International Journal of Wireless Communications and Mobile Computing
2025, Vol. 12, No. 2, pp. 81-92
https://doi.org/10.11648/j.wcmc.20251202.13

Research Article

Lightweight Blockchain Framework for Securing Internet of Things Payment Systems

Gabriel Babatunde Iwasokun[1, 2, *], Oluwaseyi Wuraola Segun[1], Samuel Oluwatayo Ogunlana[3], Michael Abejide Adegoke[1, 4], Johnson Adeleke Adeyiga[1], Ojo Stephen Aderibigbe[4]

[1]Department of Computer Science and Information Technology, Bells University of Technology, Ota, Nigeria
[2]Department of Computer Science, Federal University of Technology, Akure, Nigeria
[3]Department of Computer Science, Adekunle Ajasin University, Akungba-Akoko, Nigeria
[4]Department of Computer Science, Lagos State University of Science and Technology, Ikorodu, Nigeria

Abstract

The integration of Internet of Things (IoT) devices into modern payment systems has introduced innovative functionalities, but also significant security and performance challenges. IoT devices, such as smart sensors, wearables, and automated vending machines, are typically resource-constrained yet handle sensitive financial transactions that demand robust security mechanisms. Conventional cryptographic solutions are often unsuitable for these environments due to their high computational and memory requirements. This paper presents the design of a lightweight blockchain-based model to secure IoT payment systems by leveraging the Ethereum blockchain and AES-128 encryption. The blockchain token is encrypted with AES-128 to add layer of security before being stored in a database. The model is designed to employ a decentralised digital ledger to record and validate transactions without a central authority, and the transaction is grouped into a block and linked to the preceding block through cryptographic hashes. The chain of blocks forms an immutable record that enhances transparency and security, and the distributed nature of blockchain networks, wherein multiple participants validate each transaction, minimises the risk of fraudulent activities while ensuring consensus is achieved through predefined protocols. Analysis of results from the implementation established the minimization of computational overhead and robust security measures, and was particularly beneficial where the scalability of decentralized systems is required alongside heightened security protocols.

Keywords

Payment Systems, Lightweight Blockchain, Cryptography, Advanced Encryption Standard

1. Introduction

IoT is a dynamic and transformative invention that joins billions of devices to the Internet for all-in-one communication and automation across businesses and areas. The number of IoT-connected devices globally has exceeded 14 billion [1],

*Corresponding author: gbiwasokun@bellsuniversity.edu.ng (Gabriel Babatunde Iwasokun)

Received: 14 August 2025; **Accepted:** 26 August 2025; **Published:** 15 September 2025

a figure projected to grow exponentially. This growth has driven innovations in areas such as smart homes, healthcare, and logistics, particularly in financial transactions. IoT devices facilitate micropayments, subscription models, and machine-to-machine (M2M) transactions, making secure payment systems critical components of IoT infrastructure. Blockchain technology has also emerged as a powerful catalyst for IoT payment systems due to its distributed, transparent, and foolproof landscape. Contrary to the old-style centralized systems, blockchain eradicates the need for third parties by confirming transactions using harmonious instruments [2]. This method enhances security and also reduces transaction costs, making blockchain suitable for IoT systems. For example, towards guaranteeing efficiency and reliability in financial interactions, smart contracts are used to drive IoT tools to independently execute payment for contracts [3]. IoT payment systems, which allow devices to communicate with one another and make financial transactions autonomous, require a robust security framework to protect sensitive information and ensure reliability. Given the vast number of connected devices and the volume of data generated, security, privacy, and scalability are the principal concerns. Secure and efficient transactions are part of the basic requirements for a good payment system, even in dispersed environments where devices operate independently. Cryptography and blockchain technology are also forming a sure partnership for ensuring the confidentiality, integrity, and transparency of IoT systems. While blockchain addresses several IoT payment challenges, the combination of cryptographic security in IoT environments presents significant hurdles. Conventional cryptographic methods like Rivest-Shamir-Adleman (RSA) and Advanced Encryption Standard (AES) are computationally complex and demand significant processing resources such as power and memory. IoT devices are typically resource-constrained and formulated for synergies with minimal hardware capabilities as a means of achieving costs and power conservation [4]. A characteristic IoT device may have a processing power of a few megahertz with limited battery life, which makes it unsuitable for the effective implementation of heavy cryptographic protocols.

The scalability of IoT networks also adds complexity to cryptographic implementation as millions of devices communicate simultaneously, ensuring real-time encryption and decryption processes without causing latency, which becomes a formidable challenge. The trade-off between security and performance often leaves IoT payment systems vulnerable to data breaches, man-in-the-middle (MITM) attacks, and unauthorized access [4, 5].

The rapid proliferation of IoT devices in payment systems has created significant opportunities for innovation but also exposed critical vulnerabilities in security and efficiency. IoT-based payment systems often involve resource-constrained devices, such as smart sensors, wearables, and automated vending machines, which handle sensitive financial transactions. These devices require robust cryptographic mechanisms to ensure data security and transaction integrity. However, existing cryptographic solutions are predominantly designed for conventional computing environments with ample resources, leading to inefficiencies when applied to IoT systems [4]. Existing literature underscores the need for cryptographic solutions tailored to IoT environments, yet significant gaps remain unaddressed. Many studies focus on general-purpose lightweight cryptographic algorithms but fail to integrate these solutions effectively with blockchain technology, which is critical for secure and decentralized payment systems. Furthermore, while blockchain provides a robust framework for transaction validation and data immutability, its computationally intensive consensus mechanisms, such as Proof-of-Work, exacerbate resource constraints in IoT applications [3]. Additionally, current research often overlooks the trade-offs between security robustness and system performance in blockchain-based IoT payment systems. Solutions that prioritize security frequently sacrifice efficiency, making them impractical for real-world deployment. Conversely, attempts to optimize performance sometimes compromise critical security features, leaving systems vulnerable to attacks, such as data breaches, MITM attacks, and unauthorized access [6]. To address these challenges, there is a pressing need for a lightweight cryptographic solution specifically designed for blockchain-based IoT payment systems. Such a solution must strike a balance between security and efficiency, ensuring the confidentiality, integrity, and availability of transactions while accommodating the resource limitations of IoT devices. Bridging this gap will not only enhance the reliability and scalability of IoT payment systems but also pave the way for broader adoption across industries. The proposed blockchain model provides an additional layer of security which makes it significantly different from the existing models. The additional layer ensures that computational overhead is minimized, while maintaining robust security measures. This also gives it an edge in term of scalability of decentralized systems that is often required alongside heightened security protocols. The new model also advanced the existing models by providing a viable pathway for deploying blockchain systems in resource-restricted environments and enhancing the overall security of IoT payment system through the application of AES encryption algorithm for safe transfer of tokens.

2. Review of Related Works

Ding et al. [7] presented a lightweight key synchronization update algorithm as part of a secure communication protocol, demonstrating its ability to resist common attacks and outperform other schemes in terms of randomness and computational performance. Based on the comparison of over 50 existing algorithms in terms of implementation cost, performance, and attack resistance, Thakor et al. [8] emphasized the importance of lightweight cryptography for resource-constrained IoT devices. Pajooh et al. [9] presented a

multi-layer blockchain security model for network security. The model utilizes clustering techniques and a hybrid evolutionary computation algorithm to define K-unknown clusters within the IoT network. It also engages local private blockchains for cluster-level communications and a global blockchain for base station interactions, balancing network latency and throughput compared to traditional global blockchain approaches. Hybrid architectures that combine blockchain and lightweight cryptography for enhancing IoT payment systems were presented by Gupta et al. [10], Ray et al. [11] and Zhang et al. [12]. The systems promote payment verification processes, channel networking and efficient and secure IoT transactions, though susceptible to scalability, privacy preservation, and standardization issues [4, 13]. Maftei et al. [14] proposed a distributed data storage solution for eliminating centralized network topology in the implementation of IoT devices. Tukur et al. [15] introduced an edge-based blockchain-enabled anomaly detection technique that uses smart contracts to detect and correct abnormalities in incoming sensor data. Alkhader et al. [16] combined Ethereum smart contracts with the InterPlanetary File System (IPFS) for decentralized storage of IoT device records and manufacturing details. Sefati et al. [17] introduced the BFLIoT framework that integrates blockchain, federated learning and edge computing to form a scalable, secure, and low-latency solution for smart city applications. The implementation of the framework demonstrated a trend towards leveraging off-chain storage, sidechains, and smart contracts to address the challenges of data management, security, and scalability in IoT systems.

The integration of blockchain technology with IoT systems has been an area of active research, particularly in securing decentralized payment systems. Despite the substantial body of work in this area, several gaps remain in terms of efficiency, scalability, and security that hinder the widespread adoption of blockchain-based IoT payment systems. The existing gaps include inefficiency of traditional cryptographic methods when applied to IoT systems, failure to establish a balance between security and resource consumption, lack of consideration for scalability, over-reliance on scaling techniques like sharding or sidechains [12, 18-20]. These gaps underscore the need for further exploration and innovation which justifies the research.

3. Proposed System

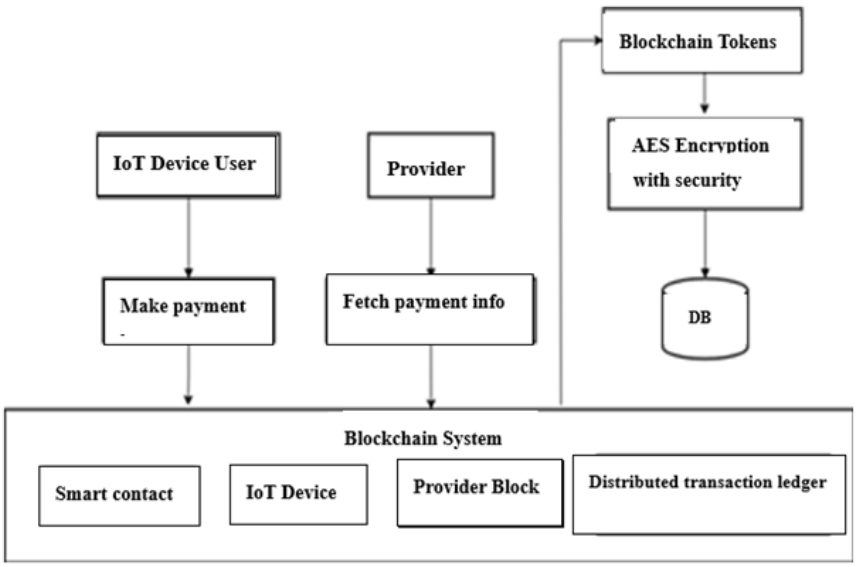

Figure 1. *Architectural View of the proposed Model.*

The proposed model is conceptualized in Figure 1. The model employs a decentralised digital ledger to record and validate transactions without the need for a central authority, and each transaction is grouped into a block and linked to the preceding block through cryptographic hashes. This chain of blocks forms an immutable record that enhances transparency and security. The distributed nature of blockchain networks, wherein multiple participants validate each transaction, minimises the risk of fraudulent activities while ensuring that consensus is achieved through predefined protocols. In addition to its inherent security features, the model engages an Advanced Encryption Standard with a 128-bit key (AES-128) to protect sensitive data. Blockchain tokens, which represent digital assets or access rights, are encrypted

using AES-128 to maintain confidentiality. The encryption process involves converting plaintext token data into ciphertext using a symmetric key algorithm. This procedure secures the tokens so that only parties possessing the correct decryption key can access the original token data. The AES-128 algorithm is recognized for its balance between computational efficiency and strong security, which is essential in environments with high transaction volumes. The integration of AES-128 encryption into blockchain increases the security of token-based transactions by ensuring that token data, even when intercepted, remains inaccessible to unauthorized users. The encrypted tokens are later decrypted by the intended recipient using the corresponding key, ensuring secure and accurate value transfer. The IoT Device is the end user who owns and operates the IoT devices. In the model, the IoT Devise user initiates the payment services or any other functionality provided through any connected device to the blockchain infrastructure. This component is the demand side of the ecosystem and drives the payment transactions. The Provider is an individual (vendor) who offers services to IoT device users and manages the business side of the system, setting prices, configuring service options, and receiving payments. The provider is also responsible for fetching the payment information for transaction verification and processing. In research contexts, a Provider could be a utility company, software service provider, or any other entity that monetizes IoT functionality. The Make Payment is the business opening process where the IoT Devise user commences a payment using any reconfigurable gateway. This component handles user authentication, payment method selection, and initial transaction validation. It is the gateway for monetary movements in the system and entails interfaces for amount specification and payment confirmation. The Fetch Payment Information component retrieves necessary payment details from the provider's systems and includes service costs, payment terms, account information, and transaction parameters. It ensures the provider's requirements are correctly integrated into the payment process, maintaining consistency between provider expectations and actual payments.

3.1. Blockchain Tokens

These are the digital assets used within the system for value transfer, representing standardized units of value on the blockchain and enabling the tokenization of real-world assets or services. They also represent the utility tokens specific to the IoT ecosystem tokens. The token design included considerations for divisibility, transfer mechanisms, and compatibility with the IoT devices' computational constraints.

3.2. AES Encryption with Security Key

This is the cryptographic security layer that protects transaction data and provides confidential and secure communication between system components. The security keys manage access control and data protection, ensuring that sensitive payment information remains protected from unauthorized access. For IoT systems, this component addresses the critical security concerns of handling financial transactions on potentially vulnerable connected devices.

3.3. Provider Block

This represents the provider's presence and operations on the blockchain and includes the provider's account, services offered and business information. It is saddled with the responsibility of preserving the provider's state within the dispersed system and playing the role of blockchain identity management and maintaining the quality of service.

The Distributed Transaction Ledger is the essential blockchain component that manages an incontrovertible record of the various transactions. It ensures transparency, non-repudiation, and consistency across the network. The distributed nature provides resilience against failures and attacks, while the ledger functionality creates an auditable history of all system activities.

3.4. System Integration

The blockchain components depict how the system is integrated, while its architectural boundary presents the blockchain environment that processes and validates transactions. The integration framework shows the connection between the traditional IoT devices and the distributed ledger technology on one hand and the connection between the conventional embedded systems and the decentralized financial technology on the other hand.

The blockchain security components of the proposed model utilize a distributed ledger technology to record and verify transactions in a secure and immutable manner. Each transaction is classified into a block that is linked cryptographically to the previous block. The transaction initiated on an IoT device using Ethereum blockchain (structure presented in Figure 2) is broadcasted to the network of interconnected nodes. The nodes validate the transaction based on the Proof-of-Stake consensus mechanism, and once consensus is achieved, the transaction is added to a permanently recorded block on the blockchain. The transaction data includes sender and recipient wallet addresses and the amount transferred. The decentralized nature of blockchain prevents the necessity for middle-level financial agencies and reduces transaction fees and processing times. Furthermore, every block has a timestamp and a cryptographic hash that ensures data integrity. Modification to transactional data always affects the hash value, and hence, its perception is always taken as a possible fraudulent activity. The proposed blockchain payment system incorporates public and private key asymmetric cryptography for securing the digital signatures. The sender’s private key is used to login for the transaction, while the corresponding

public key is used for verification purposes. The design guarantees the processing of only authenticated transactions.

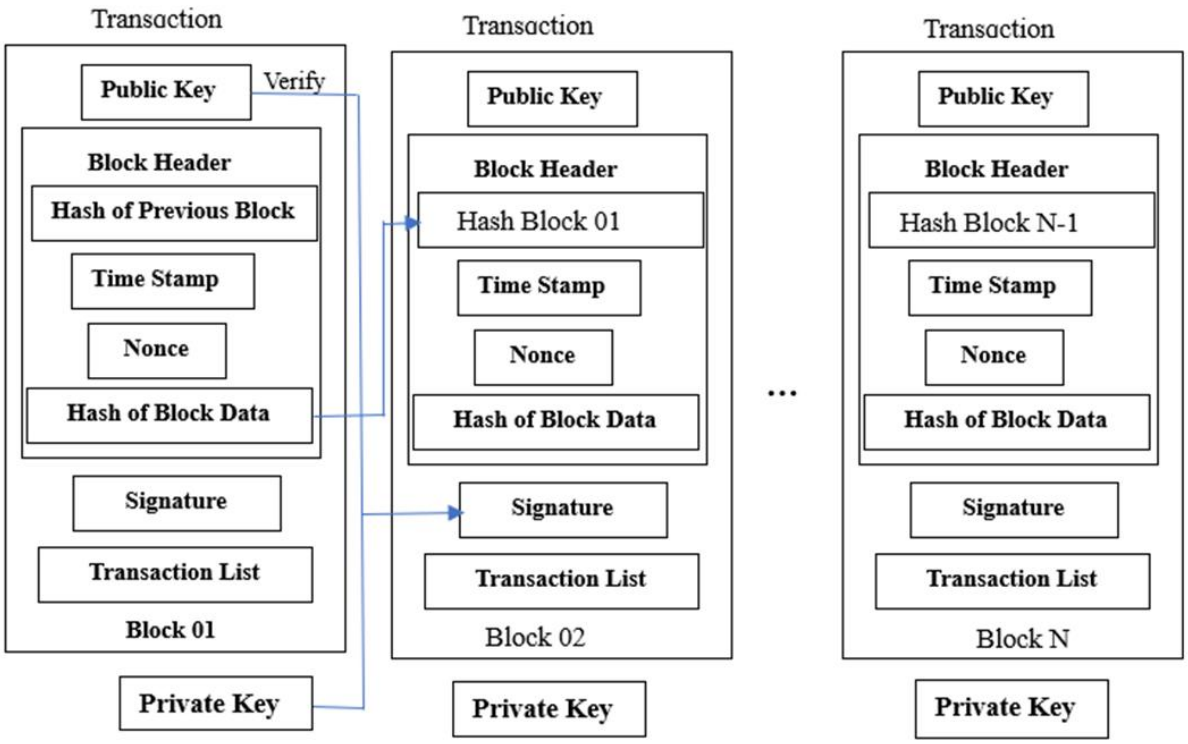

Figure 2. *The Structure and Transaction of Ethereum Blockchain [21].*

As shown in Figure 2, the Ethereum Blockchain framework comprises a block header, signature and transaction lists. The block header encapsulates relevant metadata, including the parent block hash, the mining difficulty, the timestamp, and the nonce that validates the block. The transactions section enumerates individual transactions executed within the blockchain, while the blocks contribute to network security by reducing centralization. Transactions in Ethereum are pivotal, functioning through a consensus Proof of Stake mechanism. Each transaction has essential elements like the recipient's address, the sender's signature, and the transaction value. The Smart contracts serve as the cornerstone of the Ethereum blockchain and oversee the enforcement of all contractual agreements to foster reliable and transparent transactions.

3.5. Smart Contracts

A smart contract, running on a blockchain, is a code that is used to execute, verify, or distribute contracts based on the available information. The code is sent from a blockchain wallet alongside its receiver's address in a manner similar to the transfer of value. The proposed model also utilizes a smart contract to record user-verifiable information and accumulate it into a verification tag, ensuring the verifiability of search results. It also allows only the Certification Authority to complete user registration, verify information upload, and tag based on the following algorithm [22]:

1) $Require$: $Task\ name$,$invoked\ parameters$
2) $Ensure$: $Setti\quad g\ up\ tasks$:
3) $mapping\ (string \Rightarrow mapping\ ((bytes32 \Rightarrow bytes))\ veri\_tag$;
4) $mapping\ (address \Rightarrow string)\ User$; address $addr_{ca}$;
5) $constructor$ ()
6) $addr$aca=$msg.sender$
7) $function\ Register(string\ ID, address\ addr)\ public$
8) $require(msg.sender = addr_{ca};)$;
9) $user[addr]$=1 D;
10) function $Upload(bytes32\ v_1, bytes\ v_2)\ public$
11) $require(User[msg.sender]! = null)$;
12) bytes $bytes\ \tau = Veri\_tag[User[msg.sender]][v_1]$
13) $if\ \tau.length == (\ )$;
14) $\tau = g$;
15) $Veri\_tag[User[msg - sender][v_1] = T^{v_1}\ mod\ N$;
16) $function\ quer\quad (string\ t1, bytes32\ t2)\ public\ view\ returns\ (bytes\ \tau)$
17) $require(user[msg.sender]!=Null)$
18) τ = $veri\_tag$ [$t1$] [$t2$]

3.6. AES Encryption

The Advanced Encryption Standard (AES) symmetric al-

gorithm is used to secure sensitive transaction data based on fixed-size data blocks and keys of 128, 192, or 256 bits. Motivated by the need to ensure robustness against various cryptographic attacks, the adopted AES encryption and decryption are in the stages presented in Figure 3 [23]. The encryption process begins with Key Expansion, where the initial key undergoes a series of transformations to produce a set of round keys, which are subsequently utilized in the encryption rounds. This is followed by the Initial Round (AddRoundKey), which involves XOR-ing the first block of plaintext with the initial round key, which offers the first level of security. The AES is then subjected to a series of Main Rounds, which comprises the following nine rounds for AES-128:

1) SubBytes: This is a non-linear substitution step where each byte is replaced with another specific byte by a fixed substitution table (S-box).
2) ShiftRows: This is a transposition step where each row of the state is shifted cyclically by a certain number of bytes.
3) MixColumns: This is a mixing operation that combines the bytes within each column of the state matrix to provide further diffusion.
4) AddRoundKey: This is similar to the initial round; each byte of the state is combined with a block of the round key.
5) Final Round: This is similar to the main rounds but omits the MixColumns step. It comprises the SubBytes, ShiftRows, and AddRoundKey processes only.

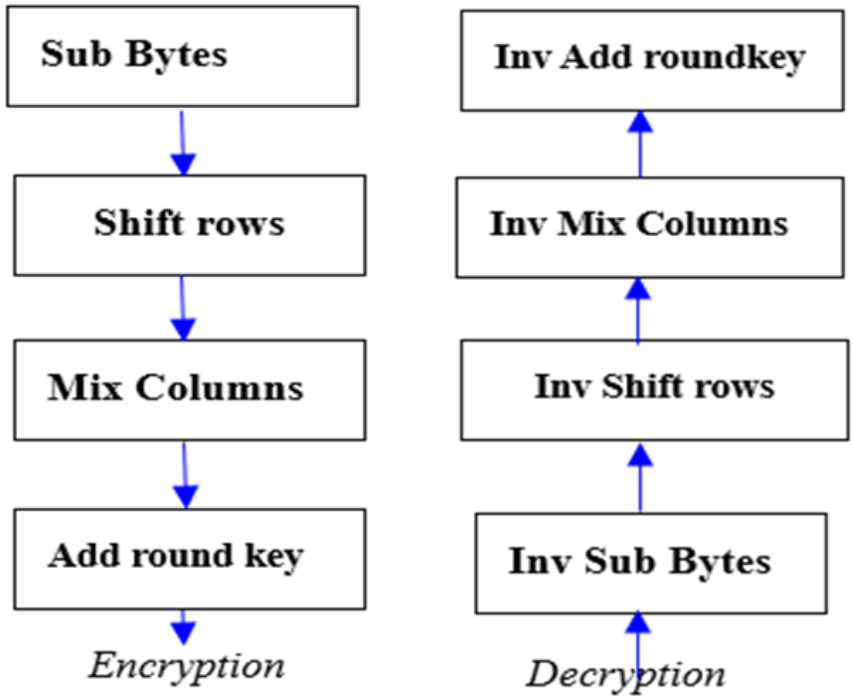

Figure 3. *AES stages [23].*

Upon completion of these rounds, the output from the last AddRoundKey becomes the ciphertext. This multi-stage approach ensures that AES encryption is robust and resistant to known cryptographic attacks.

3.7. Database

The database serves as the system's memory for user information, transaction history, and possibly device states, allowing for analytics, auditing, and service continuity. Its architecture requires balancing centralized management efficiency and distributed resilience, which is particularly important for IoT systems that may operate in environments with intermittent connectivity.

MongoDB, a document-oriented NoSQL database, was used to provide a flexible schema that can adapt to the dynamic nature of blockchain data. The database schema for the design has three interconnected tables for *User*, *Transactions*, and *Token*. Each table plays a crucial role in managing user data, recording transactions, and handling authentication tokens. The relationships between these tables ensure data consistency and integrity within the system. The *User* table serves as the foundation of the database, storing essential details about individuals interacting with the system. It includes an Identification (ID), which uniquely identifies each user; a username, which may be used for login or display purposes; and an email for communication or account recovery. This table acts as a reference point for other tables, en-

suring that transactions and tokens are always linked to authenticated users. The *Transactions* table records all actions performed by users. Each transaction has a unique ID and is associated with a user through the *userID* field, establishing a connection to the *User* table. The table also includes a *hashblock*, required for security and data integrity verification, and a *time* field that logs the exact moment the transaction occurred. The *Token* table is designed to manage authentication and verification processes. Each token is uniquely identified by an ID and is linked to a specific user through the *userID* field. The token table also maintains a relationship with the *Transactions* table via the *txnID* field, ensuring that tokens can be tied to specific transactions when needed. The token field stores the authentication token, which could be used for session management, verification, or security purposes. There is also an additional field, *Field1*, to be used for extra metadata. The relationships between these tables are essential for maintaining data integrity. The *User* table has a one-to-many relationship with both the *Transactions* and *Token* tables, meaning a single user can have multiple transactions and multiple tokens associated with them. Additionally, the *Transactions* table has a one-to-one or one-to-many relationship with the *Token* table, depending on whether multiple tokens are allowed for a single transaction. These relationships allow for seamless tracking of user activities and authentication processes. In conclusion, this database schema efficiently organizes user-related data while ensuring robust security and accountability mechanisms. By linking *users*, *transactions*, and *tokens*, the system maintains a well-structured approach to managing authentication, transaction logging, and user activity tracking.

3.8. Blockchain Tokens

The Ethereum blockchain token is used to facilitate transactions, supports decentralized applications, and supplies the computational fuel for smart contracts execution in the network. Given its dual role, the token is essential for both value transfer and incentivizing miners who conduct the proof-of-work process, although recent developments have moved the protocol toward alternative consensus mechanisms such as proof-of-stake. The issuance, circulation, and storage of tokens are managed through a transparent, public ledger that is continuously updated across a distributed network of computer nodes, and they may be stored in digital wallets that rely on cryptographic techniques to ensure secure interactions. The tokens for the IoT payment system will be generated based on the following [24, 25]:

1. Input:
1) tokenName → Name of the token
2) tokenSymbol → Symbol for the token
3) totalSupply → Maximum number of tokens to be created
4) decimalUnits → Decimal precision of the token
5) ownerAddress → Blockchain wallet address of the token creator
2. Output:

DeployedTokenContract → A functional ERC-20 token smart contract

Steps Smart Contract:

1. Initialize Smart Contract
1) Define a Solidity contract that follows the ERC-20 standard.
2) Set token attributes (tokenName, tokenSymbol, decimalUnits, totalSupply).
3) Assign totalSupply to ownerAddress upon deployment.
2. Define Token Storage Variables
1) Create mappings to store balances (mapping (address => uint256) balances).
2) Define an allowance mapping to permit third-party token spending.
3. Implement ERC-20 Token Functions
1) function balance of (address account) returns (uint256) → Returns token balance of an account.
2) function transfer (address recipient, uint256 amount) → Transfers tokens from sender to recipient.
3) function approve (address spender, uint256 amount) → Allows spender to withdraw tokens.
4) function transferFrom (address sender, address recipient, uint256 amount) → Transfers tokens using allowances.
4. Deploy Smart Contract to Blockchain
1) Compile the Solidity contract using Truffle.
2) Deploy to Ethereum (or private blockchain) using Ganache.
3) Verify the deployment transaction.
5. Secure IoT Transactions Using the Token
1) Integrate the smart contract with IoT devices using Web3.js or an API.
2) Implement cryptographic signatures for IoT payments.
3) Record all transactions on the blockchain for security and transparency.

4. Experimental Study

The experimental study was carried out on an HP laptop with 8GB of RAM and 500GB of storage on an Intel processor. The software requirements include Solidity, Ganache, Truffle, and MongoDB. Solidity is a contract-oriented programming language specifically designed for implementing smart contracts on the Ethereum blockchain. It is statically typed and supports inheritance, libraries, and complex user-defined types. Ganache, which is a personal Ethereum blockchain, was used to run tests, execute commands, and inspect state while controlling how the chain operates, followed by a Raspberry Pi-inspired simulation on a virtual machine. Truffle served as a development framework for Ethereum, and it has tools that create, deploy and test smart contracts. Raspberry Pi operating system in a virtual environment created by Vmware and a popular hypervisor as a means of avoiding heightened cost. Necessary libraries and software dependencies, such as

Node.js, were installed on the Raspberry Pi operating system, while Truffle was installed by executing the command "npm install -g truffle" in the terminal. Next was the installation of Ganache using a command-line version known as Ganache CLI on a Raspberry Pi. This methodical setup ensures a secure, consistent environment for blockchain development, experimentation, prototyping and testing without significant hardware investments.

The blockchain model was implemented using Truffle and Ganache, while AES-128 was implemented for the Lightweight cryptography to enable the model to operate in environments with constrained resources. The AES-128 provides essential mechanisms for data confidentiality, authenticity, and integrity without imposing heavy computational demands. The user interface was implemented using Web3 library via Solidity, which is a statically-typed programming language designed for creating smart contracts that run on Ethereum. The user interface offers a select menu for IoT users to choose a provider, an input field for the number of Ethereum to be transferred, and two buttons. The first button sends payment, while the second one displays the balance. Once the amount to be paid is specified, the send payment button is clicked to invoke a script that generates the blockchain token, which is encrypted using AES-128 before sending to the provider wallet. Upon receipt in the wallet, the decryption algorithm is activated prior to entering the provider blockchain address. The various blockchain addresses created on the Ganache is presented in Table 1. The first address is for an IoT device user, while the second address is for a service provider. The IoT device user has a 30 Ethereum balance while the provider has a 100 Ethereum balance before the transaction. Table 2 presents the post-transaction Ethereum balance in each of the addresses with the provider's balance increased from 100 to 101 while the IoT device User's balance reduced from 30 to 29.

Table 1. *Blockchain Accounts addresses and Balance Before Transactions.*

Address	Ethereum balance
0×bc1415C5059aC055aE44Bd02ff1ad59AbEA8123f	30.00
0×76770DdEb60e538C7A4fE43583772c4cfED09Aab	100.00
0×3888942905f12974Fb0306D5E578da79811b0aE0	100.00
0×63E88C0CbcE2870481412F7EB633Bbc30fA56b80	100.00
0×Ae681107377e36f0DC0cDb0b26B2E2bba54c312d	100.00
0×E45a9b1E8b905d16737F72CD47500e18e32E24CB	100.00
0×aB7719e5604e76A38cD27b65b4dA652137dBf0bc	100.00

Table 2. *Blockchain Accounts Addresses and Balance After Transactions.*

Address	Ethereum balance
0×bc1415C5059aC055aE44Bd02ff1ad59AbEA8123f	29.00
0×76770DdEb60e538C7A4fE43583772c4cfED09Aab	101.00
0×3888942905f12974Fb0306D5E578da79811b0aE0	100.00
0×63E88C0CbcE2870481412F7EB633Bbc30fA56b80	100.00
0×Ae681107377e36f0DC0cDb0b26B2E2bba54c312d	100.00
0×E45a9b1E8b905d16737F72CD47500e18e32E24CB	100.00
0×aB7719e5604e76A38cD27b65b4dA652137dBf0bc	100.00

The integration of Truffle with Ganache was used to simulate the blockchain environment that can mimic real-world complex scenarios. The simulation allows the evaluation of how the lightweight cryptographic algorithms perform under various network and controlled environment. The simulation provides insight into potential vulnerabilities and allows for timely updates and improvements.

The combination of blockchain model and lightweight cryptography offers an additional layer of security, promotes computational efficiency and gives room for scalability of decentralized systems which is required alongside heightened security protocols.

Result and Discussion

Ten transactions were initiated on the blockchain on two occasions and the latency and CPU utilization of the transactions computed. On the first occasion, AES-128 lightweight cryptography was engaged while the second occasion was without it. Table 3 and Table 4 present the result for the two occasions. The average CPU utilization when lightweight cryptography was employed is 5.33 while the average latency is 2.4ms.

Table 3. *Performance Evaluation of the Blockchain Based Payment which utilizes Lightweight Cryptography.*

Transactions	CPU Utilization (%)	Latency (ms)
1	5.01	2
2	5.01	2
3	5.02	2
4	6.76	4
5	5.50	2
6	5.03	2
7	5.01	2
8	5.02	3
9	5.04	2
10	5.98	3
Average	5.33	2.4

Table 4. *Performance Evaluation of the Blockchain Based Payment without Lightweight Cryptography.*

Transactions	CPU Utilization (%)	Latency (ms)
1	10.21	5
2	10.19	5
3	10.21	5
4	12.06	7
5	8.01	4
6	7.03	4
7	10.01	4
8	10.06	4
9	10.07	4
10	10.19	5
Average	9.79	4.7

Table 5. *Blockchain Transactions with AES Encryption.*

Transaction ID	Sender	Receiver	Amount (ETH)	Status	AES Applied?	Block No.
Txn001	Wallet A	Wallet B	2.0	Successful	Yes	1
Txn002	Wallet B	Wallet C	1.5	Failed	No	2
Txn003	Wallet A	Wallet D	3.0	Pending	Yes	3
Txn004	Wallet C	Wallet E	0.8	Successful	Yes	3
Txn005	Wallet D	Wallet F	4.2	Successful	No	4
Txn006	Wallet E	Wallet G	2.5	Pending	Yes	5
Txn007	Wallet F	Wallet H	1.0	Successful	Yes	5

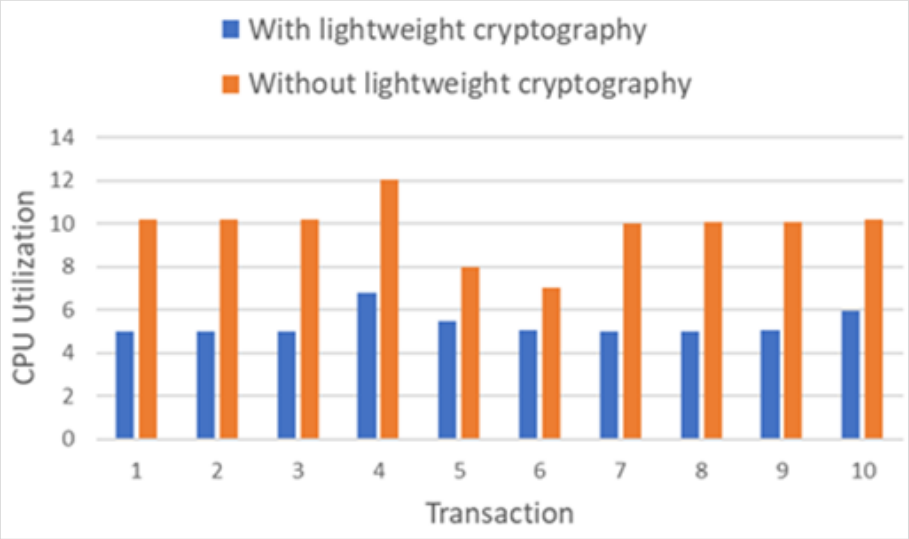

Figure 4. *CPU utilization with and without lightweight cryptography.*

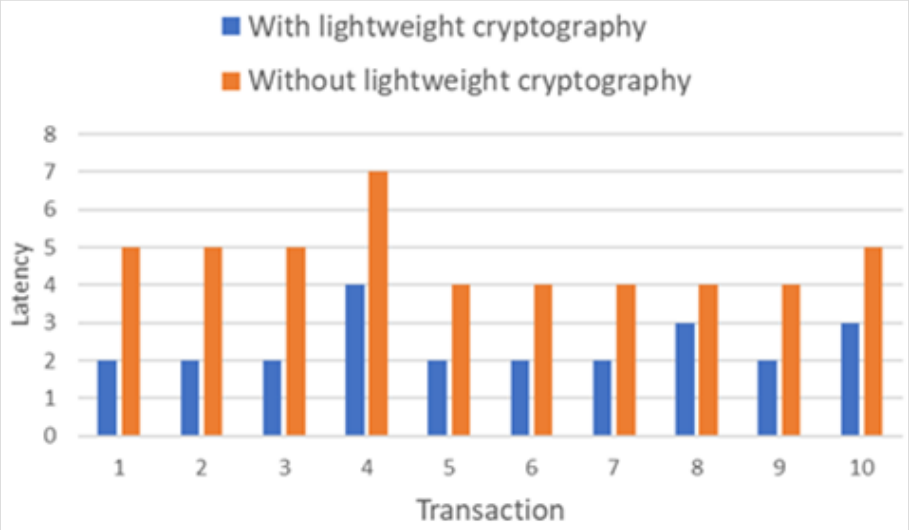

Figure 5. *Transaction latency with and without lightweight cryptography.*

Figure 4 and Figure 5 present the comparative analyses of the results for the two performance indices. The average CPU utilization for transactions without lightweight cryptography is 9.79 while average latency is 4.7ms. It is revealed from Tables 1 and 2 that there is a superior performance in terms of CPU utilization and latency when lightweight cryptography was employed in all the 10 trials of the experiment. With reduced CPU utilization and latency, lightweight cryptography encryption extended the operational lifespan of IoT devices for the transactions. The incorporation of blockchain token encryption further secured the payment process by generating tokens that represent payment values. The tokens undergo encryption based on methods that preserve a high level of data integrity, confidentiality and availability while fulfilling the performance criteria expected of any IoT applications. The specifics of the blockchain transactions, with sender and receiver wallets, the transferred amount and status, AES encryption usage, and the block number in which each transaction is recorded are presented in Table 5. Out of the seven transactions recorded in Table 5, four were successful, two remained pending, and one failed. The failed transaction, Txn002, did not use AES encryption, which may suggest a possible security or validation issue. AES encryption was applied in four of the seven transactions to ensure secure data transmission with Txn001, Txn004, and Txn007 all successful. The successful transactions are distributed across different blocks, indicating the effectiveness of the encryption method. The failed transaction, Tnx005, did not implement AES, and this development further buttressed its role in transaction integrity. Some blocks, such as Block 3 and Block 5, recorded multiple transactions while Txn003 and Txn006 recorded pending transactions, which could be attributed to factors like network congestion, gas fees, or validation delays. These results established that AES encryption contributed significantly to the security and successful

execution of the transactions.

5. Conclusions

The design and implementation of a lightweight cryptography blockchain model for securing IoT payment systems has been presented. The model emphasized security, efficiency, and scalability in modern decentralized applications. Lightweight cryptography was designed to operate in environments with constrained resources and provide essential mechanisms for data confidentiality, authenticity, and integrity without imposing heavy computational demands. This makes it ideal for the integration of blockchain applications in IoT environments and mobile devices.

The design also involved the usage of standardized AES-128 lightweight cryptographic algorithms that are verified for both security and efficiency. These algorithms were integrated into blockchain infrastructure using robust development frameworks such as Truffle, a widely recognized development framework for Ethereum applications, and Ganache, a personal blockchain for rapid testing and development. These frameworks were used to create, deploy, and test smart contracts with the incorporation of lightweight cryptography. During implementation, Truffle was used to streamline the compilation, linking, and migration of smart contracts. Its integration with Ganache was also used to simulate the blockchain environment with a view to mimic real-world complex scenarios and allow the evaluation of how lightweight cryptographic algorithms perform under various network conditions in a controlled environment. The integration also provided insight into potential vulnerabilities and allow for timely updates and improvements. Analysis of results from the implementation established the minimization of computational overhead, maintain robust security measures, and particularly beneficial where the scalability of decentralized systems is required alongside heightened security protocols. It is also revealed that the proposed model provides a viable pathway for deploying blockchain systems in resource-restricted environments and enhances overall system security. This model integrated AES-128 cryptography with blockchain among numerous lightweight cryptography algorithm and was simulated using a local development environment. However, real blockchain network are often deployed on either public or private blockchain network. It is noteworthy that scalability is a significant challenge facing the proposed systems. An increase in the number of connected devices leads to an exponential growth in the demand for scalable solutions to handle massive volumes of transactions. The present form of the proposed payment systems lacks the capacity to handle the sheer volume of microtransactions that the IoT applications may require. Future work therefore aims at exploring other cryptography models like One-Time Password (OTP) Token which is a Two-Factor Authentication (TFA), Time-Based One-Time Password (TOTP) Token, which is a Secure Login and HMAC-SHA256 Token, which is an Application Programming Interface (API) for achieving higher integrity and improved security and scalability of the IoT payment system.

Abbreviations

IoT	Internet of Things
M2M	Machine-to-Machine
RSA	Rivest-Shamir-Adleman
AES	Advanced Encryption Standard
MITM	Man-in-the-Middle
ID	Identification

Author Contributions

Gabriel Babatunde Iwasokun: Conceptualization, Methodology, Project administration, Resources, Software, Supervision, Writing, review and editing of original draft.

Oluwaseyi Wuraola Segun: Data curation, Formal Analysis, Investigation.

Samuel Oluwatayo Ogunlana: Data curation. Formal Analysis, Resources, Investigation.

Michael Abejide Adegoke: Methodology, Resources, Supervision, Writing, review and editing of original draft.

Johnson Adeleke Adeyiga: Methodology, Resources, Supervision, Writing, review and editing of original draft.

Ojo Stephen Aderibigbe: Methodology, Resources, Supervision, Writing, review and editing of original draft.

Funding

This work is not supported by any external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Statista, P. (2023). Growth of Mobile Payment Systems Worldwide.

[2] Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System.

[3] Qu, X., Wang, S., Cheng, X., and Hu, Q. (2020). Proof of Federated Learning: A Novel Energy-Recycling Consensus Algorithm. *IEEE Transactions on Parallel and Distributed Systems*, *32*(8), 2074–2085.

[4] Khalil, U., Mueen-Uddin, M.-U., Malik, O. A., and Hussain, S. (2022). A Blockchain Footprint for Authentication of IoT-Enabled Smart Devices in Smart Cities: State-of-the-Art Advancements, Challenges and Future Research Directions. *IEEE Access*, *10*, 76805–76823.

[5] Goudarzi, S., Koushanfar, F., & Azizi, S. (2021). Lightweight cryptography for IoT systems: Challenges and solutions. IEEE Access, 9, 56234-56245.

[6] Gami, B., Mehra, P. S., Mishra, D. K., Agrawal, M., and Quasim, D. (2023). Artificial intelligence ‐ based blockchain solutions for intelligent healthcare: A comprehensive review on privacy preserving techniques. Transactions on Emerging Telecommunications Technologies, 34(9).

[7] Ding, Z., He, D., Choo, K.-K. R., Gao, Y., Qiao, Q., Li, X., and Chan, S. (2024). A Lightweight and Secure Communication Protocol for the IoT Environment. *IEEE Transactions on Dependable and Secure Computing*, 1050–1067.

[8] Thakor, V. A., Khandaker, M. R. A., and Razzaque, M. A. (2021). Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities. *IEEE Access*, *9*, 28177–28193.

[9] Pajooh H. H., Rashid, M., Demidenko, S., and Alam, F. (2021). Multi-Layer Blockchain-Based Security Architecture for Internet of Things. *Sensors*, *21*(3), 772.

[10] Gupta, N., Jain, M., & Agarwal, R. (2022). Lightweight cryptographic techniques for IoT security: A comprehensive survey. International Journal of Computer Science and Security, 16(3), 222-235.

[11] Ray, P. P., Kumar, N., and Dash, D. (2020). BLWN: Blockchain-Based Lightweight Simplified Payment Verification in IoT-Assisted e-Healthcare. *IEEE Systems Journal*, *15*(1), 134–145.

[12] Zhang, Y., Choo, K.-K. R., Gai, K., Zhu, L., and Xiao, J. (2022). Blockchain-Empowered Efficient Data Sharing in Internet of Things Settings. *IEEE Journal on Selected Areas in Communications*, *40*(12), 3422–3436.

[13] Ahakonye, L. A. C., Kim, D.-S., and Nwakanma, C. I. (2024). Tides of Blockchain in IoT Cybersecurity. *Sensors*, *24*(10), 3111.

[14] Maftei, A. A., Petrariu, A. I., Lavric, A., and Popa, V. (2023). Massive Data Storage Solution for IoT Devices Using Blockchain Technologies. *Sensors*, *23*(3), 1570.

[15] Alkhader, W., Yaqoob, I., Omar, M., Sleptchenko, A., Jayaraman, R., and Salah, K. (2021). Blockchain-Based Decentralized Digital Manufacturing and Supply for COVID-19 Medical.

[16] Tukur, Y. M., Awan, I., and Thakker, D. (2020). Edge ‐ based blockchain-enabled anomaly detection for insider attack prevention in the Internet of Things. *Transactions on Emerging Telecommunications Technologies*, *32*(6).

[17] Sefati, S. S., Fratu, O., Tal, I., Arasteh, B., Halunga, S., and Craciunescu, R. (2024). Cybersecurity in a Scalable Smart City Framework Using Blockchain and Federated Learning for Internet of Things (IoT). *Smart Cities*, *7*(5), 2802–2841.

[18] Liu, X., Wang, L., & Yang, Y. (2021). Blockchain in IoT: A survey of applications, challenges, and solutions. Future Internet, 13(6), 137.

[19] Charles, V., Gherman, T., and Emrouznejad, A. (2023). A critical analysis of the integration of blockchain and artificial intelligence for supply chain. *Annals of Operations Research*, *327*(1), 7–47.

[20] Chen, Z., Wang, X., & Zhang, Y. (2021). A survey on blockchain-based IoT security: Challenges and solutions. Future Generation Computer Systems, 108, 380-396.

[21] Liang, Q., Shi, N., Tan, Y. A., Li, C., and Liang, C. (2024). A stealthy communication model with blockchain smart contract for bidding systems. Electronics, 13(13), 2523.

[22] Montgomery, H., et al. (2020). Post-quantum cryptography: A survey of quantum-resistant algorithms. Quantum Information & Computation, 20(7), 529-546.

[23] Muthamilselvan, S., Shobana, R., Sujitha, J., & Varsha, K. (2024, October). Ethereum Smart Contract in Supply Chain Management. In 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS) (pp. 1-6). IEEE.

[24] Baksi, A., & Jang, K. (2024). Quantum Analysis of AES. In Implementation and Analysis of Ciphers in Quantum Computing (pp. 51-90). Singapore: Springer Nature Singapore.

[25] Bai, W., Zhang, X., & Liu, Y. (2021). Energy-efficient lightweight cryptographic algorithms for IoT security. International Journal of Applied Cryptography, 12(4), 342-358.