Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.030528

ARTICLE

Enhancing IoT Data Security with Lightweight Blockchain and Okamoto Uchiyama Homomorphic Encryption

Mohanad A. Mohammed* **and Hala B. Abdul Wahab**

Computer Science Department, The University of Technology, Baghdad, Iraq

*Corresponding Author: Mohanad A. Mohammed. Email: Mohanad_ali1986@yahoo.com

Received: 11 April 2023 Accepted: 06 July 2023 Published: 17 November 2023

ABSTRACT

Blockchain technology has garnered significant attention from global organizations and researchers due to its potential as a solution for centralized system challenges. Concurrently, the Internet of Things (IoT) has revolutionized the Fourth Industrial Revolution by enabling interconnected devices to offer innovative services, ultimately enhancing human lives. This paper presents a new approach utilizing lightweight blockchain technology, effectively reducing the computational burden typically associated with conventional blockchain systems. By integrating this lightweight blockchain with IoT systems, substantial reductions in implementation time and computational complexity can be achieved. Moreover, the paper proposes the utilization of the Okamoto Uchiyama encryption algorithm, renowned for its homomorphic characteristics, to reinforce the privacy and security of IoT-generated data. The integration of homomorphic encryption and blockchain technology establishes a secure and decentralized platform for storing and analyzing sensitive data of the supply chain data. This platform facilitates the development of some business models and empowers decentralized applications to perform computations on encrypted data while maintaining data privacy. The results validate the robust security of the proposed system, comparable to standard blockchain implementations, leveraging the distinctive homomorphic attributes of the Okamoto Uchiyama algorithm and the lightweight blockchain paradigm.

KEYWORDS

Blockchain; IoT; integration of IoT and blockchain; consensus algorithm; Okamoto Uchiyama; homomorphic encryption; lightweight blockchain

1 Introduction

In 2008, the Blockchain concept was introduced to the technology and financial world with the birth of the Bitcoin system, which was advertised in the paper "Bitcoin: A Peer-to-Peer Electronic Cash System". The blockchain is mainly used to increase the reliability and transparency of distributed data between many users in the network. This requires the blockchain to be a distributed ledger (database) that holds the chain transactions used primarily for managing records (which are constantly increasing). It is considered an efficient way to maintain integrity and security by verifying the validity of a node to add to the chain and finding major votes of participating nodes (users) to agree/disagree on the eligibility of a user to add a block to the chain (the ledger) [1].

The Internet of Things is considered the natural growth of the Internet that connects virtual or physical things in the surrounding environment. Authors estimate that IoT devices number in the billions, and by using cheap types of sensors, users can profit from the huge advantage of using IoT devices to improve their lifestyle and their quality [2].

Homomorphic encryption is a type of encryption that allows mathematical operations to be performed on encrypted data without the need to decrypt it first. This means that data can be encrypted and then manipulated while still in its encrypted form without any loss of security. This can be useful in a variety of different applications, such as allowing data to be shared between multiple parties without the need to reveal the raw data to any of them. Homomorphic encryption is a relatively new area of research in cryptography, and there are still many open questions and challenges in this field [3,4].

Many researchers focus on using blockchain and IoT and declare the main characteristics of each technology and their advantages and disadvantages. In [5], authors addressed privacy concerns in blockchain-based IoT systems by integrating homomorphic encryption for secure and decentralized data. They compare recent technologies and discuss research challenges and future directions. In [6], the authors presented intersections between IoT and distributed ledgers and clarify that using a centralized system means the centralized server is vulnerable to certain attacks and single points of failure. Merging blockchain with IoT allows for a reliable system. In another work [7,8], blockchain was applied within the supply chain and management, and the authors declare the main characteristics of blockchain technology and limitations related to this deployment, providing some future works for authors to consider. In [9], the authors proposed an IoT service system integrated with a novel blockchain called Beekeeper. The system uses homomorphic encryption to ensure data privacy preservation. The server can operate on user data using encryption, making sure that no one can learn anything about the data within the data preservation process. In [10], the authors proposed using homomorphic and blockchain consortiums in the smart home system to preserve data privacy. The blockchain is mainly used as a verification service through verification nodes to verify transactions and nodes within the chain. The Paillier homomorphic encryption algorithm is used for encryption. In [11], a system was proposed to protect the data within the patient health system and meet its need to protect the patient data over the internet. A mechanism to use a keyword to reach user data within the supply chain system is provided. In [12], the authors combined the advantages of using edge computing and blockchain technology and built a system based on blockchain edge technology. The Paillier cryptosystem is used for data protection. The execution side encrypts the data, and decryption is done within the edge nodes when the data is received. In [13], the authors discussed the technical aspects of integrating blockchain technology with the Internet of Things and how IoT can be applied within a decentralized environment to provide valuable security features built into the blockchain.

The authors in these studies provided cover various aspects of integrating IoT, blockchain, and homomorphic encryption in different domains like supply chain management, smart home, and healthcare. They emphasize the risks associated with privacy leakage in centralized IoT systems and propose solutions for improved privacy and decentralization. Moreover, they explore the advantages of integrating blockchain into the IoT architecture, such as traceability, supply chain decentralization, and transparency.

One important gap addressed in the previous studies is the computation costs involved in implementing blockchain within IoT systems. And express the need for lightweight blockchain solutions that reduce computational overhead and enhance efficiency in IoT environments. This is significant

because traditional blockchain implementations can pose limitations on the scalability and real-time processing capabilities of IoT devices.

This paper proposes a lightweight blockchain system based on secret sharing and integrated with homomorphic encryption for securing the IoT data (e.g., supply chain data), which reduces the time and computational requirements of such a system, improves privacy security, and provides immutability using the blockchain.

2 Main Description of Blockchain Technology

Blockchain technology is a special type of data structure that uses a hash function with public-key encryption (asymmetric encryption) to protect against forgery and tampering. Blocks within the blockchain contain transactions between users and nodes that are timely ordered in cryptocurrencies such as Bitcoin or Ethereum.

Some authors refer to blockchain as a decentralized distributed ledger that allows communication between users and transferring of digital assets without the interference of a third party. The unique characteristics of blockchain, such as traceability, tamper-proofing, and decentralization, allow it to work as a protocol for the distributed network to build a trusting relationship between different participants, even though they do not know each other. Nowadays, the technology of blockchain is not exclusive to the financial field (generation and management of cryptocurrency), but it has recently entered many other areas, such as supply chains, smart cities, education, and the Internet of Things (IoT). Nevertheless, whatever the scenario in the blockchain will be used or applied, the current passion of researchers is how to design and implement efficient and secure blockchain systems that serve different needs [13].

In the traditional chain blocks, the adjacent blocks within the blockchain are connected via hash codes, which provide integrity to the data inside each block and can work as a unique ID for each block. Fig. 1 provides an intuitive representation of the blockchain and defines the structure of the chain and blocks [14].

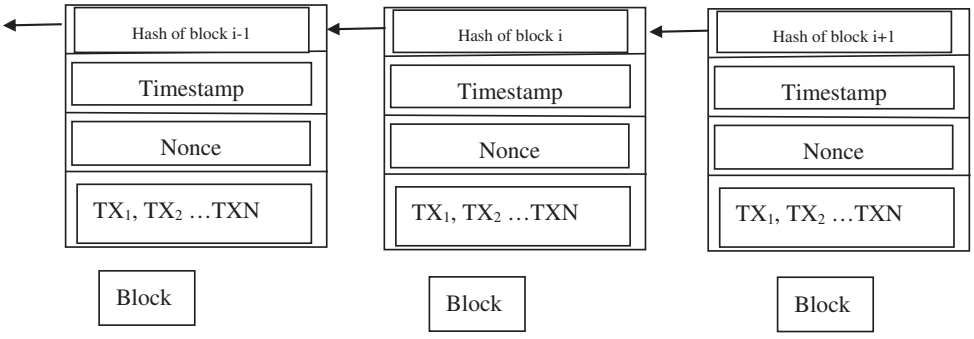

Figure 1: Blockchain [14]

2.1 Consensus Algorithm

The concept of the Byzantine general's problem inspired the development of the consensus algorithm, which ensures the agreement of honest generals in a decision-making process to attack or retreat. By applying this technique, consensus can be ensured even if a traitor is present [15]. In the context of blockchain, the consensus algorithm is utilized to maintain data consistency in the

distributed network, particularly in cases of node failures. Crash fault nodes and Byzantine fault nodes are two types of node failures that can occur. Crash fault nodes are relatively easy to handle, while Byzantine fault nodes pose a significant challenge. To achieve consensus, a consensus algorithm must adhere to certain standards, such as consistency and liveness, and meet specific requirements for scalability, throughput, and cost reduction. Various consensus algorithms exist, including proof of work, proof of stake, and practical Byzantine fault tolerance [16]. A new consensus algorithm based on secret sharing and zero-knowledge technology is proposed in [17] for blockchain.

The most known consensus algorithm can be:

1. Proof of Work (PoW):

This consensus algorithm is considered the first used in blockchain. The algorithm provides a complex mathematical puzzle and the node participating tries to solve this puzzle and mine the result. The node reaches the correct guess verifies the added block, and earns the reward (usually cryptocurrency balance). The two most famous cryptocurrencies use PoW (Bitcoin and Ethereum) [16].

2. Proof of Stake (PoS):

It is the second well-known consensus algorithm used within cryptocurrency for transactions verifying and new blocks generating within the blockchain, it assigns shares to specific nodes and this share can be increased and decreased according to the correct validations done by the node. The node that provides the right vote will raise its share against wrong votes nodes.

3. Delegated Proof of Stake (DPoS) [16]:

This method considers a derivative version of the proof of stake using the same concept based on correct/wrong votes that provide a penalty and reward to the correct vote. The main concept of the DpoS is to use nodes to elect a group of nodes that is delegated to validate the new block to be added to the blockchain [16].

4. Practical Byzantine Fault Tolerance (pBFT):

Consider the oldest consensus algorithm which mainly works with synchronous systems and provides low overhead time. Consider a wide-use consensus algorithm in the blockchain world that mainly work on majority voting and have a bottleneck when the number of nodes is huge in its performance and speed is affected [16].

5. Proof of Capacity (PoC):

The process of validating a transaction is done by allowing the devices that do mining within the network to use their free hard disk space to decide which node owns the right to validate. This allows using of this space disk node and computation power within chains and cryptocurrencies [16].

2.2 Blockchain Operations

The blockchain can be thought of as a distributed ledger where each node connected to the network can have a share in correcting it. This decentralized ledger contains different transactions added by nodes without the need for third-party interference. If a node would like to store a transaction in the distributed ledger, a global vote between nodes is applied, and the majority of participating users within the blockchain have to agree to this additional set of transactions. These transactions are then grouped and form a block that is added to the ledger and connected to the chain of blocks. To connect this block to the chain, a timestamp (optional) and hash function are calculated for the current and previous blocks. This hash can be used for the validation, integrity, as well as non-repudiation of block

data of the blockchain. Every time the chain is updated, all the nodes connected to the network must be informed to update their blockchain [13,14].

3 Internet of Things (IoT)

The Internet of Things (IoT) refers to the embedding of billions of devices, ranging in size from small ones like Radio Frequency Identification (RFID) tags, sensors, mobile phones, drones, and those with low computation capabilities, to large devices like self-driving cars, industrial control systems, and smart devices with vast resources, computing power, and high memory and storage. All these diverse devices are interconnected through the internet to exchange information. Usually, IoT systems do not require human interaction, and they are based on intelligent apps that make human life easier.

IoT devices within the network interact with each other through the internet, using a minimal amount of computational resources such as bandwidth and battery life [18]. The general architecture of the IoT network is centralized, which may cause many fault tolerance problems if any of the clients/server fails. Furthermore, any malicious device can access the network using device spoofing or false authentication.

Currently, the security tools and applications used in the IoT environment are vulnerable to many privacy and security issues and are not robust enough. This is due to the many constraints of power and computation ability that IoT devices have, which make it difficult to implement a robust security mechanism for protection [18].

3.1 IoT Challenges

Since the internet is available to people all around the world, many researchers have been working on providing technological solutions that are based on Internet connectivity. One of the new trends in the 21st century is the Internet of Things (IoT), where numerous devices are connected to the Internet. According to experts, in the future, all things will be connected via the internet and based on IoT.

IoT development faces two main challenges:

- Low security, as the devices have the low computational power to provide full security.
- High costs, including operational and maintenance costs.

Merging blockchain with IoT can provide security to diverse devices and create an intelligent architecture.

Centralized systems have certain vulnerabilities that can expose them to potential risks and attacks. The vulnerabilities may have:

- Single Point of Failure: In centralized systems, all control and authority are concentrated in a single point. This makes the system highly dependent on that central point. If it fails or gets compromised, the entire system becomes inaccessible or vulnerable to exploitation.
- Data Breaches: Centralized systems store large volumes of data in a single location, making them attractive targets for hackers. A successful breach can result in unauthorized access to sensitive information, leading to data theft or manipulation.
- Denial of Service (DoS) Attacks: Attackers can overwhelm a centralized system by inundating it with excessive traffic or requests. As a result, the system becomes unable to serve legitimate users, causing disruption in normal operations and potentially leading to financial losses or damage to reputation.

- Lack of Transparency: Centralized systems often lack transparency, making it challenging for users to verify the accuracy and integrity of the data stored or processed by the system. This lack of transparency can erode trust and confidence in the system, especially when it comes to sensitive information.
- Limited Scalability: Centralized systems may encounter difficulties in scaling up to accommodate growing data volumes or increasing user demands. The process of expanding such systems can be complex and expensive, posing challenges to their adaptability and efficiency.
- Insider Threats: Centralized systems are vulnerable to insider threats, where individuals with authorized access misuse their privileges for personal gain or engage in malicious activities. This can include unauthorized data access, theft, unauthorized modifications, or even intentional sabotage [18].

3.2 Integration of Blockchain and IoT

Since IoT is subject to different kinds of constraints, such as topology and resource limitations, traditional security mechanisms cannot be fully applied to its architecture. The security of IoT is typically represented through time series and data encryption.

According to recent studies, blockchain technology will increasingly be combined with concepts like big data, mobile internet, IoT, cloud computing, fog computing, and many other recent technologies [19]. However, blockchain itself can face many security challenges during its usual maintenance and operations, such as key management (generation and distribution), access control, and countering Distributed Denial-of-Service (DDoS) attacks.

If an attacker within the network wants to steal cryptocurrency or spend a specific coin multiple times, the possible solution is to write every block of the blockchain in a public ledger (called a long-term blockchain ledger). Usually, blockchain technology is integrated with the following technologies: peer-to-peer networks, distributed ledger, and smart contracts. This integration provides a new generation of secure, reliable, fair, efficient, and intelligent data processing with the highest priority to the security concept. Blockchain technology became popular due to the guarantee of secure transactions [20]. Additionally, blockchain can be employed in the metaverse world to improve security and prove ownership [21] and applied to NB-IoT [22].

Blockchain and medical fields can be integrated in different ways, such as employing the proposed system in [23] and saving the related data within the blockchain.

Another suggestion is to improve the public-private key generation of blockchain using Chebyshev polynomial [24] and an improved version of NTRU [25].

4 Homomorphic Algorithms [26]

Homomorphic encryption is a technique that enables secure computation on special data that is encrypted. This means that processed data can be computed securely without the need to decrypt it first. Homomorphic encryption has the potential to enable new applications and use cases that were not previously possible, such as secure data sharing and cloud computing. It is a rapidly developing field of research in cryptography and has the potential to greatly improve the security and privacy of data in various apps [27].

One of the key advantages of homomorphic encryption is that it enables data to be shared and processed securely without revealing the raw data to any of the parties involved. This can be

particularly useful in scenarios where multiple parties need to collaborate on a project or share data but where there are security concerns that prevent the raw data from being shared.

Another advantage of homomorphic encryption is that it allows for a high degree of privacy and security. Because the data remains encrypted throughout the entire process, even the parties performing the mathematical operations on the encrypted data cannot see the raw data. This makes it much harder for any unauthorized party to access the data, even if they can intercept the encrypted data as it is being transmitted.

Overall, homomorphic encryption is an exciting area of research in cryptography that has the potential to revolutionize the way that data is shared and processed. While there are still many challenges and open questions in this field, the potential applications of homomorphic encryption are numerous and could have a significant impact on the way that data is handled in the future.

Homomorphic encryption and blockchain technology are two separate fields, but they have the potential to be integrated in various ways. For example, homomorphic encryption could be used to enable the secure sharing of data on a blockchain platform. This would allow data to be encrypted and then stored on the blockchain, where it could be processed and manipulated without the need to decrypt it first. This could be useful for protecting sensitive data on the blockchain and enabling secure collaboration between multiple parties on blockchain-based projects.

Another potential use for the integration of homomorphic encryption and blockchain technology is in the area of secure transactions. Homomorphic encryption could be used to encrypt financial data such as credit card numbers and then perform mathematical operations on the encrypted data to verify the authenticity of transactions without revealing the raw data. This could be integrated with blockchain technology to enable secure, transparent, and auditable financial transactions on a decentralized platform.

Overall, the integration of homomorphic encryption and blockchain technology has the potential to enable a wide range of new applications and use cases. As these two fields continue to evolve and develop, we are likely to see more and more examples of how they can be integrated to provide enhanced security and functionality [28].

4.1 Okamoto Uchiyama Cryptosystem [29]

The Okamoto–Uchiyama cryptosystem is a public key encryption algorithm that possesses homomorphic characteristics that work with (Z/nZ)∗.

The Okamoto–Uchiyama algorithm accepts integer input since all the characters or groups of characters are represented by integer values used within the secret communications between parties.

The key generation algorithm steps are explained in Algorithm 1, where two values are selected randomly—P and Q—and then N values are calculated by multiplying the square of P with Q; then, the g value is generated according to the condition where $g \in \{2 \ldots . N-1\}$ is chosen such that $gp^{-1} \not\equiv 1 \bmod p^2$, and finally, the value of H is calculated.

Algorithm 1: Okamoto–Uchiyama key generation

Input: randomly selected values
Output: public and private pairs

Step 1: generate two large primes P and Q
Step 2: compute n as follows:

(Continued)

Algorithm 1 (continued)

$$N = P^2 * Q$$

where N represents the modulus, P, Q, is a prime numbers
Step 3: choose $g \in \{2 \ldots . N-1\}$ such that $gp^{-1} \not\equiv 1 \bmod p^2$
Step 4: compute H as follows:

$$H = g^N \bmod N$$

Step 5: public keys (n, g, h), private keys (p, q)

The encryption process of Okamoto–Uchiyama is described in Algorithm 2.

Algorithm 2: Okamoto–Uchiyama encryption

Input: plaintext, keys
Output: cipher text

Step 1: randomly select a value of r that is between 1 and n−1
Step 2: compute c as follows:

$$C = g^m h^r \bmod N$$

M = message to be encrypted

The decryption process of Okamoto–Uchiyama is carried out by calculating the values of A and B and the inverse of B with reference to module P and the value of the original message, found by multiplying these values Modulo p, as shown in Algorithm 3.

Algorithm 3: Okamoto–Uchiyama decryption

Input: cipher text, keys
Output: plaintext

Step 1: compute the value of A as follows:

$$A = \frac{(c^{p-1} mod\, p^2) - 1}{p}$$

where c is the cipher text while p is the prime number
Step 2: compute the value of B as follows:

$$B = \frac{(g^{p-1} mod\, p^2) - 1}{p}$$

where $g \in \{2 \ldots . N-1\}$ such that $gp^{-1} \not\equiv \bmod p^2$
Step 3: compute the inverse of B Modulo P as follows:

$$B' = B^{-1} \bmod P$$

Step 4: compute m as follows:

$$M = AB' \bmod P$$

5 Proposed System

As current blockchain technology requires heavy computations and a huge amount of resources, it is not suitable for use with IoT systems. This has raised the need to design a lightweight blockchain that requires less computation power and resources and can be used within the IoT environment.

Blockchain and homomorphic encryption provide strong security measures for IoT applications. With blockchain, you can ensure the integrity of data, authenticate devices, and securely update firmware at the application layer and business layers. On the other hand, homomorphic encryption allows for analytics while preserving privacy, secure sharing of data, confidential machine learning, and secure offloading of computations at the business layer. By leveraging these technologies together, protecting sensitive information, preventing unauthorized tampering, and maintaining privacy, ultimately enhancing the security of your IoT systems. However, it is important to carefully consider the specific requirements and limitations to achieve the best results.

The proposed system replaces the existing proof of work (POW) algorithm using proof of secret sharing and secures the data by applying the Okamoto Uchiyama encryption algorithm as an encryption method. The contributions of the proposed system are:

Since this system is mainly for private blockchains, it provides equal chances for the participating nodes to add blocks to the chain, and there is no need for complex mining mathematical operations.

The existing PoW consensus algorithm is replaced with the proof of secret shares, where each node's share is used for authentication of the node to the blockchain system. After authentication is done, these shares are used to reconstruct the secret, allowing the node that would like to update the ledger to add its node.

Original distributed ledger technology is used within a text file, and the Merkle tree is replaced with the comparison of text files for authentication and verification on nodes.

Using homomorphic encryption, the Okamoto Uchiyama algorithm encrypts only the sensitive data, which reduces the time and power required to encrypt every data within the chain.

The proposed system mainly consists of four main phases, as shown in Fig. 2.

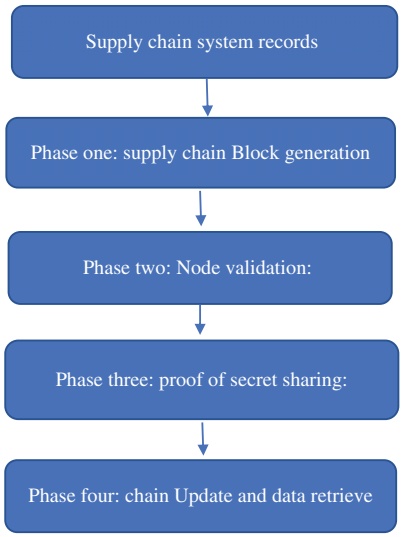

Figure 2: Proposed system architecture

Pre-phase: encrypt system data:

This phase is selective to the management of nodes for data that need to be encrypted

Generate n value

Select the g values

compute H as follows:

$$H = g^N \bmod N$$

where $g \in \{2 \ldots . N-1\}$ such that $gp^{-1} \not\equiv 1 \bmod p^2$, N is the modulus.

Encrypt the data which is now ready to add to the chain. This is done via Okamoto Uchiyama.

1. Phase one: Supply chain Block generation

The management node is responsible for generating the secret related to each node and synchronizing the network nodes with each other to apply proof of secret sharing (PoSS).

The management node chooses a secret to generate shares for each node within the network, which can be changed periodically as needed.

The management node chooses the number of shares, W, which is equal to the number of nodes. If a new node is added, the shares need to be recalculated based on the new node number.

W secret shares, Z, are generated using Shamir's secret sharing.

This phase includes the process where the data is collected and turns into a block that is ready to be validated and added to the node.

And it consists of many internal steps to accommodate the preparation of the blocks within the system:

1. Data generating: The supply chain nodes generate data to initiate the block that will be added to the blockchain each data related to a node from the supply chain nodes consider a transaction. Every node participating in the data generation must have a valid encrypted secret share for the authentication process as well as an identical ledger to all network nodes.
2. Transaction pool: if there is more than one adding node, the data generated (transactions) are collected in a special pool and formulate the block. Usually, for every Supply chain center after the completion of the data-gathering process time, one unique block for the results forms the final block.
3. Block-adding node role: one of the Supply chain center's nodes needs to take the role of the block-adding node to request to add a new block to the blockchain distributed ledger.

2. Phase two: Node validation

The validation and authentication of adding nodes as well as the nodes participating in the consensus process, is done within this phase. This phase consists of many internal steps to accommodate the validation of the nodes within the system:

1. Network nodes authentication: each node, including the adding block node need to be authenticated before starting the consensus steps. If one of the authenticated nodes fails the authentication process then the whole process is stopped and further pre-defined steps are applied to reach this node and solve this issue. The authentication is done by comparing the hashes of all nodes and the distributed ledger of all nodes within the private peer-to-peer network. And sharing the encrypted secret shares with the management node.

2. Consensus nodes selection: from all the authenticated nodes within the peer-to-peer network N-1 nodes are selected randomly to participate in the consensus decision of reconstructing the secret within phase three, where the N number represents the threshold of Shamir secret sharing algorithm.
3. Authenticated node secrets sharing: each node from the selected nodes and the adding block node need to share its encrypted share of secret with the consensus process. The value of the correct share will be calculated in phase 3.

3. Phase three: Proof of secret sharing

This phase consists of many internal steps to accommodate the consensus and addition of the block to the chain:

1. Generated clear secret shares: since the secret shares are encrypted using the Okamoto Uchiyama homomorphic encryption algorithm, then these secrets need to be decrypted before they can participate in the reconstruction of the secret. Each node secret share needs to be decrypted on nodes applying the consensus to provide the required security to the PoHSS algorithm where even if an attacker were able to attack the system nodes, including the N nodes that participated in the consensus, he will be unable to attack all nodes Okamoto Uchiyama algorithm to generate the shares and reconstruct the secret.
2. Reconstruction of the secret: this step includes collecting all the decrypted shares and reconstructing the secret using the secret sharing algorithm used within the system (within this system Shamir's secret sharing was used). If the secret is reconstructed correctly, then all the participating nodes are validated included the adding block node and all nodes consensus. The block connected to the blockchain if it is not reconstructed correctly, then the operation is stopped and more investigation is needed to detect the network node that shares wrong information to correct or isolate it to avoid malicious behavior or special correction to the encryption/decryption of the Okamoto Uchiyama algorithm.
3. Block addition and ledger update: after the consensus is met in the previous step for the secret generated from all nodes that are selected randomly and authenticated in phase 2, then, the block is added to the blockchain and the ledger is updated and all nodes within the peer-to-peer network ledgers need to update.

4. Phase four: Chain update and data retrieve

To incorporate the encrypted data into the block, the system include it as part of the block's content. This ensures that the data remains secure and protected within the blockchain.

Next, generate the previous hash and current hash. The previous hash represents the hash value of the preceding block in the chain, while the current hash is the hash value of the current block. These hash values serve as unique identifiers and play a crucial role in maintaining the integrity and immutability of the blockchain.

Finally, update the ledger of the blockchain by adding the information from the new block. This ensures that all transactions and data within the blockchain are recorded and organized in a transparent and chronological manner.

Algorithm 4: Lightweight blockchain with homomorphic for IoT

Input: generated keys, shares, secret, encrypted IoT data
Output: updated ledger

(Continued)

Algorithm 4 (continued)

Step 1: the node aiming to update the chain and ledger receives the distributed ledger's latest hashes and every node secret share
Step 2: check the compatibility and consistency of nodes by comparing the hashes and in case of mismatch, the operation aborted
Step 3: in case of matching of all hashes, then the nodes validation is done and authentication is required within phase two
Step 4: management cell receives the nodes share and recalculate the secret by Choosing random U nodes to reconstruct the secret of the system
Step 5: in case of mismatch, the operation aborted
Step 6: in case of matching, then the node has permission to add a new block to the blockchain and authentication is done via proof of secret shares
Step 7: generate n value
Step 8: select the g values
Step 9: compute H as follows:

$$H = g^N \bmod N$$

Step 10: encrypt the data, which now ready to add to the chain
Step 11: add the encrypted data from step10 to the chain using the standard blockchain mechanism
Step 12: generate the previous hash of accumulative hash values and the current hash
Step 13: update the ledger of the blockchain by adding the new node
Step 14: end

6 Implementation

To demonstrate the practical use of the proposed system, an implementation of the system and a description of the dataset used in this work are presented. The system uses the data described in the next subsection, which is encrypted using the Okamoto-Uchiyama algorithm and saved within the blockchain. To meet the requirements of IoT devices, a lightweight version of the blockchain is provided and tested using the algorithm's phases described in Algorithm 4.

6.1 Dataset

An international network of technology-driven grocery stores heavily depends on cutting-edge technologies like the Internet of Things (IoT) to gain a competitive advantage over other grocery stores. Foods like groceries are quite perishable. Understocking runs the danger of losing consumers, while overstocking costs money in the form of extra storage and waste. Businesses are interested in learning more efficient product stocking techniques. This is a complex business challenge that requires analysis of data to provide recommendations for how to fix the supply chain problem. It is important that this data is secure and cannot be leaked to competitors or other companies. Therefore, only selective references to important data should be saved, and the number of supplies needs to be securely saved in a distributed manner for other branches to use as needed.

6.2 Proposed System Implementation

An implementation of the proposed system is presented for the original blockchain with homomorphic encryption, as well as for the lightweight version with homomorphic encryption, to demonstrate the differences in the main factors that affect the blockchain system, such as computation power

and speed. Table 1 shows the implementation of the system with lightweight blockchain technology, while Table 2 shows the comparison between the proposed system (lightweight blockchain) and the original system (standard blockchain) with the homomorphic encryption Okamoto Uchiyama algorithm. Fig. 3 shows the difference in time and computation power between the proposed and original blockchains.

Table 1: Proposed the system implementation of a lightweight blockchain

IoT data	Encrypted data	Hash data	Average time
20	1702319485063250896424 7236444	e3b0c44298fc1c149afbf4c8996fb92427 ae41e4649b934ca495991b7852b855	1.00293669999 99993
5	3939885914230121426014 0334796	e3b0c44298fc1c149afbf4c8996fb92427 ae41e4649b934ca495991b7852b855	1.00617030000 00009
3	7934782660694479498748 31716	e3b0c44298fc1c149afbf4c8996fb92427 ae41e4649b934ca495991b7852b855	0.99910939999 99983
31	1224403029069905076976 329963	e3b0c44298fc1c149afbf4c8996fb92427 ae41e4649b934ca495991b7852b855	0.99396949999 99986
20	1702319485063250896424 7236444	e3b0c44298fc1c149afbf4c8996fb92427 ae41e4649b934ca495991b7852b855	1.00293669999 99993

Table 2: Comparison of the proposed system and the original system

Metric	Standard blockchain	Lightweight blockchain
Transactions per second (TPS)	7–37	262
Transaction size	0.02 KB	0.02 KB
Block size	0.3310546875 kilo-byte	0.3310546875 kilo-byte
Block generation time	1.6 s	0.013 s
Block verification time	6.2033 s	0.975 s
Final time (Block generation + verification time)	8.10 s	1.01 s
Average CPU usage	13.2	2.8
Average CPU user time	5287.89	5257.89
Average CPU system time	3788.6	3786.6
Average idle time	1,987	1,980
Average interrupt time	2482.5	2473.40625
Average RAM %	47.8/100	50.8/100

In reference to Tables 1, 2 and Fig. 3, the following characteristics of the results obtained:

- IoT Data: The amount of IoT data processed in each test case, ranging from 3 to 31 units.
- Encrypted Data: The encrypted representation of the IoT data processed during the tests.
- Hash Data: The resulting hash value is computed from the encrypted data using a standard hashing algorithm (SHA-256).

- Average Time: The average time taken to process the given amount of IoT data, measured in seconds.

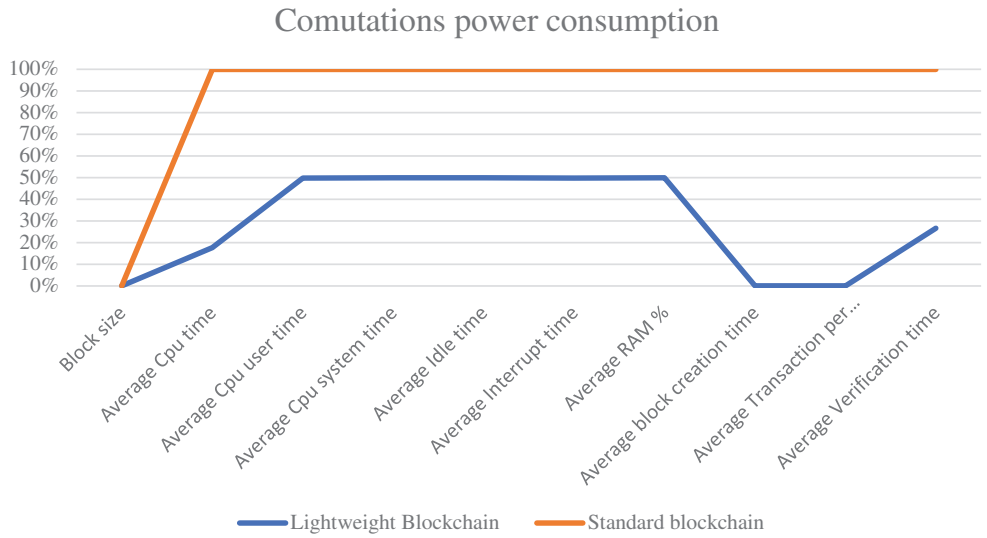

Figure 3: Difference in time and computation power between the proposed and original blockchain

The evaluation metrics compare the performance of the lightweight blockchain system to a standard blockchain system:

- Transactions per second (TPS): The first system achieved a range of 7–37 transactions per second, whereas the second system recorded a significantly higher value of 262 transactions per second. This indicates that the second system processed a much larger volume of transactions within the same time frame, demonstrating its ability to handle higher transaction loads.
- Transaction size: Both systems had a transaction size of 0.02 KB, indicating that the size of the individual transactions was the same for both systems.
- Block size: The block size for both systems was 0.3310546875 kilobytes, demonstrating that they allocated a similar amount of storage space for each block.
- Block generation time: The first system had a block generation time of 1.6 s, whereas the second system boasted an impressively low block generation time of only 0.013 s. This signifies that the second system was capable of generating blocks at a much faster rate.
- Block verification time: The first system required 6.2033 s for block verification, while the second system achieved much faster verification times at 0.975 s. This indicates that the second system's verification process was significantly more efficient.
- Final time (Block generation + verification time): When combining the block generation and verification times, the first system had a final time of 8.10 s, whereas the second system achieved a considerably lower final time of 1.01 s. This shows that the second system was able to generate and verify blocks much more quickly.
- Average CPU usage: The first system had an average CPU usage of 13.2, while the second system exhibited a lower average CPU usage of 2.8. This indicates that the second system required fewer CPU resources to perform its operations.

- Average CPU user time: Both systems had similar average CPU user times, with the first system recording 5287.89 and the second system recording 5257.89. This suggests that the user-level CPU usage was comparable for both systems.
- Average CPU system time: The average CPU system times for both systems were nearly identical, with the first system at 3788.6 and the second system at 3786.6. This indicates that the system-level CPU usage was similar for both systems.
- Average idle time: The average idle time for the first system was 1987, whereas the second system had a slightly lower average idle time of 1980. This suggests that the second system had slightly less idle time.
- Average interrupt time: The first system recorded an average interrupt time of 2482.5, while the second system achieved a slightly lower average interrupt time of 2473.40625. This implies that the second system experienced slightly fewer interruptions.
- Average RAM %: The RAM usage for both systems was relatively close, with the first system utilizing 47.8% of available RAM and the second system using 50.8%. This indicates that both systems made efficient use of system memory.

7 Conclusions and Discussions

The proposed system, which uses Okamoto Uchiyama and lightweight blockchain technology, offers many benefits, including:

Reliability: The blockchain ensures data reliability by using various hash algorithms. If any modifications or tampering occur within the block's hash codes, the system detects it, and the new block cannot be accepted.

Authenticity: Proof of secret sharing ensures the authenticity of nodes connected to the network and determines whether or not to add their blocks to the chain.

Decentralization: Integrating blockchain and IoT devices provide decentralization, meaning data is still accessible in emergencies like attacks or system failure, and there's no need for a third party to manage the system.

Availability: The IoT system consists of many devices connected, so even if one device fails, data remains available in the blockchain.

Security and reliability: The IoT data is stored in a distributed ledger and goes through the PoSS algorithm, increasing data reliability. Additionally, the homomorphic encryption method Okamoto Uchiyama is used to encrypt data within these ledgers, enhancing data security.

Using lightweight blockchain technology will enable the system to apply decentralized management and provide security to IoT devices within a trustless environment. Integration of blockchain, Okamoto Uchiyama homomorphic encryption, and IoT provides the advantage of using blockchain technology to solve security issues and homomorphic encryption to solve privacy issues related to the IoT and provides a secure channel for shared information between heterogeneous IoT devices. This makes the IoT system more resilient against different types of attacks and provides features such as data integrity, authenticity, immutability, availability, and reliability to IoT devices.

The proposed system improves the time required for each operation and reduces the general need for computational power.

8 Recommendations and Future Work

- Embedding with AI: Explore the integration of artificial intelligence (AI) techniques to analyze data stored in the blockchain, enabling advanced analytics, predictive modeling, and anomaly detection.
- Application in the Metaverse: Investigate the application of the proposed system in the context of the metaverse, creating a secure and decentralized environment for storing and exchanging virtual assets, ensuring data integrity, and enabling trusted interactions between virtual entities.
- Privacy-Preserving Enhancements: Further explore and enhance the privacy-preserving capabilities of the Okamoto Uchiyama encryption algorithm by integrating techniques such as differential privacy or secure multi-party computation, ensuring stronger privacy guarantees for IoT-generated data.
- Pushing the Boundaries: Embrace these recommendations to unlock new opportunities for integrating AI, applying the technology in the metaverse, and enhancing privacy preservation.
- Evolving Blockchain Solutions: Contribute to the evolution and development of blockchain-based solutions for the Internet of Things, opening doors to innovative applications and use cases across various domains.

Acknowledgement: None.

Funding Statement: This research received no external funding.

Author Contributions: The authors contributed equally to this work.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article and real election data used are available in this link https://www.kaggle.com/datasets/abhinayasaravanan/grocery-supply-chain-isuue.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Nofer, M., Gomber, P., Hinz, O., Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering, 59(3),* 183–187.
2. Tasatanattakool, P., Techapanupreeda, C. (2018). Blockchain: Challenges and applications. *2018 International Conference on Information Networking (ICOIN)*, pp. 473–475. Chiang Mai, Thailand, IEEE.
3. Zivic, N., Ruland, C., Ur-Rehman, O. (2019). Addressing Byzantine fault tolerance in blockchain technology. *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, pp. 1–5. Bahrain, IEEE.
4. Paulavičius, R., Grigaitis, S., Filatovas, E. (2021). An overview and current status of blockchain simulators. *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–3. Sydney, Australia, IEEE.
5. Shrestha, R., Kim, S. (2019). Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities. In: *Advances in computers*, vol. 115, pp. 293–331. USA, Elsevier.
6. Atlam, H. F., Wills, G. B. (2019). Intersections between IoT and distributed ledger. In: *Advances in computers*, vol. 115, pp. 73–113. USA, Elsevier.

7. Alalwi, B., Mazzuchi, T., Hamdan, A., Mubarak, M. A. (2021). Blockchain technology implications on supply chain management: A review of the literature. In: *Applications of artificial intelligence in business, education and supply chain*, pp. 23–38. UK.
8. McBee, M. P., Wilcox, C. (2020). Blockchain technology: Principles and applications in medical imaging. *Journal of Digital Imaging, 33(3),* 726–734.
9. Zhou, L., Wang, L., Sun, Y., Lv, P. (2018). Beekeeper: A blockchain-based IoT system with secure storage and homomorphic computation. *IEEE Access, 6,* 43472–43488.
10. She, W. E. I., Gu, Z. H., Lyu, X. K., Liu, Q. I., Tian, Z. et al. (2019). Homomorphic consortium blockchain for smart home system sensitive data privacy-preserving. *IEEE Access, 7,* 62058–62070.
11. Ali, A., Pasha, M. F., Ali, J., Fang, O. H., Masud, M. et al. (2022). Deep learning based homomorphic secure search-able encryption for keyword search in blockchain supply chain system: A novel approach to cryptography. *Sensors, 22(2),* 528.
12. Yan, X., Wu, Q., Sun, Y. (2020). A homomorphic encryption and privacy protection method based on blockchain and edge computing. *Wireless Communications and Mobile Computing, 2020(3),* 1–9.
13. Ghani, R. F., Salman, A. A., Khudhair, A. B., Aljobouri, L. (2022). Blockchain-based student certificate management and system sharing using hyperledger fabric platform. *Periodicals of Engineering and Natural Sciences, 10(2),* 207–218.
14. Hasan, I. M., Ghani, R. F. (2021). Blockchain for authorized access of health insurance IoT system. *IRAQI Journal of Computers, Communications, Control and Systems Engineering, 21(3),* 76–88.
15. Du, M. X., Ma, X. F., Zhang, Z., Wang, X. W., Chen, Q. J. (2017). A review on consensus algorithm of blockchain. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2567–2572. Canada, IEEE.
16. Panda, S. S., Mohanta, B. K., Satapathy, U., Jena, D., Gountia, D. et al. (2019). Study of blockchain-based decentralized consensus algorithms. *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp. 908–913. Kerala, India, IEEE.
17. Mohammed, M. A., Abdul Wahab, H. B. (2022). Proposed new blockchain consensus algorithm. *International Journal of Interactive Mobile Technologies (iJIM), 16(20),* 79–97.
18. Chen, S., Xu, H., Liu, D., Hu, B., Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with China perspective. *IEEE Internet of Things Journal, 1(4),* 349–359.
19. Panarello, A., Tapas, N., Merlino, G., Longo, F., Puliafito, A. (2018). Blockchain and IoT integration: A systematic survey. *Sensors, 18(8),* 2575.
20. Zheng, J., Dike, C., Pancari, S., Wang, Y., Giakos, G. C. et al. (2022). An in-depth review on blockchain simulators for IoT environments. *Future Internet, 14(6),* 182.
21. Jaber, T. A. (2022). Security risks of the metaverse world. *International Journal of Interactive Mobile Technologies, 16(13),* 182–204.
22. Jaber, T. A., Hussein, M. A. (2019). Study on known models of NB-IoT applications in Iraqi environments. *IOP Conference Series: Materials Science and Engineering, 518,* 052013.
23. Tutsoy, O. (2021). Pharmacological, non-pharmacological policies and mutation: An artificial intelligence based multi-dimensional policy making algorithm for controlling the casualties of the pandemic diseases. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(12),* 9477–9488.
24. Wahab, H. B. A., Jaber, T. A. (2016). Using Chebyshev polynomial and quadratic Bézier curve for secure information exchange. *Engineering and Technology Journal, 34,* 27–46.
25. Wahab, H. B. A., Jaber, T. A. (2015). Improve NTRU algorithm based on Chebyshev polynomial. *2015 World Congress on Information Technology and Computer Applications (WCITCA)*, pp. 1–5. USA, IEEE.
26. Jaber, T. A. (2022). Artificial intelligence in computer networks. *Periodicals of Engineering and Natural Sciences, 10(1),* 309–322.

27. Fontaine, C., Galand, F. (2007). A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security, 2007(1),* 1–10.
28. Yi, X., Paulet, R., Bertino, E. (2014). Homomorphic encryption. In: *Homomorphic encryption and applications*, pp. 27–46. Cham: Springer.
29. Suwandi, R., Nasution, S. M., Azmi, F. (2016). Okamoto-Uchiyama homomorphic encryption algorithm implementation in e-voting system. *2016 International Conference on Informatics and Computing (ICIC)*, pp. 329–333. Hammamet, Tunisia, IEEE.