and practical Byzantine fault tolerance (PBFT) used in other cryptocurrencies. This puzzle is hard to solve yet straightforward to verify. When a miner or mining node finds a valid solution, it sends the result to other nodes in the Blockchain network to verify and validate. PoW, however, comes at a real cost in terms of energy and computation. In Bitcoin, mining nodes increment a special variable called Nonce until the block hash meets the required difficulty target (Nakamoto, 2008).

4. Proposed Work

In this section, we first present the architecture of our solution, we describe how data are collected and transmitted, and we show how the BC nodes are connected, how blocks are created and validated and finally present our implementation. All full nodes across these layers are operated by a single water-utility authority, making this a consortium (permissioned) Blockchain rather than an open, trustless network.

4.1. Architecture of the Proposed Solution

Our proposed solution consists of three layers such as the perception layer, the fog layer and the cloud layer. We describe the functionalities and the role of each layer as follows:

Perception layer: The perception layer consists of SWMs equipped with LoRaWAN interfaces. LoRaWAN is a communication protocol based on LoRa modulation, developed and maintained by the LoRa Alliance. Its network follows a star-of-stars topology, where gateways (GWs) relay messages between end-devices (EDs) and the network server (NS). Each ED uses single-hop wireless communication to reach one or more gateways. Each message sent by an ED is received by all gateways located in its communication range. The redundant information will be removed by the NS. LoRaWAN gateways connect to the NS via standard IP links. The underlying LoRa modulation is based on Chirp Spread Spectrum (CSS), supporting six orthogonal Spreading Factors (SF7 to SF12) within a fixed channel bandwidth. This scheme covers distances of 1–5 km in urban areas and up to 15 km in rural areas at very low power consumption (Lalle et al., 2019, 2020). LoRaWAN

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

operates in licensed-free bands (for instance, in Europe it operates in 868 MHz bands, 915 MHz in US and 433 MHz in Asia) with configurable SF and configurable bandwidth (125 KHz, 250 KHz and 500 KHz). Depending on the SF, the data rates vary from 300 bps to 50 kbps in Europe and 900 bps to 100 kbps in US (Lalle et al., 2019). LoRaWAN protocol is largely used in IoT applications as well in smart water metering applications (Lalle et al., 2020). The SWMs deployed in this layer are limited in terms of power and storage space. Therefore, they cannot store the whole BC or participate in the mining process (creation of new block by solving the mathematical puzzle). We call these nodes lightweight nodes. They can only validate the trustfulness of transactions without downloading the entire BC. As shown in Fig. 2, they will just save the hash values of transactions.

Fog layer: The fog layer consists of LoRa gateways and fog servers. These devices are called full nodes in this paper because they have a large storage space to save the entire BC and strong computing capability of solving the mathematical puzzle. The LoRa gateways are in charge of receiving data collected by SWMs, identify the source of each information and prioritize its upload to the fog server. The fog servers after receiving the data from gateways, perform local processing and store the data for the mining process. Then, the data is sent to the cloud server for other processing tasks. It is worth mentioning that the gateways communicate with the fog servers using IP standard connection.

Cloud layer: It contains a cloud server with large storage space and high power for solving the PoW. The communication between fog servers and cloud server is achieved through IP standard connection. At each time instant, nodes that take part to the mining process (such as gateways, fog servers and cloud server) try to create a new block by solving the PoW. Once a full node finds a solution to the PoW, it creates a new block and broadcasts this block to other nodes for validation. After validation, each full node appends this block to the BC stored in its memory space. The SWMs, on the other hand, just save the hash values in order to validate the trustfulness of transactions.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

As SWMs have energy constraint and support only lightweight operations, in our paper, we call these nodes lightweight nodes and other nodes like base stations, gateways, fog servers and cloud servers are called full nodes. The full nodes have a large storage space to save the entire BC and strong computing capability of solving the PoW (Dai et al., 2019). In our paper, the gateways that relay data from SWMs are LoRaWAN gateways. According to LoRaWAN specification, each gateway has a memory space of 32 GB which is largely sufficient to store the entire BC. Based on the transaction structure in Fig. 9 (under 500 bytes each), a conservative estimate of 10,000 SWMs reporting every 10 minutes produces roughly 0.72 GB of raw data per day, giving the 32 GB gateway about 44 days under ideal conditions. In practice, blockchain overhead (block headers, Merkle structures) will reduce this figure, and city-scale deployments will need off-chain archiving or periodic storage rotation. Although there are several initiatives such as Raspnode and EthArmbian that address the initialization of BC at resource-constraint devices, Raspnode targets Bitcoin installation on Raspberry Pi microcomputers, while EthArmbian provides a Linux image for ARM-based Ethereum nodes, due to limited storage of SWMs, and particular requirements of these metering devices (such as installation in harsh environments which requires low power consumption in order to extend the lifetime of these devices), we still keep SWMs as lightweight devices. Routing strategies for such multi-hop LoRaWAN networks have been studied to optimize data forwarding between nodes (Lalle et al., 2021).

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

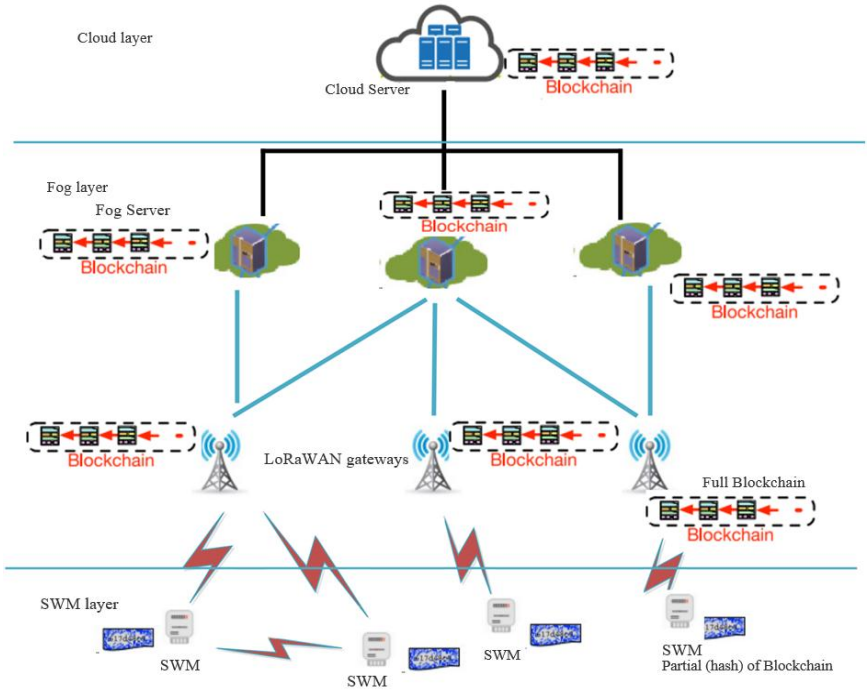

Figure - 2. Architecture of the proposed solution.

4.2. Data Gathering and Processing

In order to protect users' privacy, during data collection, we assigning a pseudonym to each SWM instead of using the meter's real identity. During data processing, the pseudonym is considered as the SWM public key or its ID. In this work, a new block is created once a predefined number of transactions is stored in the block and each SWM sends his water consumption data every 10 min for example. Every 10 minutes, each SWM combines his water consumption reading d, the timestamp tstamp, and its pseudonym pk (pk is the SWM's public key used as its pseudonym). Each SWM keeps one fixed key pair, so pk does not change. Each SWM then hashes the concatenated data and signs the digest with its private key sk. Finally, the water consumption reading d, timestamp stamp, pseudonym pk, and digital signature are sent in plaintext to LoRaWAN gateways. Each SWM transmits the following message:

$$[d, tstamp, pk, Sig_sk(SHA256(d + tstamp + pk))] \quad (1)$$

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Where Sig_sk(SHA256(d + tstamp + pk)) is the signature obtained by hashing d, tstamp and pk together with SHA-256 and signing the digest with the SWM's private key sk.

When the LoRaWAN gateways receive [d, tstamp, pk, Sig_sk(SHA256(d + tstamp + pk))] from SWMs, they compute the SHA-256 digest of (d, tstamp, pk) and verify the signature with the sender's public key pk. A valid signature shows that the message was signed with sk and was not changed; otherwise the transaction is considered invalid and is rejected. The process of data collection and transmission is shown in Fig. 3. If the message is authentic, the LoRaWAN gateways store it for the mining process and send the same data to fog servers. The fog servers will then send the data to the cloud server. It is worth mentioning that, according to LoRaWAN network features, a message sent by a SWM can be received by all gateways located in its communication range (Lalle et al., 2019). This will speed up the data communication between the nodes of the P2P network. The communication between BC network nodes is the same as shown in Fig. 3. The difference is that BC nodes save the entire BC for mining process while SWMs save only the hash values.

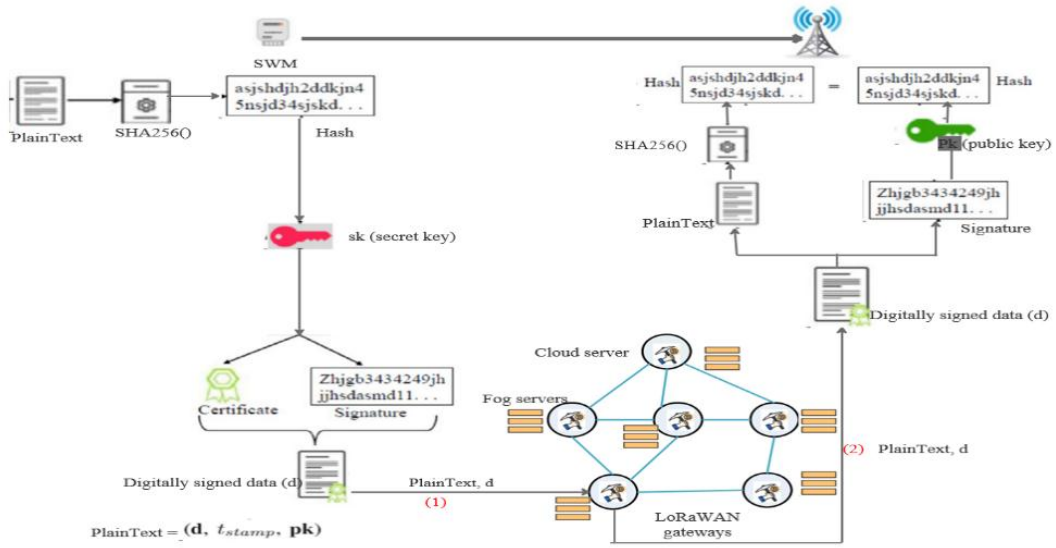

Figure - 3. Data collection and transmission.

4.3. Our Peer-to-Peer Network

As mentioned earlier, BC is a distributed ledger deployed in a P2P network where nodes participating in the network can create and

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

broadcast continuously transactions. In this part, we illustrate how the nodes that participate in our BC network are connected. SWMs are lightweight nodes and just save only the hash values without taking part neither in the mining process nor saving the entire BC. Our P2P network is shown in Fig. 4, where cloud server, fog servers and LoRaWAN gateways are connected into P2P manner. In the following we will consider two gateways, three fog servers, and one cloud server. The same assumption will be considered in the implementation part.

4.4. Block Creation and Validation

In our proposed solution, we follow that assumption. The first miner node that finds a solution to the mathematical puzzle can create a new block. The created block is broadcasted to other nodes for validation. In bitcoin, Satoshi Nakamoto (2008) used HashCash for solving the mathematical puzzle. When the n number of transactions is reached, the miner nodes start the mining process. The miner nodes search for a nonce such that the hash of the block (index, timestamp, transactions, previous hash and nonce) meets the difficulty target.

.

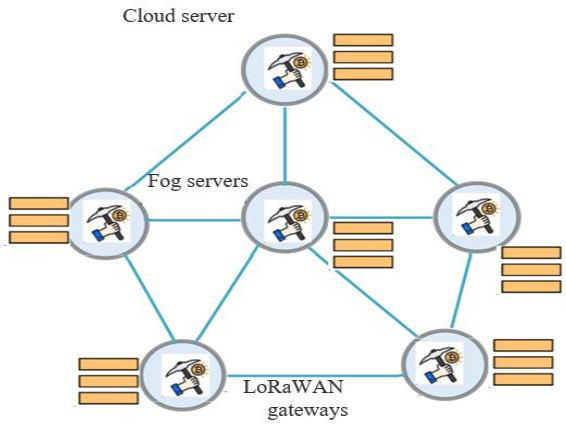

Figure - 4. Our P2P network.

Once a mining node finds the solution to the problem, it creates a block by applying the following steps:

- The mining node collects the pending transactions into the block.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

- It records the index, the timestamp, the hash of the previous block and the nonce it found.
- Finally, it sends the created block to other mining nodes for validation.

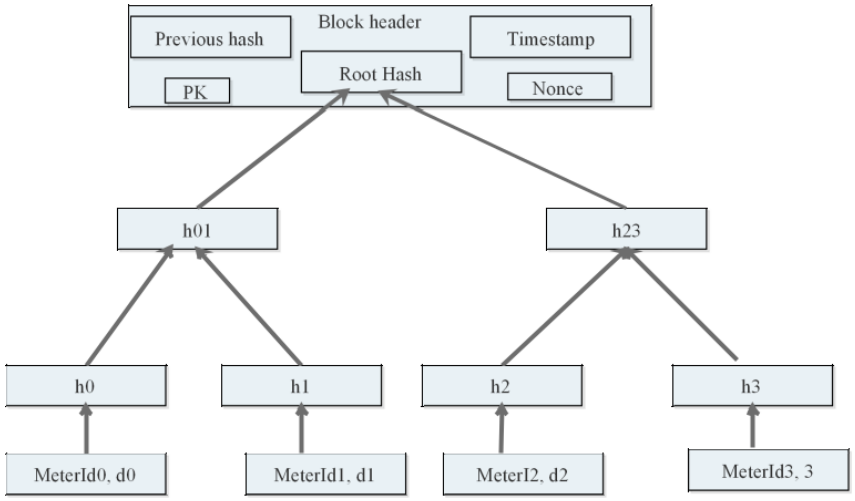

Figure - 5. General block structure.

Fig. 5 shows the general structure of a block. Our prototype uses the simpler block described above and hashes the transactions directly. When the nodes of our BC receive the created block, they verify all transactions stored in the block. If all transactions are correct, the block will be appended to the BC store in their memory space.

4.5. Implementation

As mentioned previously, our solution provides security services such as data integrity and makes data manipulation detectable from data collection at SWMs side to where they are processed definitively (cloud layer). To achieve this goal, we created, using Python programming language, a BC network consisting of 6 virtual nodes (2 gateways, 3 fog servers and 1 cloud server). The simulation ran on a personal computer (Intel Core i7, 16 GB RAM, Windows 10) using Python 3.9 and Flask. The main goal of this implementation is to detect any modification by a hacker of the data collected and sent by a SWM, as it will be sent wirelessly through LoRaWAN and cellular networks. Therefore, the reading of the meters must be the same at SWMs level, gateways level, fog server's level, and cloud server level. Our BC network is set up at

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

port 8000 and the nodes are running using ports 8001, 8002, 8003, 8004, 8005, 8006 (SWMs do not take part to the BC network).

A meter reading sent by a SWM is susceptible to tampering while sending to the BC network. To address this problem, as in bitcoin, the protocol uses public-private key cryptography (Section 4.2). Every new transaction is signed with the node's private key. This signature is added to the transaction data along with the node information. During the verification phase, while mining the transactions, miners can verify if the claimed owner of the data (meter reading) is the same as the one specified in the transaction data, and also that the data has not been modified. In this paper, a transaction, shown in Fig. 6, is defined in the form of JSON format.

The creation of our BC consists of 3 steps described as follows: Creation of block: Each block holds an index, a timestamp, a set of transactions, the previous block's hash, a nonce and its own hash. Algorithm 1 details the creation steps.

```
20  {
21      watermeterId: "",
22      DateTime: "",
23      Reading: ""
24
25  }
```

Figure - 6. Transaction.

Algorithm 1: Block Creation

Input: Transaction list T, previous block hash P_hash

Output: New candidate block B

1: Initialize empty block B
2: Set B.index ← current chain length + 1
3: Set B.timestamp ← current system time
4: Set B.transactions ← T
5: Set B.previous_hash ← P_hash

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

6: Set B.nonce ← 0

7: Compute B.hash ← SHA256(B.index || B.timestamp || B.transactions || B.previous_hash || B.nonce)

8: Return B

Add transaction to the block: Valid transactions are added to the pending pool and collected into the block at mining time. Algorithm 2 formalizes this process:

Algorithm 2: Add Transaction to Block

Input: Transaction message M = [d, tstamp, pk, Sig_sk(SHA256(d||tstamp||pk))]

Output: Updated pending transaction pool

1: Compute hash_computed ← SHA256(d || tstamp || pk)
2: valid ← Verify_pk(hash_computed, Sig)
3: If valid Then
4: Add transaction to pending pool
5: Broadcast transaction to all P2P nodes
6: Else
7: Reject transaction
8: End If

Mining: As transactions are continuously added to the block, at a certain time, the block will be full. Thus, the mining process will be performed in order to add this block to the chain. The mining process produces a hash of the created block, which cannot be changed without redoing the PoW. Changing a block means redoing the PoW of this block and the PoW of all the following blocks, which is very hard to do. Therefore, any later change to a recorded reading can be detected without a central entity. Algorithm 3 formalizes the mining process:

Algorithm 3: Mining Process

Input: Candidate block B (Algorithm 1), pending transaction pool, difficulty target D

Output: Valid mined block B

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

1: Collect pending transactions into B
2: Set B.nonce ← 0
3: Repeat
4: Compute B.hash ← SHA256(B.index || B.timestamp || B.transactions || B.previous_hash || B.nonce)
5: If B.hash satisfies difficulty target D Then
6: Broadcast B to all network nodes
7: For each node in network Do
8: Verify all transactions in B
9: If valid Then append B to local Blockchain
10: End For
11: Return B
12: End If
13: Increment B.nonce ← B.nonce + 1
14: Until valid block found

It is worth mentioning that there exist other functions and source code in our implementation; we showed here the most important ones. These three algorithms formalize the design. The prototype is described next. After the implementation of our BC system, we created a web interface with Python using the Flask framework, as it is a lightweight framework which eliminates the need for a third party to provide functional components. The main interface of our web application is shown in Fig. 7.



Figure - 7. Web application interface.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Then, we created mainly two routes, one to connect or add a node and another to post a transaction. To connect to the system, each node sends a connection request using POST method. We connect the 6 nodes (2 gateways, 3 fog servers and 1 cloud server) that will take part in the network. The public key or the address of the node is included in the request message. After connecting these 6 nodes, we can now post a transaction. The main goal is to detect data manipulation. Therefore, any tampering with the data between the SWMs and the cloud server must be detectable. Through the Web interface shown in Fig. 7, transactions can be posted using the POST method.

The web interface that we created plays the role of SWMs by allowing us to post or send transactions to the P2P network. In order to see how SWMs will send the water consumption data to the P2P in a deployment, we perform a test in our Laboratory by using two computers which represent the SWMs. The experimental set up is shown in Fig. 8. Then, we used a tunneling service, especially ngrok, to create a public URL and be able to interact with multiple machines. ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. Our web application is running on port 5000. The two computers act as SWM by enabling us to post transactions. They use https://fa2d1471ace8.ngrok.io address in order to send transactions to the P2P network. The virtual BC nodes are running on one computer using the above-mentioned ports (8001–8006).

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

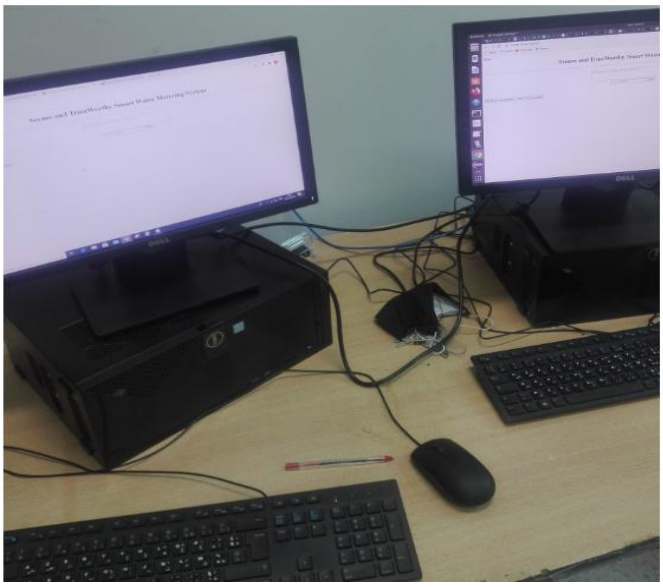

Figure - 8. Experimental setup.

When a certain number of transactions is added to the block, we need to mine the block and add it to the chain. By calling Algorithm 3, the block will be mined and added to the chain. In the prototype, mining is started from the web interface (Fig. 7). By using POSTMAN, the features of the mined block can be seen in Fig. 9. As we can see in Fig. 9, the block contains the hash of the previous block and timestamp which are the main components of BC technology. Any change in the block will automatically change the hash of this block and will require redoing the PoW of this block and the PoW of all the following blocks in order to make the BC valid again, which is very hard to do. Therefore, a recorded transaction (water meter reading) cannot be altered without redoing that work, so tampering becomes detectable. The transactions in Figs. 6 and 9 carry the meter ID, the reading and a timestamp. The signature of Section 4.2 belongs to the protocol design and is not shown in the prototype.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752


```
{
    "index": 2,
    "nonce": 47,
    "previous_hash": "008365097b19964aad0802adb8724e2ef18c86585734462a2065f9a5a4f7a6da",
    "timestamp": 1605183279.42733,
    "transactions": [
        {
            "meterId": "meter_1",
            "reading": "104",
            "timestamp": 1605183238.2282312
        },
        {
            "meterId": "meter_3",
            "reading": "108",
            "timestamp": 1605183249.9405704
        },
        {
            "meterId": "meter_2",
            "reading": "245",
            "timestamp": 1605183265.3219404
        }
    ]
}
```


Figure - 9. Result of mined block using POSTMAN in JSON format.

5. Security Analysis

In this section, we evaluate the performance of our proposed solution. Our security architecture fully considers the advantages of BC technology. In contrast to solutions that are only based on cloud paradigm, BC is a decentralization system which has an advantage in defending against DOS or DDOS attacks. While the single point of failure problem can happen in the cloud-based solutions, BC does not suffer from this problem because all nodes that take part in the network have a full copy of the BC. In cloud-based solutions, the cloud server is managed or controlled by a manager. If an attacker steals the account of manager, he/she can arbitrarily modify the water meter reading data stored in the server. In contrast, the data in the BC network is tamper resistant.

We consider three attackers: an outsider who alters or injects messages on the LoRaWAN or cellular link, a compromised meter or one whose private key was stolen, and a malicious consortium node that alters recorded blocks. The design makes the first and third detectable after recording. A compromised meter can sign a false reading, and the chain then stores it unchanged, so the blockchain cannot show that a meter measured correctly. The design has no key revocation. Algorithm 2 does

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

not check for duplicate or old messages, so replayed measurements are not covered.

A 51% attack needs majority control of mining nodes, all of which run under water utility authorities in this setup, so the attack surface is narrow (Nakamoto, 2008). With only six nodes, selfish mining is not worth worrying about. Sybil attacks require forging key pairs; since each node has a unique public-private pair, that is not straightforward (Dogo et al., 2019). And because every node connects directly to every other, there is no isolated node to eclipse. All gateway, fog, and cloud server operators are administered by the water utility. Blockchain integrity here does not rest on that administrative trust: an operator who alters a block must recompute the PoW for that block and every block after it, a cost that grows with the difficulty target and with the number of later blocks. The experiments of Section 5.1 use 16 bits, and the prototype appears to have used a lower difficulty. This also addresses insider threats at the fog and gateway layers. As Algorithm 2 specifies, a transaction whose signature check fails is rejected before it is added to a block. A modified block invalidates the chain beyond that point, since every node independently validates incoming transactions and holds a full copy of the chain. This design is best understood as a consortium (permissioned) blockchain, where decentralization is scoped to verification among known operators rather than open, trustless participation; a deliberate trade-off given the closed, utility-operated nature of smart metering infrastructure. We also clarify that decentralization at the data layer is nonetheless preserved: every full node independently maintains and validates a complete copy of the ledger, so there is no single point of failure or single authoritative database, even though participation in that ledger is administratively scoped to the utility's own infrastructure.

We chose PoW because it is simple to implement, its security behavior is well studied, and it needs neither a list of authorities (PoA) nor a token economy (PoS). PBFT needs repeated message rounds between all nodes, and its cost grows quadratically with n (Section 5.1). In our measurements the simplified PoA model was cheaper than both PoW and PBFT, and PoW gives no consensus finality, so we do not claim PoW is the best choice here. Tables 1(a) and 1(b) compare the three

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

mechanisms. Energy was not measured. Testing the alternatives in the full prototype is future work.

In cloud-based solutions all smart meters' measurements are sent to the centralized cloud server. Users' privacy constitutes a threat if all measurements are always transmitted to the cloud. An adversary can easily extract user's lifestyle profiles from the detailed consumption data collected using a Non-intrusive load monitoring (NILM) technique for water. BC network can provide the autonomy service by public key cryptography mechanism. In addition, by using fog servers in our proposed solution, some data can be processed locally and help protect users' privacy. The privacy gained is limited. Each meter uses one fixed key pair, so any ledger node can link the readings of one meter, and the utility can link a key to a customer. Periodic pseudonym rotation would reduce this and is future work; it is not implemented.

In what follows, we will evaluate our solution in terms of PoW cost and computation time as the number of transactions grows. Blockchain performance is tested through two metrics: PoW computational cost (Section 5.1) and computation time against a growing transaction count (Section 5.2). Throughput, block confirmation time and round-trip time are measured on one machine in Table 2. Latency over a real network would need a distributed testbed and was not measured; the analyses that follow extend this simulation-based validation analytically rather than through physical deployment. These metrics are crucial to evaluate the performance of our proposed solution because if a hacker gains access to a block and modifies this block, he/she will redo the PoW of this block and the PoW of all the following blocks in order to make the BC valid again, which becomes costlier with every later block.

5.1. PoW Cost

Assuming the message digest produced by SHA-256 is 256 bits, the number of bits hashed for each block header is:

$$Hout = SHA256([256+256+32+32+32]) = SHA256(608\ bits)\ . \quad (2)$$

where, for the general block header of Section 3, the previous hash and Hash root are 256 bits each and timestamp, nonce, and pk (public key of the miner that found the solution to the puzzle) are 32 bits each. The total input size is 608 bits and the final output calculated after message digest is 256 bits using SHA-256 function.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

The input size is fixed per hash and is not the cost of PoW. The cost is the number of hash attempts. With a difficulty of d leading zero bits, a node needs 2^d attempts on average (65,536 at d = 16), and the mining time is random: it is roughly geometric, with a standard deviation close to its mean. In 200 runs at d = 16, the mean number of hashes was 68,130, close to the expected 65,536. PoW needs no extra message rounds, so its message count grows only linearly with n. BFT-style consensus such as PBFT requires O(n2) communication among validating nodes, a cost that grows sharply as the network expands from the six-node testbed toward the node counts a city-wide deployment would need. Simulation results support this trade-off. PoW's power draw stays roughly flat regardless of node count, while PBFT's three-phase voting drives energy consumption up quadratically as nodes are added (Bhavana et al., 2025). Token-based alternatives such as PoS report energy savings exceeding 99% relative to PoW in deployed networks (Ethereum Foundation, n.d.), though at the cost of a token economy, which this closed, single-operator network does not need.

To move beyond a citation-based comparison, we implemented and measured three consensus mechanisms directly (PoW, PBFT and a simplified PoA model), calibrating the PoW difficulty to 16 bits (four leading hexadecimal zeros) to keep mining time measurable within a single experimental run while preserving the same hashing procedure described in Section 4.4. Fig. 9 suggests that the prototype used a lower difficulty (the previous hash starts with two zeros), so the 16 bits apply to the experiments of this section only. These measurements are consistent with the trend reported in the cited literature (Bhavana et al., 2025; Ethereum Foundation, n.d.): we detected no dependence of PoW cost on node count at these sample sizes, while PBFT cost grows with the number of validating nodes.

The experiments of this section are standalone Python scripts (standard library only: hashlib, sockets, multiprocessing). They ran on a cloud virtual machine with one Xeon 2.8 GHz core, 4 GB RAM and Python 3.12.3, not on the Flask prototype, which ran on the Intel Core i7 machine of Section 4.5. Tables 1(a) and 2 report the mean and sample standard deviation over N runs, with N stated in each table. No run was excluded, and random seeds were not fixed.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

A single-process simulation of the three-phase PBFT protocol (pre-prepare, prepare, commit) gives the expected $O(n^2)$ message growth: from 65 messages at 6 nodes to 18,335 at 96, a 282-fold increase for a 16-fold increase in node count (Table 1(a)). We repeated the measurements with each node as a separate operating-system process on the same machine, communicating over TCP sockets. The PBFT and PoA times are higher, and mostly reflect process and connection overhead on one machine, not a real network. PoW times show no clear growth with n at these sample sizes. PoA times grow roughly in proportion to n, and PBFT times grow much faster, in both setups.

Table 1(a). Cost of PoW, PBFT and a simplified PoA model, mean ± SD in ms (single-process N = 30, cross-process N = 10).

n	Mechanism	Messages	Single-process (ms)	Cross-process (ms)
6	**PoW**	5	489 ± 466	471 ± 384
6	**PBFT**	65	0.074 ± 0.018	20.4 ± 7.7
6	**PoA**	5	0.0075 ± 0.0004	3.6 ± 0.7
12	**PoW**	11	740 ± 565	498 ± 583
12	**PBFT**	275	0.294 ± 0.051	84.3 ± 24.9
12	**PoA**	11	0.017 ± 0.010	7.0 ± 0.9
24	**PoW**	23	788 ± 731	360 ± 217
24	**PBFT**	1,127	1.35 ± 0.42	333 ± 61
24	**PoA**	23	0.030 ± 0.011	13.0 ± 2.3
48	**PoW**	47	641 ± 485	537 ± 354
48	**PBFT**	4,559	5.41 ± 1.36	1,399 ± 100
48	**PoA**	47	0.052 ± 0.008	28.5 ± 4.4
96	**PoW**	95	695 ± 604	428 ± 436
96	**PBFT**	18,335	20.7 ± 3.5	6,589 ± 583
96	**PoA**	95	0.106 ± 0.023	52.2 ± 5.8

Single-process: all nodes are simulated in one process, and each received message costs one SHA-256 evaluation (a stand-in for signature verification). PoW time is mining plus (n − 1) block verifications. Cross-process: each node is a separate operating-system process on the same machine, and messages travel over TCP on localhost.

The three cross-process sessions were run at different times on the same machine and are not pooled; the table gives session 3.

PBFT: three phases; every node waits for the messages of all other nodes (no faults, no quorum shortcut), so the times are an upper bound. PoA: our simplified model, not a real PoA implementation; one round-robin authority signs once and broadcasts to n − 1 nodes. PoW: difficulty of 16 bits (four leading hexadecimal zeros).

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Cross-process times mostly reflect process and TCP overhead on one machine, not network latency. They include one start message and n completion reports, identical for every mechanism.

Table 1(b) compares the three mechanisms. PoA here is our simplified model (one authority per round signs once and broadcasts to the other n − 1 nodes), not Aura or Clique. The consistency classes come from De Angelis et al. (De Angelis et al., 2018): PBFT favours consistency (CP), Aura and Clique favour availability (AP), and PoW has no consensus finality because forks can happen. Energy was not measured.

Table 1(b). Qualitative comparison of PoW, PBFT and PoA.

Property	PoW	PBFT	PoA
Message rounds per block	1 broadcast	3 phases	1 broadcast (our model)
Messages per block	n − 1	(n − 1) + 2n(n − 1)	n − 1
Time in our measurements (Table 1(a))	Random, no dependence on n detected	Grows with n	Lowest of the three (simplified model)
Consensus finality and CAP class	No finality; forks can happen	Finality; CP	AP (Aura, Clique)
Energy	Not measured	Not measured	Not measured

5.2. Computation Time

In this section, we computed the computation time with respect to an increase in the number of transactions. In computation, the operations used are addition (add transaction in a block), one-way hashing function (SHA-256) and append (append the new block created to the chain). It is clear that increasing the number of transactions will increase the computation time. According to our simulation, the computation time with respect to an increase in the number of transactions is displayed in Fig. 10; computation time grows linearly with the number of transactions. The evaluation ran six virtual nodes on a single machine, using ngrok in place of a real network. The computation time grows linearly under those conditions. We do not present this as evidence of performance in a large-scale water network. Latency over a real network and block propagation

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

across the three tiers were not measured. Neither were packet loss, LoRaWAN interference, gateway congestion, or the energy use of meters and nodes. The multi-hop nature of LoRaWAN introduces gateway-level propagation delays that neither the single-machine setup nor the ngrok-based testbed can capture. A simple analytical check confirms this: at SF10 (125 kHz bandwidth, ~50-byte payload), a single hop's Time-on-Air is approximately 0.58 s by the standard LoRaWAN airtime formula. A two-hop relay path, typical of the topology surveyed in Lalle et al. (2021), therefore adds roughly 1.1 to 1.2 s of transmission delay alone, before any gateway queuing is even considered. The linear growth result is also limited in scope: it covers only 50 transactions on one machine, with no radio contention involved. At city scale, thousands of SWMs transmitting simultaneously will cause packet collisions and channel saturation at the gateway. This is a network-layer problem rather than a blockchain-layer one, and would need to be addressed through duty-cycle tuning or additional gateways. Under simulation, a single LoRaWAN gateway supports on the order of 1,000 SWMs at ~92% packet delivery ratio (Lalle et al., 2020), suggesting that a city-scale rollout would require gateway densification rather than any change to the blockchain layer itself. This is consistent with a simple analytical check: under the pure-ALOHA channel model for LoRaWAN (Bor et al., 2016), a single channel sustains at most ~18% utilization before collisions dominate, so serving 10,000 SWMs reporting every 10 minutes would require on the order of ten gateway-equivalents of channel capacity, the same order of magnitude as the estimate above. Comparing these numbers against related work is also tricky, since hardware configurations and test conditions vary significantly across studies.

We also measured round-trip time, block confirmation time and throughput with the multi-process setup of Section 5.1 (Table 2).

Table 2. Multi-process network measurements on one machine (mean ± SD [min–max]).

Metric	n = 6 nodes	n = 12 nodes
Round-trip time (ms), N = 100	0.20 ± 0.19 [0.10–1.93]	0.24 ± 0.25 [0.09–2.46]
Block confirmation time (ms), N = 100 blocks	3.8 ± 1.3 [1.9–13.5]	7.3 ± 3.5 [4.5–38.4]
Throughput (tx/s), N = 10	8.7 ± 4.2 [3.5–15.4]	8.7 ± 2.3 [5.3–12.0]

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Round-trip time: one JSON round trip from the driver to a node over a new TCP connection. Block confirmation time: from the end of mining at the miner until a majority (⌊n/2⌋ + 1) of acknowledgements arrive; it excludes mining. Throughput: 3 transactions per block over 10 consecutive blocks; it includes mining, and the next block starts after all n acknowledgements.

Throughput does not differ between 6 and 12 nodes, and the values are indicative: they come from one machine, 10 runs and a random mining time. All measurements in Table 2 were obtained on the same single machine used for Table 1, so they reflect socket and serialization overhead, not geographic network latency.

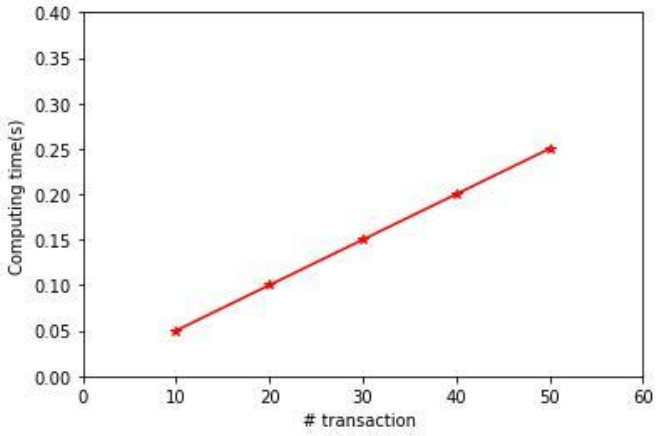

Figure - 10. Computation time.

6. Results and Discussion

Earlier blockchain and IoT work on water management has focused mostly on smart contracts and water quality sensors. This study aims to make the meter data itself trustworthy.

The architecture has three layers, with security measures wrapping around each one to protect data integrity at the core (Fig. 11). In the design, smart meters sign their readings with a private key. Fog servers validate those transactions and run Proof-of-Work. Cloud servers store the resulting Blockchain. SHA-256 hashing is used throughout. The goal is a system where any later change to recorded measurement data can be detected: no single point of control and no single copy to alter.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Whether PoW is the right consensus mechanism for this kind of infrastructure is a fair question (Tables 1(a) and 1(b) compare it with PBFT and a simplified PoA model).

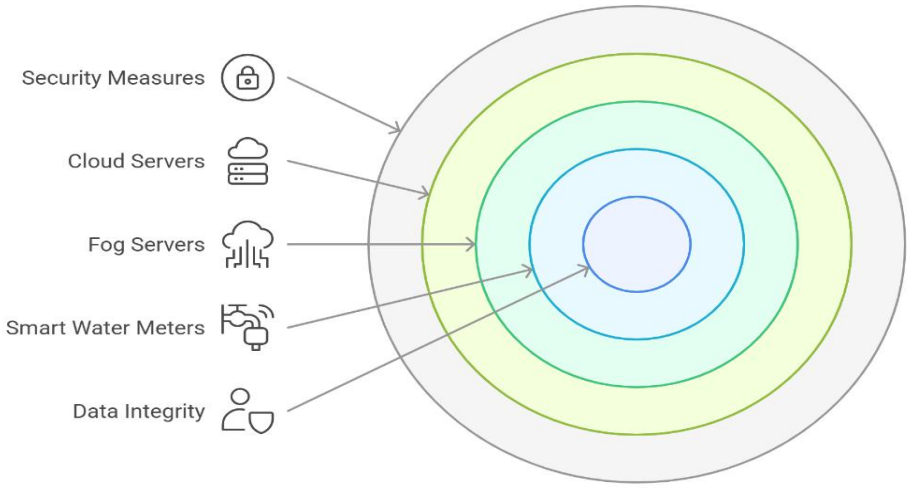

Figure - 11. Decentralized Water Measurement System.

When compared to prior studies, this research aligns with earlier findings regarding the benefits of Blockchain in water management. Xia et al. (2022) focused on smart contracts for water quality management. Similarly, while Sarantakos et al. (2024) demonstrates the feasibility of using Ethereum-based smart contracts for water distribution automation, this study extends beyond transaction automation to address fundamental security concerns. Unlike studies focused on water quality monitoring, where Blockchain secures sensor-collected environmental data (Xia et al., 2022), the present work targets the integrity of metering data itself, from the point of collection at the SWM through to the cloud layer.

Further, this study diverges from the broader review presented by Satilmisoglu et al. (2024), which identifies Blockchain scalability challenges but does not propose a specific implementation strategy for secure water metering. The practical implementation of lightweight and full nodes in the current research mitigates computational overhead, a limitation observed in Blockchain-based water management frameworks such as that proposed by Samanta and Sarkar (2023), which relies on

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Hyperledger Fabric for tamper-proof data storage but does not explicitly address metering security. In contrast to the approach adopted by Chang et al. (2021), which applies Blockchain and IoT in agricultural water distribution with a focus on supply efficiency, this study integrates advanced cryptographic mechanisms to enhance the security of urban and industrial smart water metering infrastructures.

Each study has its own strengths and limitations, and they differ in what they contribute to the field. The current study advances the security of smart water metering by implementing a peer-to-peer Blockchain model, making the integrity and traceability of collected data verifiable after recording. This is particularly relevant for smart city infrastructures facing privacy and trust issues.

Putting numbers side by side with prior work is not straightforward. Studies differ in hardware, transaction load, and what they measure. Our prototype ran 10 transactions in 0.05 seconds and 50 in 0.25 with six virtual nodes on one machine. This is computation time. Other papers rarely report the same thing under the same conditions. Until the field agrees on common benchmarks, these comparisons will stay rough.

Our results suggest that water metering can benefit from decentralized, verifiable data handling. Testing Hyperledger Fabric as an alternative, adding predictive analytics for consumption, and extending smart contracts to billing and fraud detection are the logical follow-ons. Scalability at city scale is the bigger open question.

Table 3 gives our qualitative assessment, based on Section 5. Security level reflects how easily recorded data can be altered without detection, and dependency reflects how much the system relies on a third party or a central server.

Table 3. Comparison with Existing Approaches.

Approach	Security Level	Architecture	Dependency
Cloud-Based Systems	Medium	Centralized	High
Traditional IoT Security	Medium	Semi-centralized	Medium
Proposed Blockchain Framework	**High**	**Decentralized**	**Low**

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Other work covers related ground. Pacheco et al. (2017) built a multi-layer security framework for smart water IoT. Wang et al. (2018) focused on lightweight encryption for constrained devices. Pahontu et al. (2023) moved water monitoring to a decentralized Blockchain. Tian et al. (2023) combined digital signatures, homomorphic encryption and Blockchain for smart-grid metering data. Nododile and Nyirenda (2025) paired Blockchain with IPFS to handle storage at scale. Table 4 puts these side by side with the proposed system.

Table 4. Comparison with Related Works.

Reference	Year	Security Mechanism	Architecture	Third-Party	Water Metering
Ntuli and Abu-Mahfouz (2016)	2016	Publish-subscribe + AAC	Centralized	Yes	Yes
Pacheco *et al. (2017)*	2017	Multi-layer IoT security	Centralized	Yes	Yes
Wang *et al. (2018)*	2018	Lightweight encryption	Centralized	Yes	No
Yan *et al. (2019)*	2019	Identity-based encryption	Centralized	Yes	No
Pahontu *et al. (2023)*	2023	Blockchain decentralized	Decentralized	No	Yes
Tian *et al. (2023)*	2023	Signatures + homomorphic enc. + BC	Three-tier	**Yes**	**No**
Nododile and Nyirenda (2025)	2025	BC + IPFS	Distributed	No	Yes
Proposed	**2026**	**SHA-256 + PoW + BC**	**Three-tier**	**No**	**Yes**

Limitations and future work. All measurements come from a software simulation on one machine, so we did not measure packet loss, LoRaWAN interference, gateway congestion, real network delay or energy use. Each meter keeps one fixed key pair, and the design has no key revocation. The blockchain cannot show that a meter measured

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

correctly. Algorithm 2 does not check for duplicate or old messages, so replayed measurements are not covered. Fig. 10 shows five points without error bars. PoA was tested only as a simplified model. Future work includes pseudonym rotation, key revocation, tests on Raspberry Pi and real LoRaWAN hardware, energy measurements, and a comparison of PoW, PBFT and PoA in the full prototype.

7. Conclusion

This paper has focused on the use of BC technology to make data secure and trustworthy in smart water metering system. The data collected by SWMs are the sensitive data of users and thus require a high level of security. The architecture of our proposed solution is based on fog computing where the fog servers are deployed between the IoT devices and the traditional cloud computing. The resources constraint nodes (SWMs) are called lightweight nodes whereas the nodes that take part in the BC network are called full nodes. We implemented our BC network using Flask which is a lightweight Python framework. We then built a Web application to test the Blockchain and the nodes. The mined blocks viewed through POSTMAN show readings chained by hash, which makes manipulation of recorded data detectable. We analyzed the performance of our proposed solution by considering the PoW cost and the computation time with respect to an increase in the number of transactions. As a next step beyond this software simulation, we plan to validate our solution on physical hardware using Raspberry Pi microcomputers. We will schedule the time at which data will be sent instead of sending manually transactions through the Web interface. Our system will follow the IoT concept by working fully without human intervention.

Table 5. List of Abbreviations.

Abbreviation	Meaning
SWMs	Smart Water Meters
SWG	Smart Water Grid
ICT	Information and Communication Technologies
IoT	Internet of Things
BC	Blockchain

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

P2P	Peer-to-Peer
PKC	Public Key Cryptography
ECC	Elliptic Curve Cryptography
RSA	Rivest, Shamir and Adleman
PBFT	Practical Byzantine Fault Tolerance
GWs	Gateways
EDs	End-Devices
NS	Network Server
CSS	Chirp Spread Spectrum
PoW	Proof-of-Work
PoS	Proof of Stake
PoA	Proof-of-Authority
CAP	Consistency, Availability, Partition tolerance
LoRaWAN	Long Range Wide Area Network
SF	Spreading Factor
IPFS	InterPlanetary File System
DoS / DDoS	Denial of Service / Distributed Denial of Service
NILM	Non-Intrusive Load Monitoring

Acknowledgements
The authors received no specific funding for this work.

Ethical Statement
This article does not contain any studies involving human participants or animals performed by any of the authors.

Conflict of Interest
The authors declare that they have no conflicts of interest.

AI Use Disclosure
We used AI tools to edit language and fix grammar. The research, findings, and conclusions were written by the author.

References

Alsheakh, H., Ahmad, A., Al-Haj, A., & Al-Fuqaha, A. (2024). TIoT: A trust by design architecture for secure and private Internet of Things (IoT). In Proceedings of the 21st International Multi-Conference on Systems, Signals & Devices (SSD) (pp. 391–400). IEEE. https://doi.org/10.1109/SSD61670.2024.10549150

Amaxilatis, D., Chatzigiannakis, I., Tselios, C., Tsironis, N., Niakas, N., & Papadogeorgos, S. (2020). A smart water metering deployment based on the fog computing paradigm. Applied Sciences, 10(6), 1965. https://doi.org/10.3390/app10061965

Banoth, R., Addula, S. R., Godishala, A. K., Sannapu, R., Sajja, G. S., Kumar, D., Kasula, V. K., & Tumma, C. (2026). Evolving IoT botnet threats and practical honeypot observation: A summary review and experimental study. Journal of Cybersecurity and Privacy, 6(3), 82. https://doi.org/10.3390/jcp6030082

Bhavana, G. B., Anand, R., Ramprabhakar, J., Guerrero, J. M., Thakkar, N., & Ambikapathy, A. (2025). Comparative evaluation and simulation of blockchain consensus mechanisms for secure and scalable peer-to-peer energy trading in microgrids. Scientific Reports, 15, Article 43546. https://doi.org/10.1038/s41598-025-27431-w

Bor, M. C., Roedig, U., Voigt, T., & Alonso, J. M. (2016). Do LoRa low-power wide-area networks scale? In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) (pp. 59–67). https://doi.org/10.1145/2988287.2989163

Chang, Y., Xu, J., & Ghafoor, K. Z. (2021). An IoT and blockchain approach for the smart water management in agriculture. Scalable Computing: Practice and Experience, 22(2), 105–116. https://doi.org/10.12694/scpe.v22i2.1869

Dai, H. N., Zheng, Z., & Zhang, Y. (2019). Blockchain for Internet of Things: A survey. IEEE Internet of Things Journal, 6(5), 8076–8094. https://doi.org/10.1109/JIOT.2019.2920987

De Angelis, S., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., & Sassone, V. (2018). PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. In Proceedings of the Second Italian

Conference on Cyber Security (ITASEC 2018). CEUR Workshop Proceedings.

Dogo, E. M., Salami, A. F., Nwulu, N. I., & Aigbavboa, C. O. (2019). Blockchain and Internet of Things-based technologies for intelligent water management system. In Artificial Intelligence in IoT (pp. 129–150). Springer. https://doi.org/10.1007/978-3-030-04110-6_7

Ethereum Foundation. (n.d.). Ethereum energy consumption. ethereum.org. Retrieved September 1, 2026, from https://ethereum.org/en/energy-consumption/

Hafsa, A., Sghaier, A., Machhout, M., & Malek, J. (2019). A new security approach to support the operations of ECC and AES algorithms on FPGA. In Proceedings of the 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (pp. 95–100). IEEE. https://doi.org/10.1109/STA.2019.8717272

Jose, D. V., & Vijayalakshmi, A. (2018). An overview of security in Internet of Things. Procedia Computer Science, 143, 744–748. https://doi.org/10.1016/j.procs.2018.10.439

Kimani, K., Oduol, V., & Langat, K. (2019). Cyber security challenges for IoT-based smart grid networks. International Journal of Critical Infrastructure Protection, 25, 36–49. https://doi.org/10.1016/j.ijcip.2019.01.001

Lalle, Y., Fourati, L. C., Fourati, M., & Barraca, J. P. (2019). A comparative study of LoRaWAN, SigFox, and NB-IoT for smart water grid. In Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS) (pp. 1–6). IEEE. https://doi.org/10.1109/GIIS48668.2019.9044961

Lalle, Y., Fourati, M., Fourati, L. C., & Barraca, J. P. (2020). LoRaWAN network capacity analysis for smart water grid. In Proceedings of the 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP) (pp. 1–6). IEEE. https://doi.org/10.1109/CSNDSP49049.2020.9249623

Lalle, Y., Fourati, M., Fourati, L. C., & Barraca, J. P. (2021). Routing strategies for LoRaWAN multi-hop networks: A survey and an SDN-based solution for smart water grid. IEEE Access, 9, 168624–168647. https://doi.org/10.1109/ACCESS.2021.3135080

Louw, J., Niezen, G., Ramotsoela, T., & Abu-Mahfouz, A. M. (2016). A key distribution scheme using elliptic curve cryptography in wireless sensor

networks. In Proceedings of the IEEE 14th International Conference on Industrial Informatics (INDIN) (pp. 1166–1170). IEEE. https://doi.org/10.1109/INDIN.2016.7819336

Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., & Guizani, S. (2017). Internet-of-Things-based smart cities: Recent advances and challenges. IEEE Communications Magazine, 55(9), 16–24. https://doi.org/10.1109/MCOM.2017.1600514

Mohanta, B. K., Jena, D., Panda, S. S., & Sobhanayak, S. (2019). Blockchain technology: A survey on applications and security privacy challenges. Internet of Things, 8, 100107. https://doi.org/10.1016/j.iot.2019.100107

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

Nododile, T., & Nyirenda, C. (2025). A hybrid blockchain-IPFS solution for secure and scalable data collection and storage in smart water meters. In 2025 IST-Africa Conference (IST-Africa) (pp. 1–9). IEEE. https://doi.org/10.23919/IST-Africa67297.2025.11060537

Ntuli, N., & Abu-Mahfouz, A. (2016). A simple security architecture for smart water management system. Procedia Computer Science, 83, 1164–1169. https://doi.org/10.1016/j.procs.2016.04.239

Pacheco, J., Ibarra, D., Vijay, A., & Hariri, S. (2017). IoT security framework for smart water system. In Proceedings of the IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) (pp. 1285–1292). IEEE. https://doi.org/10.1109/AICCSA.2017.85

Pahontu, B.-I., Petcu, A., Predescu, A., Arsene, D. A., & Mocanu, M. (2023). A blockchain approach for migrating a cyber-physical water monitoring solution to a decentralized architecture. Water, 15(16), 2874. https://doi.org/10.3390/w15162874

Perera, C., Qin, Y., Estrella, J. C., Reiff-Marganiec, S., & Vasilakos, A. V. (2017). Fog computing for sustainable smart cities: A survey. ACM Computing Surveys, 50(3), 1–43. https://doi.org/10.1145/3057266

Ray, S., Bhunia, S., Jin, Y., & Tehranipoor, M. (2016). Security validation in IoT space. In Proceedings of the IEEE 34th VLSI Test Symposium (VTS) (p. 1). IEEE. https://doi.org/10.1109/VTS.2016.7477259

Samanta, S., & Sarkar, A. (2023). IoT and blockchain for smart water quality management in future cities: A Hyperledger Fabric framework for smart

water quality management and distribution. Research Square. https://doi.org/10.21203/rs.3.rs-3727101/v1

Sarantakos, T., Patsakis, C., Koutsouris, D., & Papadopoulos, G. (2024). A tool to facilitate the design of smart contracts in smart water distribution networks. In 2024 IFIP Networking Conference (IFIP Networking) (pp. 708–713). IEEE. https://doi.org/10.23919/IFIPNetworking62109.2024.10619818

Satilmisoglu, T. K., Sermet, Y., Kurt, M., & Demir, I. (2024). Blockchain opportunities for water resources management: A comprehensive review. Sustainability, 16(6), 2403. https://doi.org/10.3390/su16062403

Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. Future Generation Computer Systems, 78, 964–975. https://doi.org/10.1016/j.future.2016.11.031

Tian, H., Zheng, N., & Jian, Y. (2023). Advanced metering infrastructure data aggregation scheme based on blockchain. International Journal of Advanced Computer Science and Applications (IJACSA), 14(11), 265. https://doi.org/10.14569/IJACSA.2023.0141126

Wang, C., Shen, J., Liu, Q., Ren, Y., & Li, T. (2018). A novel security scheme based on instant encrypted transmission for Internet of Things. Security and Communication Networks, 2018, Article 3680851. https://doi.org/10.1155/2018/3680851

Wang, N., Wang, P., Alipour-Fanid, A., Jiao, L., & Zeng, K. (2019). Physical-layer security of 5G wireless networks for IoT: Challenges and opportunities. IEEE Internet of Things Journal, 6(5), 8169–8181. https://doi.org/10.1109/JIOT.2019.2920809

Xia, W., Chen, X., & Song, C. (2022). A framework of blockchain technology in intelligent water management. Frontiers in Environmental Science, 10, 909606. https://doi.org/10.3389/fenvs.2022.909606

Xu, Q., Ren, P., Song, H., & Du, Q. (2016). Security enhancement for IoT communications exposed to eavesdroppers with uncertain locations. IEEE Access, 4, 2840–2853. https://doi.org/10.1109/ACCESS.2016.2575863

Yan, H., Wang, Y., Jia, C., Li, J., Xiang, Y., & Pedrycz, W. (2019). IoT-FBAC: Function-based access control scheme using identity-based encryption in IoT. Future Generation Computer Systems, 95, 344–353. https://doi.org/10.1016/j.future.2018.12.061

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.139) 2026, pp.717-752

Yu, W., & Köse, S. (2017). A lightweight masked AES implementation for securing IoT against CPA attacks. IEEE Transactions on Circuits and Systems I: Regular Papers, 64(11), 2934–2944. https://doi.org/10.1109/TCSI.2017.2712619

Zhang, Z., Qin, Z., Zhu, L., Weng, J., & Ren, K. (2016). Cost-friendly differential privacy for smart meters: Exploiting the dual roles of the noise. IEEE Transactions on Smart Grid, 8(2), 619–626. https://doi.org/10.1109/TSG.2016.2587778

نظام قياس المياه الذكي الآمن والموثوق استنادا إلى تقنية البلوك تشين

مستخلص البحث:

مع التقدم المتسارع في تقنيات المعلومات والاتصالات وتطبيقات هندسة البرمجيات، باتت دول العالم تستبدل عداداتها المائية الميكانيكية التقليدية بعدادات مياه ذكية للتخلص من قراءة العدادات اليدوية. يتكون نظام قياس المياه الذكي من آلاف الأجهزة المقيّدة الموارد المنتشرة عبر شبكة توزيع المياه لتوفير مراقبة وجمع بيانات شبه فوري. حتى الآن، اعتمدت هذه البنى التحتية على نهج مركزي قائم على الحوسبة السحابية، تُرسَل فيه جميع البيانات التي تجمعها العدادات الذكية وتُدار مركزياً. غير أن التزايد المستمر في عدد الأجهزة المتصلة جعل هذا النهج يواجه تحديات جدية تتعلق بالأمن والخصوصية وزمن الاستجابة. ونظراً لأن البيانات التي تجمعها العدادات الذكية تمثل معلومات حساسة للمستخدمين، فإنها تستوجب مستوىً عالياً من الأمان. وعلى الرغم من تطوير عدة مقاربات أمنية خلال السنوات الماضية، فإنها لا يمكن تطبيقها مباشرةً على هذه الأجهزة المقيّدة نظراً لتعقيد عمليات التشفير وفك التشفير فيها، فضلاً عن اعتماد معظمها على جهة خارجية. لمعالجة هذه الإشكاليات وضمان موثوقية البيانات في بنى تحتية لقياس المياه الذكية، نقترح في هذه الورقة نظام سلسلة كتل استشاري (مصرّح به) من ثلاث طبقات يمتد عبر عدادات إنترنت الأشياء، وعقد الضباب (fog nodes)، وخوادم السحاب. لا تشترك في كل طبقة (Proof-of-Work) وإثبات العمل SHA-256 تُستخدم دالة التجزئة في ذلك. للتحقق من النموذج، أنشأنا (Certificate Authority) أي جهة تصديق خارجية وتطبيقاً على الويب مبنياً على إطار Python شبكة سلسلة كتل باستخدام لغة Flask لمحاكاة سيناريو نشر. يجعل ربط كل قراءة بالتجزئة أي تلاعب لاحق بالبيانات المسجلة قابلاً للكشف. وعلى جهاز واحد، ازداد وقت الحوسبة من 0.05 ثانية عند 10 معاملات إلى 0.25 عند 50.

الكلمات المفتاحية: سلسلة الكتل، عداد المياه الذكي، إنترنت الأشياء، هندسة البرمجيات، الحوسبة السحابية.