applied sciences

MDPI

Article

Designing a Secure and Scalable Service Agent for IoT Transmission through Blockchain and MQTT Fusion

Tse-Chuan Hsu

Department of Computer Science and Information Management, Soochow University, Taipei 100, Taiwan; tchsu@gm.scu.edu.tw

Abstract: With the rapid development of the Internet of Things (IoT) in recent years, many IoT devices use communication systems to transmit data. Data packets are inevitably at risk of tampering during data transmission, which can lead to information errors and damage during communication. Since IoT terminals are often operated under human supervision, it is essential to improve and ensure the transmission of information, avoid data tampering, and ensure the accuracy of packet transmission. This research successfully improves the message transmission method of IoT communication and the communication model by combining it with blockchain architecture. Combining communication protocols with blockchain serial connections eliminates the need to control operating and managing processes. By replacing message transmission with coverage, the reliability of data transmission in the IoT communication system is improved, and the flexibility of the data transmission process is enhanced. Through practical verification through experimentation, the study successfully improved the incapability of effectively constructing a cross-coded message transmission mode in the Message Queuing Telemetry Transport (MQTT) communication protocol, removed a single layer of encryption rule, and created an encryption mode capable of providing complex arrangement organization.

Keywords: Internet of Things (IoT); data transmission; blockchain architecture; communication protocols; message transmission; encryption mode; MQTT

Citation: Hsu, T.-C. Designing a Secure and Scalable Service Agent for IoT Transmission through Blockchain and MQTT Fusion. *Appl. Sci.* **2024**, *14*, 2975. https://doi.org/10.3390/app14072975

Academic Editor: Douglas O'Shaughnessy

Received: 27 December 2023
Revised: 1 March 2024
Accepted: 29 March 2024
Published: 1 April 2024

1. Introduction

Traditionally, Message Queuing Telemetry Transport (MQTT) has been the most widely used communication protocol in IoT. Its purpose is to send and receive messages over a narrow bandwidth and with low power consumption. Currently, information interaction through the broker adopts the publish/subscribe method. The current process is to register the model to reduce the conversion time of message processing. All subscribers or senders are not encrypted by default after registration. When data are transmitted in the IoT device endpoint, the publisher is activated to send a message to the broadcast station, and other IoT nodes receive it by subscription. However, the MQTT protocol encrypted by TLS (Transport Layer Security) on port 8883 needs to be improved. Messages are encrypted in transit but require TLS credentials that are updated on the fly to ensure confidentiality. In the conventional communication protocol, nodes uphold persistent connectivity with the transmitting station, primarily executing the registration operation during connection establishment. Notably, the payload of the packet remains unencrypted throughout the communication process. Instead, the security measures especially entail TLS-based encrypted connections, alleviating potential confidentiality concerns. However, while TLS can construct a secure transmission protocol in the same broadcast network environment [1], there is no way to prevent other nodes in the same broadcast network environment from exchanging data using the same security certificate. Therefore, the combination of MQTT and TLS [2] must be improved for IoT to perform message management operations under the same broker.

Therefore, how do we set up a system under an exact broadcast communication mechanism and ensure all the receivers can be under the exact broadcast communication mechanism and still carry out distributed transmission based on different encryption keys? Can we solve the grouping problem effectively, or is it a reliability mechanism on the off-load? The Internet of Things (IoT) ecosystem includes numerous interconnected nodes. In addition to facilitating data exchange between individual nodes, the research explores the potential formation of distinct clusters, each housing specific nodes. To facilitate this, multiple tags are used for grouping purposes. Tagging allows nodes to seamlessly join or leave clusters without compromising the encrypted integrity of Lianxun messages transmitted between individual nodes and the broadcast center. When the broker sends a message to the subscriber, because the message itself is exchanged and transmitted through the bound key, the unauthorized nodes cannot read and modify the data. This research uses an innovative architecture to propose a message transmission method to improve the communication security of the Internet of Things. Previous research has solved the problem of packet message security, but our study adds a blockchain encryption method, an innovative technological development. In the past, we reviewed relevant research on the IoT data transmission process, which mainly focused on encrypting data during the transmission process through TLS. Therefore, we found that if we can combine the encryption of the blockchain through the broadcast method, converting the key into a one-to-one and a pair-to-group method to encrypt the data transmission time, respectively, it can improve this problem and solve the technical limitation, as long as the broker can be registered to receive and decrypt broadcast packets. The information security issues of cross-encoding messaging led us to initiate further research on this topic. In this study, the background knowledge and relevant literature will be discussed in the Section 2, the experimental environment design will be constructed in the Section 3, the data analysis and discussion of the experimental results will be provided in the Section 4, and the research conclusions and contributions will be described in the Section 5.

2. Reviewing Background and Related Work

2.1. MQTT IoT Communication

MQTT is a simple communication protocol that allows IoT end devices to communicate by sending small data packets over the network. The minimum size of a message is as small as 2 bytes of data. Using a relatively streamlined data transmission method, the loss of power and network performance due to transmission can be reduced. The protocol has the characteristics of a high degree of reliability and robust scalability and has been widely used in a variety of IoT applications. Since MQTT is only a communication protocol, it is not a universal protocol used for network packet transmission in current cloud computing communications, such as HTTP/HTTPS/SSL. Therefore, if it can be integrated into an integrated architecture for cloud services and message exchange, it will provide another new way to exchange encryption keys.

The MQTT protocol was developed to facilitate communication between M2M machines on the Internet of Things communication system. Its purpose is to enable real-time data transmission and handle the role of cloud computing in processing large amounts of IoT data [3,4]. Mishra (2022) [5] examined the MQTT protocol and its relationship with database and software development applications in Internet of Things communication mechanisms. The study yielded positive results and highlighted MQTT as one of the most used IoT protocol solutions. This research delves into the MQTT protocol and analyzes its application in M2M and IoT systems. Through a comprehensive review and analysis of existing literature, the author discussed the application cases, performance evaluation, and possible future development directions of MQTT in various fields. At the same time, he mainly discussed specific instances of MQTT usage in different industries and fields, proving that MQTT helps make a positive contribution to the Internet of Things. The lightweight design of MQTT enables data exchange in plain text format, which may pose security risks. The current MQTT protocol requires improvement in TLS protection, specifically in the

encryption and decryption process. This may require additional computing resources and other technologies and could cause communication delays. However, experimental results from Paris (2023) [6] show that adding TLS to the encryption and decryption process can enhance security. It is important to note that this comes at the cost of increased overall IoT power consumption. This encryption mechanism may also increase the cost and technical difficulties associated with establishing communication security.

Considering the lack of corresponding security protection measures in MQTT, on Kadhim (2023) [7] believes that the security of IoT devices can be improved by cleverly combining TLS and MQTT. However, computing performance may be affected when using a combination of the lightweight MQTT protocol and the TLS protocol, especially for IoT devices with limited resources [8]. When using TLS for encryption, the message must be encrypted between the MQTT client and broker. It is protected before transmission and provides higher protection when transmitted to the server, effectively preventing unauthorized access. Integrating MQTT and TLS uses a protocol to protect the security of network communications. Its main functions include encryption, authentication, and data integrity protection. In MQTT, TLS is used to encrypt and protect the communication content of the MQTT protocol to prevent sensitive data from being eavesdropped on or tampered with by unauthorized parties. Its core functionality supports TLS by using data integrity checking mechanisms to ensure that data have not been tampered with or corrupted during communication.

When the client sends a request to the server, the server responds with a response containing a status number and real-time data, because the technology used by traditional HTTP is the request–response model. MQTT functions have been changed to a subscription system. Although it uses the same TCP/IP structure as HTTP for data transfer, the main change is that device endpoints must first specify a proxy to communicate with, which acts as a bridge between publishers and subscribers. The most important feature of the message exchange is that it defines three QoS (Quality of Service) conditions [9,10]. Among them, QoS 0 is sent at most once, which does not guarantee that the message can be sent successfully; QoS 1 is sent at least once, which guarantees the message can be sent out. QoS 2 is transmitted once, and the message is divided into two parts during transmission to ensure that the message can reach the receiving end, but it may also cause a higher delay. Due to its characteristics of occupying less bandwidth, better power-saving performance, and more extensive data transmission capacity, this technology is currently widely used on the Internet of Things communication environment.

The MQTT architecture uses the Publish/Subscribe mode. So, we can allow multiple publishers to publish messages for different topics. By subscribing to a theme, an attendee can receive relevant notifications according to different themes. The intermediary conversion tool, the broker, does not manage the encryption operations of each subscriber. It only forwards the message to all subscribers who have subscribed to the topic, as shown in Figure 1 above. A subscriber can receive different notifications by subscribing to other topics and stop receiving messages related to a topic by unsubscribing.

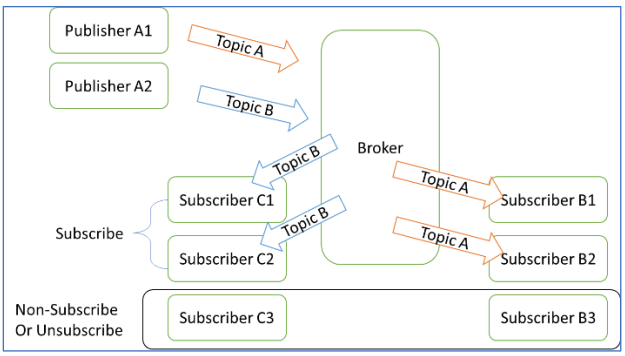

Figure 1. The message in MQTT broker structure.

Therefore, from the above architecture, we expect to encounter the first technical bottleneck: if different subscribers subscribe to the same topic, it means that the topic's message information can be stolen and monitored. If the subscribers themselves publish messages on the same topic, this will cause the information that initially needed to be sent to be duplicated and replaced. For this reason, even though MQTT provides TLS-encrypted events, the MQTT client will use the TLS protocol to encode the message when multiple subscribers/publishers subscribe to notifications on the same broker [11]. The architecture, where the MQTT server will use TLS to decrypt the message and publish it to all clients that have subscribed to the corresponding topic, is still insufficient to filter out subscribers using the same topic to avoid legitimate monitoring of message sending and delivery.

2.2. IoT Agent Design

Edge computing nodes can perform effective computing and event measurement operations through the management and control of related devices in IoT devices. In the traditional IoT model, edge data can be collected on the node in real time through the node's leading core computing and data acquisition by computing elements [12]. Software agents are designed to help process a node's digital data: storage and analysis. However, there are still some limitations in the data storage and analysis of the node data. In most studies, the task of nodes is to transmit relevant digital information. Still, for data analysis and subsequent event processing operations, related research by Chang (2023) [13] can find that edge computing devices can perform event processing and compare number signal values, reducing the technical difficulty of IoT power consumption and packet transmission overload.

All nodes in this article are endpoints for discussing the Internet of Things, called device sides. The software agent performs a blockchain-based critical exchange method to perform encryption and decryption exchange on devices installed on the IoT endpoint, so the agent serves the terminal application installed on the node.

Due to the weakness in the MQTT communication protocol, packets are tampered with during the transmission process, and data messages are captured, resulting in security vulnerabilities. Both the sender and the receiver have similar problems, so through the agent system like the software middleware, we can perform message event management before sending and receiving packets, and the agent will change the data content of digital information for encryption and decryption, thereby reducing the technical difficulties of message packets being interfered with and tampered with.

The design of IoT software agents has a vital role in assisting effective communication and data exchange between IoT devices and platforms. An agent's main functions include protocol conversion, data format conversion, and security and authentication. Because the IoT agent designed through software can convert protocols, it can understand and process multiple communication protocols [14]. This feature enables different devices to communicate seamlessly with the IoT platform, even using other communication protocols, achieving device interoperability. In response to the heterogeneity of the obtained sensing data, IoT agents can also convert data formats to ensure that data from different IoT devices are presented in a unified design, which can assist in the management and data analysis of the IoT platform [15]. Therefore, an IoT agent overcomes the challenges brought by the diversity of IoT devices and communication protocols and provides technical support for IoT systems' safe and efficient operation. Therefore, we use agents to build a secure conversion event recording service. We use the same agent design to communicate cryptographically verified device identification and authorization mechanisms to ensure that only authorized devices can communicate with the IoT platform. This security design not only helps prevent potential threats but also provides the stable operation of the IoT environment.

2.3. Real-Time Quality Control

There are two levels of work in IoT device communication; one is the transmission task. In the case of a water valve switch in a factory, it is a digital signal that does not

activate the water valve. When the device receives the start signal, it starts and closes the water valve. The other is a data signal that continuously returns relevant sensor status information through the network, such as the temperature generated by machine operation. When transmitting digital data, the digital signals released by the publisher are actively and continuously sent due to the characteristics of the MQTT communication protocol [16]. However, the physical data signals are continually sent back. Taking an intelligent factory as an example, digital signal transmission in the Internet of Things will affect the automatic production of the factory. When a fully automated sensing and control device must transmit data accurately to avoid affecting operation and control results, we must design a more secure data transmission method. The status of the production line, for example, the switching status of the water valve, will affect the cooling progress of the production line machine. We cannot let the machine stop production due to overheating. Therefore, the quality of communication messages exchanged through the Internet of Things cannot be subject to external tampering, which will affect the security and service quality of the entire service [17]. To solve similar problems, we have developed a set of agent service designs belonging to edge computing from Hsu's previous research (2020) [18]. In this study, to solve the problem of fixed-cycle backhaul patterns of endpoint devices in the Internet of Things, an intelligent computing frequency detection algorithm was constructed to calculate and evaluate the fixed-cycle detection information generated by the same node using the backward algorithm. This method reduces the high network communication cost caused by the devices using fixed-cycle backhaul. This method verifies the value generated by the sensing device before the device endpoint returns the message, thereby reducing error in the message return. Based on the agent's assistance in monitoring the data transmission of the Internet of Things, data errors can be reduced, and a monitoring mechanism for service-quality assurance can be provided. From the research results, one of the characteristics of edge computing is the ability to transmit information data based on physical principles through digital transmission. However, when the error message data are generated in this digital transmission mode, it is also sent back to the broker simultaneously. A software agent will automatically connect the wrong work item event due to an abnormality in the transmitted digital signal. Therefore, the agent's verification of the data generated by the packet will be a critical technical issue. It cannot just forward the digital call; it must read and verify the digital signal and compare it with the data received the previous time. We need to reduce the impact of erroneous message packets on subsequent calculations and judgments.

Since the received message can be inspected and audited by the edge service agent, a live interactive verification is performed each time the data are obtained [19]. The process can improve the accuracy of the retrieved data. Each node performs the physical data retrieval and sending. A data correctness check is carried out simultaneously with the transmission. This method can effectively improve the problem of incorrect data, which we are concerned about. However, after the data are corrected and checked by the edge agent [19], the second issue that we need to be concerned about is how to exchange the correct data with other data collectors or servers in an accurate way.

Based on the above considerations and the review of related literature, it is found that the correctness of data processing is an essential technical issue in the service quality monitoring of IoT [20]. When using service agent tools to support IoT service quality monitoring and ensuring the convenience of packet delivery on the communication link, it is also necessary to consider whether the data can be accurately delivered to the destination. Since the current communication method of MQTT broadcast is mainly used to facilitate data transmission, communication registration is used to perform simple operations to authenticate the connection. In the broadcast packet, there is no framework for securing the link to the package message. Based on IoT communication, broadcast is a continuous and long-term information exchange, a relatively risky communication channel for managing network security [8,21,22].

Assuming that the packet is monitored for a long time, we can observe that when the authentication of the broker connection is tested successfully, illegal numerical results that change the signaling transmission can be sent [23]. Let us take the above example of the water valve, when the temperature reaches 25 degrees Celsius. The IoT device will start the water valve to lower the temperature of the machine. After the intrusion, the temperature value received by the broker is illegally modified. Each datum is reduced by 5 degrees when it is returned. Therefore, activating the water valve to cool down at 30 degrees will result in a severe technical bottleneck and loss when the machine temperature reaches its true temperature. Since traditional computers and smartphones can use VPN (Virtual Private Network) mode to reduce possible data theft and tampering problems during data transmission, IoT device endpoint technology does not yet have good communication channel management that can reduce the obstacles to packet signal theft. Therefore, how to effectively solve the security technology management of IoT data packet transmission is an important issue.

2.4. Data Encryption Transmission Technology

As the TCP/IP transport layer provides reliable network transmission, packet encryption is not performed in the standard packet state [24]. This ensures that packets can be checked quickly and easily, with the checksum only verifying whether the package is being transmitted. If damage occurs during the process, inspection operations must be carried out. Therefore, in the current network transmission process, we will use secure connection operations such as SSH (Secure Shell Protocol), SSL (Secure Socket Layer), and other associated technologies to perform packet security encryption operations at the application layer. In many network communication applications, a VPN is commonly used to provide comprehensive channel encryption [25]. This system connects the data port at both ends and uses encryption to prevent tampering and eavesdropping during the transmission of information.

However, this method may encounter another challenge. For example, in Figure 2, when the user enters the network in response to the connection, if the communication node of the peer's target is not directly connected to the VPN system, secure communication encryption can be performed through commercially available secure connection channels. However, there is still an information security risk because the data port is the corresponding network port of the VPN. The main concern is that in the environment corresponding to the VPN connection port, the original network packets are not converted and encrypted for secure communication, and the data remain in clear text. It is still possible to extract clear text information using basic tools such as Wireshark.

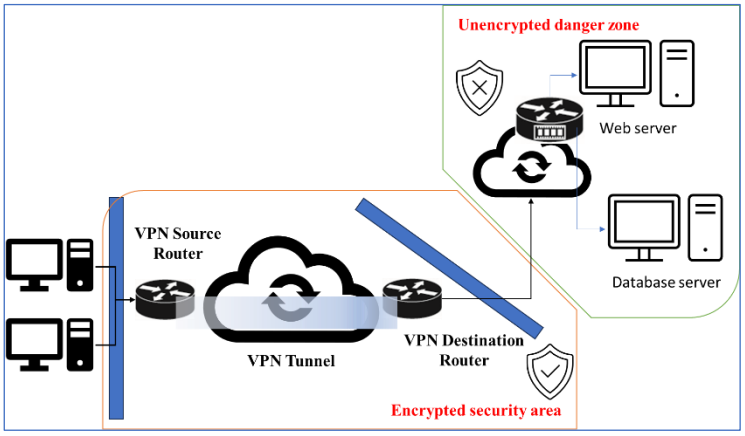

Figure 2. Vulnerabilities in using VPN to encrypt communications.

To effectively address security issues in communication, one can use the application layer system model design. This design involves the sender and receiver performing

encryption and decryption simultaneously in the communication channel environment. Information-hiding operations are established before the data are transmitted, and the data are obtained through software. After the agent identifies the data, they are transferred to the secure encryption operation service. This study utilizes the technical characteristics of blockchain to merge information with the blockchain's channel operation. The software agent synchronizes the block keys of cluster nodes in different regions and establishes the cluster's encrypted channel service. When the IoT sensing device sends and transmits the obtained data, the information is encrypted to complete the data conversion. Unauthorized parties cannot tamper with or distribute the data. Since the key is converted, a secure and dependable information conversion service is constructed.

2.5. Blockchain Technology Improves Security by Recording Events

During the data transmission stage, preventing tampering with data message packets and causing data changes is a dilemma. Blockchain technology solves this problem by having each node in the application network keep a digital ledger containing a list of all transactions performed on the web [26]. Public and private key pairing is used to ensure security. When executing a blockchain transaction, it is recorded as a ledger update. The transaction is encrypted using a unique private key, a digital signature. This means that no node can tamper with transaction data without invalidating its digital signature, thereby rendering the original transaction invalid [27,28]. Therefore, when exchanging IoT data through MQTT, data tampering can be avoided by signing the data using the data encryption feature of block training. If different essential processing can be included for various nodes in the vital part, each node can update the critical management with other nodes through the subscription mechanism. This can eliminate the technical bottleneck caused by standard keys. Since each node has a public key shared with all other nodes, this solves the problem. When a node receives a ledger update message from another node, it verifies it using the public key corresponding to the initiating node. If the ledger has not been tampered with, the verification is successful, and the node updates its catalog copy. This ensures that every transaction is propagated through the blockchain network and becomes part of the ledger stored at each network node.

The main objective of blockchain technology is to utilize smart contracts to establish secure communication channels and record events, thereby addressing security and privacy concerns that arise from decentralized data exchange. This technology can offer immutable distributed data when stored, making it highly suitable for safeguarding critical data. The research considers that when using MQTT's broadcast characteristics, any IoT device must authenticate the connection with the broadcast center to ensure that the subscriber and sender have synchronized data information. This allows for the integration of data before exchanging it. The application of blockchain technology builds a secure communication encryption framework. The blockchain security agent integrates edge computing into a fast encryption framework to perform group operations. This strengthens the security of large-scale data transmission and storage under the Internet of Things.

3. Experimental Structure and Description

It is necessary to design a solution to address the transformation of communication information data systems due to the current practical limitations of network communication technology. In our research, we have developed a novel framework that includes a blockchain security agent service model. This model implements encrypted message conversion at the application layer level through an agent, ensuring that all data expected to be transmitted by the sensing device undergo the necessary critical conversion process. Therefore, we redesigned a communication encryption application model based on the upper layer of TCP/IP to improve the security and performance of the overall system. This research applies a blockchain security agent service model to handle the conversion of communication information, ensuring data integrity and confidentiality. Through agent intervention, we can achieve a higher standard of security. When dealing with sensitive

data, it is essential to avoid multiple reads or tampering, as this may affect the accuracy of the data. In addition, we have designed an approach that can be applied in different network environments, including blockchain methods with additional permissions. Smart contracts record the status and process information of all key exchanges.

In this study, the communication mechanism of the Internet of Things (IoT) system is thoroughly investigated in the connection environment of the communication layer. The primary mechanism mainly depends on the broker, which facilitates the services of registering and subscribing. Once all the edge nodes are registered, by subscribing to the same topic they can access the various data information provided by the endpoints of different devices. A notable feature of this mechanism is that when a node sends a message, all the subscribers will receive the same content information simultaneously.

We have redesigned the critical exchange model based on a standard communication architecture to address security concerns. Each node is responsible for creating its own public and private keys, emphasizing active key management, and preventing data leakage through unauthorized connections. In addition, we assume that each IoT device has registered with the Certificate Authority (CA) server and obtained its key. Public critical information is broadcasted to all nodes using the MQTT protocol.

This process ensures the security of the key. It also establishes a reliable and secure channel for node communication. The public key-based communication encryption mode helps protect the integrity and confidentiality of data, which further enhances the overall security of the system. In this study, we focus on the security issues at the communication level of the Internet of Things, and the design plan is as follows in Figure 3.

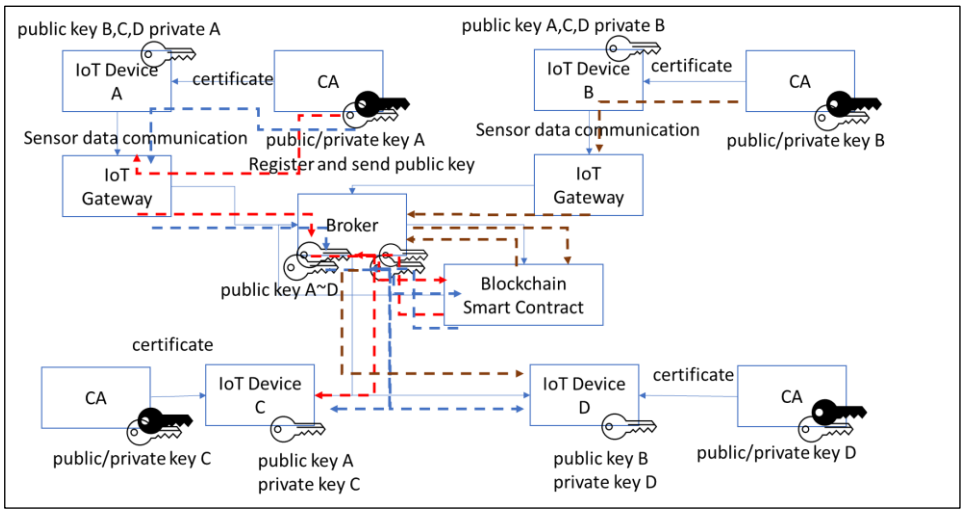

Figure 3. Experiment on key exchange architecture based on broker.

In this study, we changed the system design. We no longer relied solely on MQTT to register hardware devices but instead subscribed the hardware devices to the communication intermediary environment. Therefore, we will distribute the keys required by all nodes with the broadcast environment instead of just registering through a single MQTT channel. In our research, we handle key delivery and message delivery separately. We only provide subscriptions for specific topics (such as key-type notes) when subscribing to messages. Before communication between endpoints, keys will be exchanged, and pre-deployment will be completed before contact. The design of this offload operation ensures the security of communication and the correct transmission of keys between endpoints. This is a more effective and secure communication mechanism.

According to our design architecture in Figure 3, the packets of the key will first pass through the red-dotted radio operation. We assume that node A generates a pair of public and private keys and broadcasts the public key. However, the public key cannot be successfully retrieved in the case of node D, a non-gold participant.

To enhance non-one-way transmission, we added an encryption key to node B, enabling node D to subscribe to our experimental architecture. According to the certification plan of the key, the public key that provides service vouchers supports different subscribers to synchronize with the corresponding node.

Based on our plan, we designed procedures for the experiment to ensure reliable information transmission during communication:

(1) Establish MQTT communication rules between IoT devices and applications and set up an MQTT broadcast agent to handle releasing and subscribing messages.
(2) Write a smart contract to process data sent by MQTT devices and verify its effectiveness before writing it to the blockchain.
(3) Configure communication between the MQTT nodes. To connect the MQTT client to the MQTT agent and publish data on the Internet of Things device, ensure that the node side can format data into the required format for smart contracts.
(4) To complete the interactive transmission of the key, combine it with the blockchain. The MQTT client must be capable of sending data to the blockchain smart contract.

4. Results

In the simulation dataflow design, as shown in Figure 4, the communication process commenced with an attempt to transmit a message through the Raspberry Pi 4 (Wales, UK), intending to relay an IoT message to a specific node. Upon the node's reception of the message, all nodes that had subscribed to the same topic were notified. The blue-dotted line in the illustration symbolizes the data processing sequence. Given that MQTT operates by subscribing through a shared theme, the message is dispatched to nodes subscribed to the topic, such as nodes B, C, and D. Upon the message's arrival, it becomes apparent that only node C possesses the key assigned by node A.

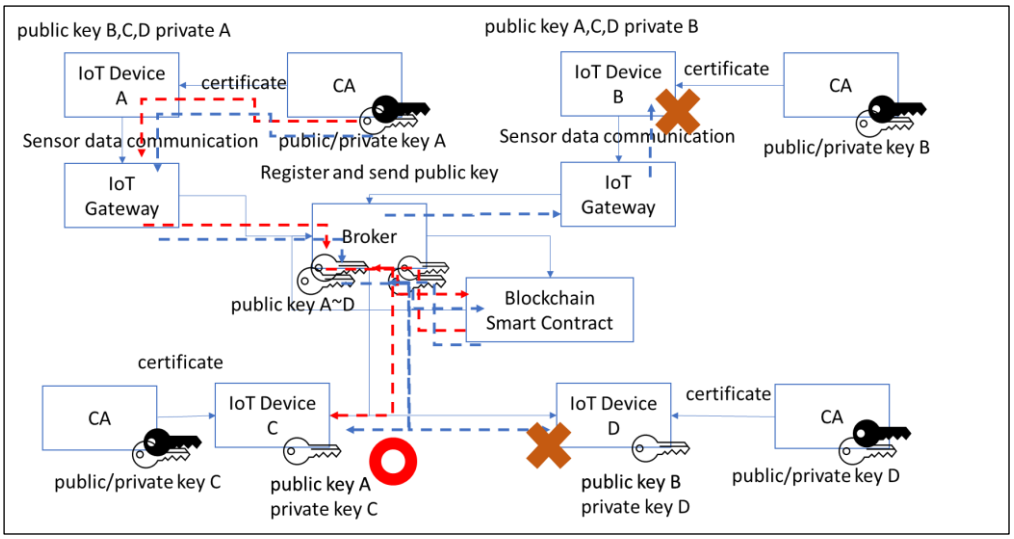

Figure 4. Simulation message data transfer operation.

Consequently, nodes B and D cannot access the corresponding key, rendering the message unreadable and unidentifiable to these nodes. This mechanism ensures that data privacy is maintained, and only authorized nodes with the requisite key can interpret the transmitted messages, enhancing the overall security of the communication process.

The experiment began by setting up a Certification Authority (CA) system that automatically generates golden keys. Once a key had been generated for each node, it was sent to the MQTT broker along with the subscription topic and the submission of the key.

The four edge devices, 1111, 1112, 1113, and 1114, in Figure 5, were created by simulation. An MQTT broker server was also established to facilitate connection registration for these four nodes. The broker provided a preview, as shown in Figure 4. Four nodes suc-

cessfully synchronized their connections, ensuring the effective generation and distribution of keys and successful connections between nodes and broker.

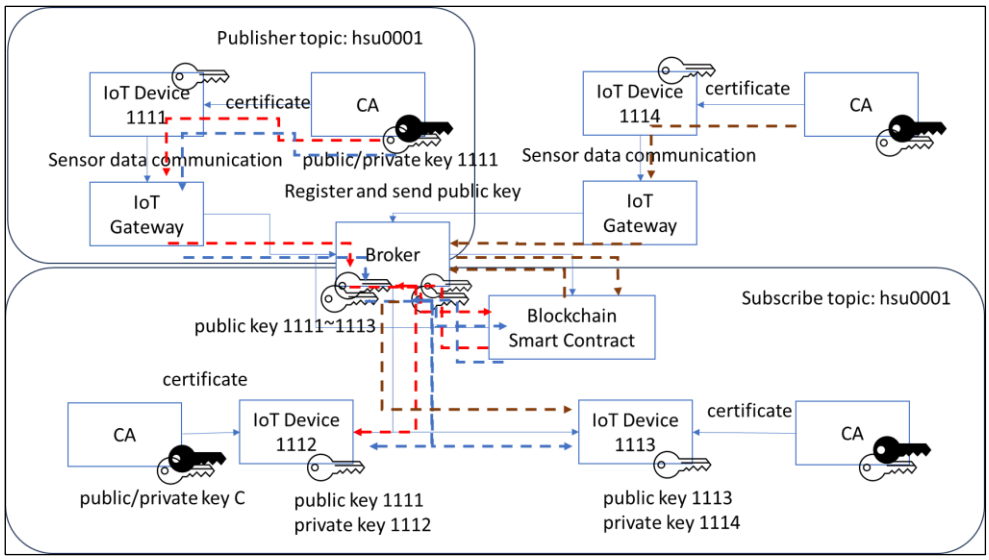

Figure 5. Experimental experiments 1 to 4 in total, representing the representative entity.

To maintain certificate consistency, all node codes implement a CA certificate generator. This tool simultaneously generates a public and private key, as shown in Figure 5. When sending a message from node 1111, nodes 1112 and 1113 receive the MQTT message simultaneously. However, since 1112 has the public key of 1111, it can analyze the information, while 1113 cannot, due to encryption. The system's intelligent contract screenshot in Figure 6 shows that nodes 1111 to 1114 automatically generate the CA voucher and broadcast the public key voucher. Each node encrypts the message using its private key. For instance, using blockchain technology, node 1112 generates an event record of its location and private key for intelligent contracts. With our designed architecture, we can guarantee the accurate generation and distribution of vouchers while ensuring safe and reliable communication between nodes and smart contracts.

Figure 6. The key information is automatically generated for the four nodes.

We can observe that after the node automatically generates the encryption key, it transmits it back to the broker through MQTT while the broker is registered. Figure 6 shows the result of the broker receiving the key returned from different nodes in red. The box shows four other nodes generating keys to complete the backhaul operation.

We will explain the communication certificate generated by node 2. When the node opens the connection, the system immediately generates the blockchain intelligent contract number and public key information, as shown in Figure 7, No. 1. We send the message to the broker by the agent. The delivery format information is (number/key), and the node simultaneously generates the corresponding private key. Like No. 2, the data are automatically encrypted during data transmission.

Figure 7. Display the broadcast registration number and private key of node 1112.

In the experiment, we automatically encrypt the file to verify whether the key reaches the MQTT message exchange state. We follow the architectural environment of Figure 5. When node 1111 broadcasts the data to the broker, and if 1112 and 1113 register and subscribe at the same time for the same topic, the message sent by 1111 will be obtained by two nodes, 1112 and 1113. Since our domain first divides the group to obtain the key, node 1112 has the public key of 1111, while 1113 does not have the public key of 1111. After the message is sent, the experimental results are shown in Figure 8. Node 1112 can parse the encrypted subscription message sent, but 1113 cannot parse and parse out the incorrect authentication information that failed to decrypt.

Figure 8. Status results for encrypted subscriptions.

To experiment with the idea that group key exchange can enable key exchange, we set 1112 and 1113 as the same group. We adjusted the configuration settings. We reset the topic through the broker, exchanged the 1111 key message to the 1113 node, and constructed the build. As shown in Figure 9 below, the provided node 1113 has both 1111 and 1114 encryption keys. The results in Figure 10 show that node 1113, shown on the right, can successfully read the encrypted message frame information after the message is re-sent.

As shown in Figure 11, the study examined the state of information exchange in the experiment using blockchain contract data 1113. In the architecture environment shown in Figure 9, the third node of 1113 contains both hsu0001 and hsu0002 groups in the subscription public key obtained through grouping. This allows 1113 to exchange data across groups between 1111 and 1114 and record the key exchange event in the blockchain contract.

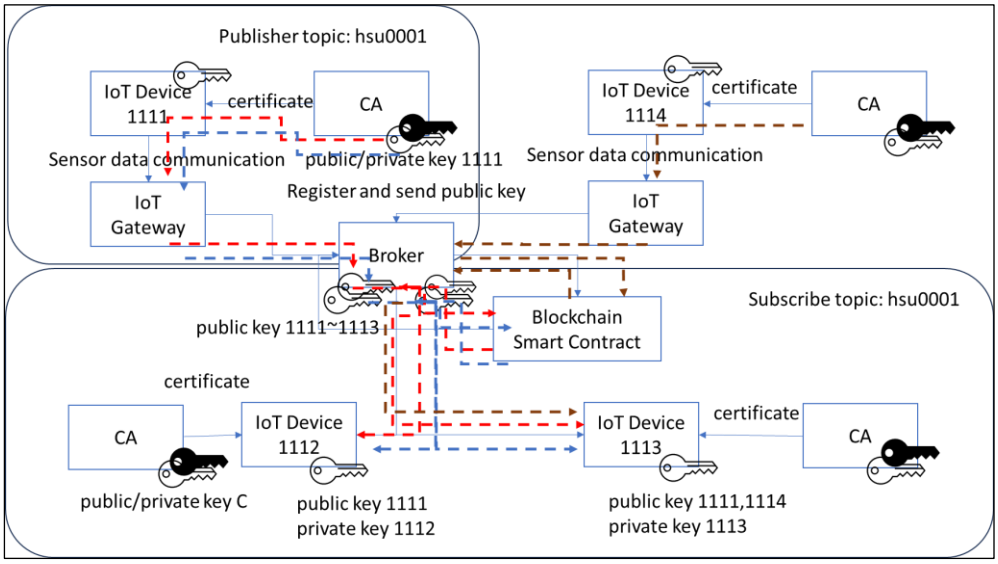

Figure 9. Added 1111 key broadcast to provide node 1113 with three keys.

Figure 10. Update subscription group message delivery results.

The study utilized a broker to verify message exchange. Within the broker system, the core system successfully established a group operation. Nodes 1112 and 1113 are subscribed to the same hsu0001 topic group. When node 1111 sends a message to the hsu0001 topic, nodes 1112 and 1113 can complete the task of decrypted reading. It is assumed that the hsu0002 group exists, and only authorized users in group 1113 have decryption capabilities. Although other subscribers can subscribe to the topic, they cannot access the data information because they are not part of the object group. Refer to Figure 12. Based on our experimental encrypted message transmission results, the encrypted broadcast topic 'hsu0001' sent by 1111 had two nodes complete synchronous registration, while the encrypted broadcast topic 'hsu0002' sent by 1114 underwent synchronous registration only by 1113. Thus, if the service process sends messages from 1114, only nodes 1113 and 1114 can communicate, while nodes 1111 and 1112 cannot receive them. This is illustrated in Figure 13 of the experimental results.

Figure 11. Node 1113 blockchain contract status.

Figure 12. Group subscriber passes encryption key.

Figure 13. The reverse packet 1112 sent by 1114 could not be read.

The Wireshark network package analysis tool is utilized to inspect packets. Duplicate packets are automatically converted to encryption technology by the message agent based on the blockchain proposed by our research. When any message is captured through the network, the dataset bits of the packet can be observed. Security has been enhanced. In Figure 14A, it can be observed that the data field information of the TCP/IP packet is encrypted and cannot be parsed. In contrast, Figure 14B shows that when using MQTT for data exchange, the generated message information can be easily interpreted.

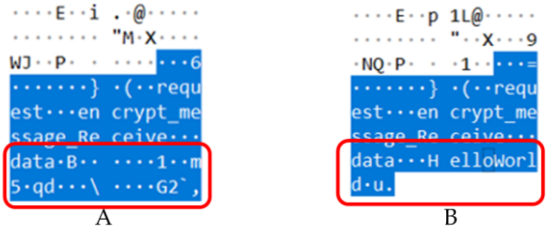

Figure 14. (**A**). MQTT broker with message encrypt (**B**). Original message data information.

5. Discussion

The Internet of Things system adopts a registration mode for the communication protocol. This allows different IoT devices to subscribe to each other and share message tasks, such as power and water valve switches. However, issues may arise when only the subscription task is performed. During the transmission, agents or subscribers may record and monitor data packets due to the TCP/IP-based transmission operation. This can lead to information security issues, such as hackers impersonating nodes to perform publishing operations and invade the system, which can further damage the stability of the IoT system.

Our study assigned different subscribers to the group that created the topic. Even though non-authorized subscribers can complete the subscription through MQTT, the retrieved data information remains uninterpretable because the non-publisher has granted the right to the public key. The experimental verification results show that when the connection is forcibly obtained, the data cannot be effectively interpreted and used, as shown in Figures 8 and 13.

During our research, we examined the material format obtained by the broker and the subscriber. We discovered that the subscribed contract address and key format can be recorded through the contract formulation of the blockchain. However, other nodes cannot obtain the publisher's private key, ensuring security and preventing denial. At the same time, the private key-generated information is also bound to the contract address. Therefore, nodes in the group that obtain subscription information can only access the public key through group subscription. Decrypting the data packets retrieved by itself provides a secure method for exchanging communication data packets at the network transport layer.

6. Conclusions

This study establishes data key exchange rules using blockchain data transfer record logs. These rules construct data information under the Internet of Things communication mode into a multi-communication topic subscription channel, allowing for data exchange through different channels. Data encryption technology ensures non-repudiation and makes it difficult to tamper with the rules.

As a contribution to our research, we propose a secure IoT communication model that can prevent illegal acquisition and control of related service status during network data exchange. Our model was tested through interactive experiments, which confirmed that data cannot be obtained or controlled through any encrypted channel technology other than blockchain. The research findings indicate that messages sent by different MQTTs were successfully encrypted and decrypted using a mechanism that employs public and private keys. The MQTT channels can also be parsed directly by targeting the same subscription topic. This research establishes rules to improve the accuracy and reliability of data collection and prevent unauthorized access to information transmitted by the Internet of Things.

We initially limit the number of allocated nodes to propose an innovative communication encryption operation. This is to prevent excessive subscription rules that could reduce the effectiveness of data encryption and decryption. In the study, we tested and established experimental regulations based on the simulation of four unit nodes. We suc-

cessfully cross-encrypted messages of different subscription topics, avoiding any data loss. In future experiments, we will design a performance test plan for multi-node data frame encryption and decryption to monitor the processing performance of relevant hardware. We will also include the calculation performance of communication channel encryption in subsequent research.

Funding: This paper was partially supported by the National Science and Technology Council of the Republic of China under contract NSTC 112-2221-E-031-005-.

Data Availability Statement: Data is included within the article.

Conflicts of Interest: The author declare no conflict of interest.

References

1. Mishra, B.; Kertesz, A. The use of MQTT in M2M and IoT systems: A survey. *IEEE Access* **2020**, *8*, 201071–201086. [CrossRef]
2. Paris, I.L.B.M.; Habaebi, M.H.; Zyoud, A.M. Implementation of SSL/TLS Security with MQTT Protocol in IoT Environment. *Wirel. Pers. Commun.* **2023**, *132*, 163–182. [CrossRef]
3. Patel, C.; Doshi, N. A novel MQTT security framework in generic IoT model. *Procedia Comput. Sci.* **2020**, *171*, 1399–1408. [CrossRef]
4. Al Enany, M.O.; Harb, H.M.; Attiya, G. A Comparative analysis of MQTT and IoT application protocols. In Proceedings of the 2021 International Conference on Electronic Engineering (ICEEM), Menouf, Egypt, 3–4 July 2021; pp. 1–6.
5. Spielvogel, K.; Pöhls, H.C.; Posegga, J. TLS Beyond the Broker: Enforcing Fine-Grained Security and Trust in Publish/Subscribe Environments for IoT. In Proceedings of the Security and Trust Management: 17th International Workshop, STM 2021, Darmstadt, Germany, 8 October 2021; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 145–162.
6. Prantl, T.; Iffländer, L.; Herrnleben, S.; Engel, S.; Kounev, S.; Krupitzer, C. Performance impact analysis of securing mqtt using tls. In Proceedings of the ACM/SPEC International Conference on Performance Engineering, Virtual Event, 19–23 April 2021; pp. 241–248.
7. Kadhim, O.N.; Ketab, A.S.; Obaid, A.J.; Albermany, S.A.; Raheem, A.R.; Hussien, N.A. Simulation Secure MQTT Protocol Based on TLS in IoT-Fog Computing Environment. In Proceedings of the Fourth Doctoral Symposium on Computational Intelligence, Lucknow, India, 3 March 2023; Springer Nature: Singapore, 2023; pp. 13–21.
8. Hintaw, A.J.; Manickam, S.; Aboalmaaly, M.F.; Karuppayah, S. MQTT vulnerabilities, attack vectors and solutions in the internet of things (IoT). *IETE J. Res.* **2023**, *69*, 3368–3397. [CrossRef]
9. Ruzzier, F. Adaptive Quality of Service for MQTT-SN. Ph.D. Thesis, Politecnico di Milano, Scuola di Ingegneria Industriale e dell'Informazione, Milano, Italy, 2022.
10. Liu, Y.; Al-Masri, E. Slow Subscribers: A novel IoT-MQTT based denial of service attack. *Clust. Comput.* **2023**, *26*, 3973–3984. [CrossRef]
11. Xiong, C. Secured System Architecture for the Internet of Things Using a Two Factor Authentication Protocol. Ph.D. Thesis, University of Ottawa, Ottawa, ON, Canada, 2020.
12. Alexakis, G.; Panagiotakis, S.; Fragkakis, A.; Markakis, E.; Vassilakis, K. Control of smart home operations using natural language processing, voice recognition and IoT technologies in a multi-tier architecture. *Designs* **2019**, *3*, 32. [CrossRef]
13. Chang, D.M.; Hsu, T.C.; Yang, C.T.; Yang, J. A Data Factor Study for Machine Learning on Heterogenous Edge Computing. *Appl. Sci.* **2023**, *13*, 3405. [CrossRef]
14. Al-Masri, E.; Kalyanam, K.R.; Batts, J.; Kim, J.; Singh, S.; Vo, T.; Yan, C. Investigating messaging protocols for the Internet of Things (IoT). *IEEE Access* **2020**, *8*, 94880–94911. [CrossRef]
15. Aminizadeh, S.; Heidari, A.; Toumaj, S.; Darbandi, M.; Navimipour, N.J.; Rezaei, M.; Talebi, S.; Azad, P.; Unal, M. The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things. *Comput. Methods Programs Biomed.* **2023**, *241*, 107745. [CrossRef] [PubMed]
16. Fahim, M.; El Mhouti, A.; Boudaa, T.; Jakimi, A. Modeling and implementation of a low-cost IoT-smart weather monitoring station and air quality assessment based on fuzzy inference model and MQTT protocol. *Model. Earth Syst. Environ.* **2023**, *9*, 4085–4102. [CrossRef]
17. Rizzardi, A.; Sicari, S.; Coen-Porisini, A. Analysis on functionalities and security features of Internet of Things related protocols. *Wirel. Netw.* **2022**, *28*, 2857–2887. [CrossRef]
18. Hsu, T.C.; Yang, H.; Chung, Y.C.; Hsu, C.H. A Creative IoT agriculture platform for cloud fog computing. *Sustain. Comput. Inform. Syst.* **2020**, *28*, 100285. [CrossRef]
19. Joshi, P.; Kalita, A.; Gurusamy, M. Reliable and Efficient Data Collection in UAV-based IoT Networks. *arXiv* **2023**, arXiv:2311.05303.
20. Liu, C.H.; Liang, W.Y.; Tsai, M.Y.; Chang, W.C.; Chen, W.K. A Novel Approach to Automate IoT Testing of Gateways and Devices. *J. Inf. Sci. Eng.* **2022**, *38*, 317.
21. Boppana, T.K.; Bagade, P. Security Risks in MQTT-Based Industrial IoT Applications. In Proceedings of the 2022 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain, 1–3 August 2022; pp. 1–5.

22. Yuan, B.; Song, Z.; Jia, Y.; Lu, Z.; Zou, D.; Jin, H.; Xing, L. MQTTactic: Security Analysis and Verification for Logic Flaws in MQTT Implementations. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2024; p. 13.
23. Diro, A.; Reda, H.; Chilamkurti, N.; Mahmood, A.; Zaman, N.; Nam, Y. Lightweight authenticated-encryption scheme for internet of things based on publish-subscribe communication. *IEEE Access* **2020**, *8*, 60539–60551. [CrossRef]
24. Shilpa, V.; Vidya, A.; Pattar, S. MQTT based secure transport layer communication for mutual authentication in IoT network. *Glob. Transit. Proc.* **2022**, *3*, 60–66.
25. Abbas, H.; Emmanuel, N.; Amjad, M.F.; Yaqoob, T.; Atiquzzaman, M.; Iqbal, Z.; Shafqat, N.; Bin Shahid, W.; Tanveer, A.; Ashfaq, U. Security Assessment and Evaluation of VPNs: A Comprehensive Survey. *ACM Comput. Surv.* **2023**, *55*, 1–47. [CrossRef]
26. Zhu, P.; Hu, J.; Li, X.; Zhu, Q. Using blockchain technology to enhance the traceability of original achievements. *IEEE Trans. Eng. Manag.* **2021**, *70*, 1693–1707. [CrossRef]
27. White, R.; Caiazza, G.; Cortesi, A.; Im Cho, Y.; Christensen, H.I. Black block recorder: Immutable black box logging for robots via blockchain. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3812–3819. [CrossRef]
28. Zubaydi, H.D.; Varga, P.; Molnár, S. Leveraging Blockchain Technology for Ensuring Security and Privacy Aspects in Internet of Things: A Systematic Literature Review. *Sensors* **2023**, *23*, 788. [CrossRef]