Journal of Computers Vol. 31 No. 3, 2020, pp. 275-288
doi:10.3966/199115992020063103022

Data Privacy-Preserving of Consortium Blockchain in the Internet of Things

Yechen Wang[1*], Haibing Mu[1], Yingsi Zhao[2]

[1]Beijing Key Laboratory of Communication and Systems, School of Electronic and Information Engineering, Beijing Jiaotong University, Haidian District, Beijing, China
Ycwang17@bjtu.edu.cn, hbmu@bjtu.edu.cn

[2] School of Economics and Management, Beijing Jiaotong University, Haidian District, Beijing, China
yszhao@bjtu.edu.cn

Received 28 March 2020; Revised 1 April 2020; Accepted 28 April 2020

Abstract. Blockchain has the characteristics of tamper-proof, traceability and trusted interaction, which can make up for some existing defects of the Internet of Things (IoT), such as single point of failure and data out of control caused by centralized management. The blockchain Internet of things (BIoT) has become a hot research field. However, the introduction of blockchain in the IoT scenario has also brought new privacy issues, including the disclosure of user data and transaction records. This paper firstly summarizes the role of blockchain in the current Internet of things scenarios, and proposes a BIoT model applicable to most Internet of things scenarios. Then, on this model, the corresponding privacy-preserving mechanism is proposed for the two application scenarios of micropayment and data interaction. Finally, the validity and privacy of the proposed mechanisms are verified by mechanism analysis and experiments, and the experimental results show that the proposed mechanisms have a short algorithm execution time and only generates a small amount of data.

Keywords: blockchain, Internet of Things (IoT), Pederson commitment, privacy

1 Introduction

With the emergence of a large number of low-cost, feature-rich devices (such as sensors, RFID, etc.) and the development of various communication media, the IoT has gained tremendous popularity in the past decade [1]. According to the report released by Ericsson, the number of IoT devices connected with cellular network will reach about 18 billion in 2022 [2]. The gradual popularization of the IoT has also brought new challenges. Most of the current IoT device and data management adopt the centralized server mode, so the following problems are inevitably introduced: (1) Centralization. The centralized service model may have a single point of failure, and the unavailability of key central services will lead to the paralysis of the IoT. (2) Lack of security and privacy. Users have little control over the data stored on centralized servers, and sensitive data can be misused. The data lacks traceability and may be tampered with or deleted.

The emergence of blockchain provides a new way to solve the above problems in IoT. Blockchain is a typically decentralized technology which was designed by Nakamoto with Bitcoin in 2008 [3]. Each node in blockchain maintains a local ledger that records all valid transactions in the network and ensures the consistency of the ledger through consensus protocol. Blockchain uses cryptographic tools and distributed storage to protect the transaction data recorded in the block from tampering.

The combination of blockchain and IoT can deal with some defects of the current IoT environment, so as to build a relatively trusted, efficient and reliable IoT environment. Many studies combine blockchain with the IoT to provide trusted storage and data interaction by taking advantage of the feature of

* Corresponding Author

blockchain [4-5]. It is shown in [6] that BIoT can reduce the cost of supply chain management and improve the execution efficiency. Some BIoT applications use blockchain as a digital currency to provide micro-payment for the IoT scenarios. For example, Aitzhan and Svetinovic [7] proposed an energy-trading system for smart grid scenarios, which can purchase and sell energy and record historical transactions.

Most of the research in BIoT field is aimed at solving the combination of blockchain and IoT. But one primary problem with BIoT field is that most of the current research lacks the privacy-preserving mechanism for the information on the blockchain and the data of the IoT. There's a lot of research on IoT privacy or blockchain privacy, but only a few studies have shown that BIoT privacy should be a major concern, and no specific solutions have been proposed. The transaction information in the blockchain is public, which means that any node in the blockchain network can obtain it. At present, the transactions of most blockchain applications is in plaintext, which will disclose the address and transaction contents of both parties. The IoT data in many scenarios have high privacy requirements, such as the patient medical data in electronic medical scene, transaction information in smart grid scene, etc. A new approach is therefore needed for protect the privacy of BIoT scenarios.

The main goal of this work is to design a privacy-preserving mechanism for BIoT applications. The contributions of this paper are presented as follows: Firstly, according to the current research in the field of BIoT, this paper divides the application scenarios of BIoT into two categories: micro payment and data interaction. Then, a blockchain model is proposed for the BIoT and a special account model is designed for the BIoT scenarios, and the transaction process of the two scenarios are described. After that, for the micro payment scenario, the payment balance and the identity of the trader are hidden through Pederson commitment, range proof [8], and commit merge etc. to protect the privacy of the transaction process; IoT data is hidden through encryption, transaction credentials and arbitration for data interaction scenarios, which provides greater security while protecting privacy. Finally, the experimental results prove that the proposed privacy-preserving mechanism can provide high privacy for BIoT scenarios. Compared with the results of similar literatures, the proposed mechanism has a shorter algorithm execution time and generates a relatively small amount of data.

Section II introduces the related work. Section III describes the blockchain model proposed in this paper. Section IV proposes the privacy-preserving mechanism of micro-payment and data interaction scenarios. Section V analyzes the security and privacy of the mechanism proposed in this paper. Section VI describes the experimental results to verify the effectiveness and efficiency of the mechanism. The paper summarizes in Section VII.

2 Related Work

The current privacy issues of BIoT can be divided into two categories: identity privacy and data privacy [9].

2.1 Identity Privacy

Identity privacy-preserving refers to hiding the sender and receiver identity information of transactions on the blockchain, and the current identity privacy-preserving mechanism of blockchain mainly includes the following two ways:

CoinJoin is a privacy-preserving measure that mixes multiple transactions and disrupts the original transaction input-output correspondence [10]. F K Maurer et al. define a model for CoinJoin transactions [11], and present an output splitting approach to prevent linking in CoinJoin transactions. However, at least one party knows all the transaction information contained in CoinJoin.

Ring signature can hide the address of the sender of the transaction in a set of addresses, and the blockchain node cannot identify the real sender when verifying the transaction. Malavolta G et al. introduced a new ring signature protocol with anonymity and unforgeability [12], which is based on the NIZK parameter and the returnable random key, but the construction of ring signature is not so efficient.

2.2 Data Privacy

Data privacy-preserving refers to the data generated by BIoT application and the knowledge behind the data.

In the digital currency application of the Internet of things, the data represents the currency balance. Homomorphic encryption can be used to process the ciphertext directly, and the result is the same as that obtained by encrypting the result after processing the plaintext. S.L. Ma et al. proposed an efficient NIZK scheme for account-model blockchain [13], in which the transaction balance is hidden by using homomorphic encryption. Zero-knowledge proof is a popular technology to protect blockchain privacy, which can prove that a party knows a secret without disclosing it. Zerocoin is a digital currency similar to Bitcoin [14], which uses zero-knowledge proof to ensure that the relevant address information of both parties to the transaction is not disclosed. Sasson et al. developed Zerocash on the basis of Zerocoin [15], which uses zk-SNARKs to hide the transaction balance. Zero-knowledge proof technology can effectively protect privacy, but the cost of using it is that it takes a long time to generate and verify proof. Z.S. Guan et al. proposed an efficient privacy-preserving account-model blockchain based on zk-SNARKs [16], which used zero-knowledge proof to hide the account balance, transaction amount and the relationship between sender and receiver.

In the reliability storage application of the IoT, it generally refers to the data generated by the Internet of things devices. The easiest way is to encrypt the data and store it on a blockchain. M. Zhang et al. proposed a practical scheme by using the Identity-Based encryption system [17], which effectively improves the data privacy for non-transaction applications. A more appropriate alternative is to store IoT data in a third-party location, with access control tokens and access records stored on the blockchain. J.T Hao et al. proposed a storage scheme for agricultural products tracking using blockchain and IPFS (InterPlanetary File System) [18]. Ali et al. proposed a multilayer blockchain architecture that uses smart contract to implement access control [19]. This solution uses IPFS as the storage medium for IoT data. The work of [20] used a token-like approach to control the data, but the ciphertext of the data was stored off-chain, and the access permission records were stored in the blockchain. Chen et al. proposed a data management method combining private cloud and blockchain called JointCloud [21]. The private cloud stores IoT data, uses the blockchain to record data transmission and IoT interaction logs, and smart contracts manage transactions and monitor data. This method protects the privacy of the data while increases the maintenance cost of the private cloud server. Dorri et al. proposed a double-chain blockchain privacy-preserving architecture [22], and applied it in the smart home. IoT data is stored in the cloud, while IoT data access history is recorded in the private chain, and users can share IoT data with others through the public chain on the upper layer.

3 System Model

In this section, we first analyze the BIoT scenario and determine a combination method of blockchain and IoT, then design an account structure for this scenario, and finally describe the blockchain model that can be used in the BIoT scenario.

3.1 Basic Architecture for BIoT Scenario

First and foremost, blockchain will not completely change the current IoT architecture. The true combination of blockchain and the Internet of Things should be to connect the blockchain as a functional platform to the current Internet of Things architecture.

The IoT is a scenario that requires access control, in which the information generated by IoT devices cannot be accessed by everyone freely. At the same time, the Internet of Things also requires high scalability. The consortium blockchain is a kind of permission chain, which is participated and jointly managed by several institutions. Only nodes with certificate issued by the certificate authority (CA) of these institutions can join the consortium blockchain network. The IoT is neither a scenario that allows arbitrary connections, nor a completely unscalable fixed scenario, so the consortium blockchain is the most suitable for BIoT.

Although the consortium blockchain limits the “Internet of everything” capability of the IoT inevitably, but we can use a dual-chain approach similar to that described in [22] to add a permissionless public

blockchain overlay network on multiple consortium blockchain. Users interact through public blockchain in the form of consortium blockchain organizations.

Most IoT devices can only afford weak cryptographic tools and cannot execute complex transaction verification processes, because of limited computing and storage resources, so they cannot serve as normal consortium blockchain nodes. In the BIoT scenario, we can choose some IoT edge devices and IoT gateway devices with sufficient resources can participate in the blockchain network as blockchain nodes, while ordinary IoT devices and users can send transaction information to them as clients.

3.2 Account Model for BIoT

Currently, most BIoT application scenarios use blockchain as an access control system or a data interaction medium for IoT data, such as patient data interaction in electronic medical scenarios and data management in smart home. We summarize the functions of blockchain in the BIoT scenario into two types: micropayment and data interaction.

Micropayment uses blockchain as a digital currency system to provide trusted transaction functions for the IoT scenarios. Data interaction generally refers to the transfer of IoT data. As mentioned at the beginning of this section, blockchain is not a pure storage system, so the IoT data should better be stored in other locations, such as the distributed file system IPFS. The "IoT data" in this paper refers to the index of IoT data which is stored in other file systems. Of course, a small number of critical IoT data can also be stored in the blockchain.

This paper will use the account model instead of the UTXO (Unspent Transaction Output) model to implement the above two scenarios. In the micro-payment scenario, the account model is simpler than UTXO, and it is convenient for the development of smart contract. The validity of transactions can be verified by verifying the account information directly. In data interaction scenario, the account model only needs to save the IoT data under the user's account address.

We design a dual-balance account model to solve the privacy problem in the micro-payment scenario. The account model includes two types of accounts: one is public account, which balance is open to the blockchain network in plaintext; the other is private account, and the balance it contains will be hidden in the Pederson commitment. For example, the account A includes public account $Account_{public}$ and private account $Account_{private}$, $Account_{public}$ includes the amount of balance in plaintext, but $Account_{private}$ contains only a string-style commitment. In data interaction scenario, account A has only one data account address: $Account_{data}$.

3.3 Blockchain Model of BIoT

The blockchain system described in this paper aims to provide trusted and highly collaborative decentralized applications for the IoT environment. As the client of the block chain network, the Internet of things device keeps the private information such as the device's private key and certificates locally, and sends the data interaction or micro-payment request to the blockchain network node by way of transaction. IoT gateway devices or some powerful IoT edge devices can be used as blockchain nodes, such as PCs, smartphones, etc. The consortium blockchain nodes includes the CA nodes of the BIoT application organization, which are responsible for certificate management, node discovery, etc. The blockchain model of BIoT scenario is show in Fig. 1.

4 Design of Privacy-Preserving Mechanisms

This section will design corresponding privacy-preserving mechanisms for the two scenarios of micropayment and data interaction. In the micro payment scenario, we focus on the privacy of the identity of the two parties to the transaction and the privacy of the transaction balance; in the data interaction scenario, we mainly provide privacy for IoT data.

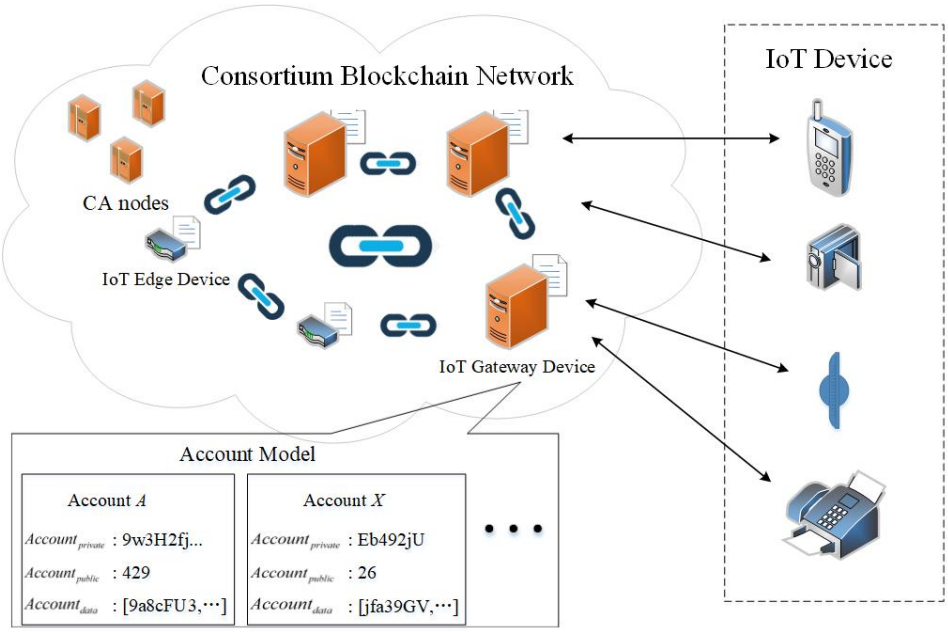

Fig. 1. Blockchain model for BIoT

4.1 Micro-payment

Micro-payment refers to scenarios in which IoT users purchase data with digital currency. In the consortium blockchain based on account model, each user can create their own balance account and save tokens for micro-payment.

As described in section 3, the dual-account model for micro-payment in this paper. Double accounts are not mandatory, users can choose to create only one public account, if there is no need to hide the balance and other information. BIoT users can choose to use public account for public transactions, and the balance of the transaction can be known to all users in the network. In order to ensure the privacy of the transaction, BIoT users can choose to use a private account for private transactions, and the balance of the transaction will be hidden. At the same time, users can transfer balances from public accounts to private accounts or redeem balances from private accounts to public accounts. The BIoT micro-payment process is shown in Fig. 2. This scheme is composed of five polynomial-time algorithms: $< CreateAccount, Exchange, Redeem, Send, Merge >$.

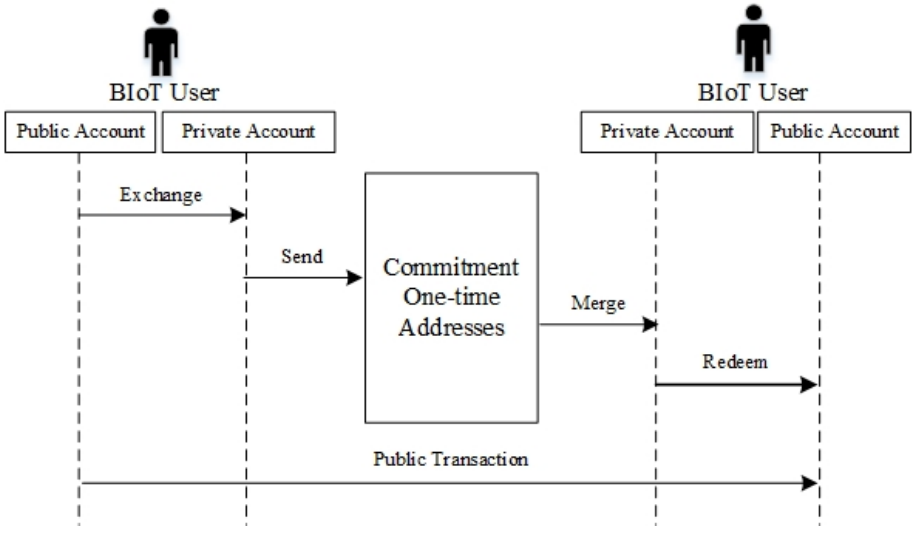

Fig. 2. The schematic illustration of BIoT micro-payment

CreateAccount. This algorithm is executed by blockchain user, and it will generate account address, public and private key pairs for user accounts. In this paper, the elliptic curve algorithm is selected as the user key generation and signature algorithm. The points G and H are the base points of the curve $secp256k1$. First, the user calculate the private key of public and private account sk_p, sk_s using local random entropy seed, and calculates the public key: $pk_p = sk_p * G, pk_s = sk_s * G$. After that, this algorithm will calculate the public key hash and encode it to obtain the address $Addr_p, Addr_s$. This is the same as the bitcoin address generation algorithm [3]. The two addresses obtained by this algorithm will be used as the storage address of the user's balance. The balance of the private account can only be obtained through private account transfer or public account exchange, and the balance of the private account can be redeemed back to the public account.

Exchange. The algorithm is executed by the user's public account to convert the public balance into the private balance. The user saves the public account's private key sk_p locally and publishes the public key pk_p. The user selects the balance to be converted and this algorithm inevitably discloses the balance of conversion to the blockchain network. Suppose the user will exchange v balance, then it will select a random number r and build a Pederson commitment: $Comm_e = r * G + v * H$. After that, the user selects a new random number r_s, and uses the private key of the public account to sign the commitment to obtain a digital signature $sig_e = r_s + sk_p * Comm_e$. At this point, this algorithm will create a blockchain Exchange transaction include: $\{Comm_e, sig_e, pk_p, r_s * G, v\}$

After receiving the Exchange transaction, the blockchain node first verifies the validity of the signature, and then checks whether the user's public account balance is greater than or equal to the balance to be converted. If the validation is successful, the balance of the user's public account is deducted and a new balance commitment is added to the private account when the block is confirmed.

Redeem. The algorithm is executed by the user's private account, and the balance in the user's private account can be converted back into the balance in the public account. Assume that the user's private account has a Pederson commitment $Comm_r$ with balance v_r, where $Comm_r = r_r * G + v_r * H$. The Redeem transaction will include the balance v_r contained in the pending redemption commitment and the corresponding blind point factor $r_r * G$. Finally, the user signs the original commitment with the private key sk_s of the private account, and attach it to the transaction: $sig_r = r_s + sk_s * Comm_r$. At this time, the Redeem algorithm will create a blockchain Redeem transaction include: $\{Comm_r, sig_r, v, r_r * G, pk_s\}$

Similar to Exchange algorithm, the blockchain node first verifies the signature after receive the Redeem transaction and uses the blind spend factor points and balances in the transaction to reconstruct the commitment after verification. If the reconstructed promise is the same as the promise saved in the private account, then the Redeem is proved to be legitimate. When the block containing this transaction is confirmed, the commitment of the private account will be deleted, and the balance contained in the commitment will be added to the public account.

Send. The Send algorithm describes the user's micropayment process, in other word, is the process of splitting a commitment into two commitments. The initial commitment can be a commitment in the user's private account, or a commitment in a one-time address owned by the user. The two commitments output will be saved in one-time address. The following will describe the Send algorithm process in detail with micro-payment of a private account as an example.

The algorithm inputs a Pederson commitment with balance v_o from private account, and outputs two Pederson commitments pointing to different one-time addresses $addr_r, addr_s$. These two commitments include: a transfer commitment $Comm_t = r_t * G + v_t * H$ for the receiver and a change commitment $Comm_c = r_c * G + v_c * H$ for the sender. And the input and output balance are equal: $v_o = v_t + v_c$. The algorithm will append a range proof for each commitment to ensure that the balance in the commitment is valid. Only the receiver knows the private key corresponding to the one-time address, and only the sender and receiver know the transaction balance. The sender gets the blinding factor points $r_t * G$ of the receiver's commitment off-chain and uses it to build the transfer commitment to receiver. the change commitment still belongs to the sender but is stored at a new one-time address. The receiver calculates

the commitment hash value by function $H(*)$ and construct the transaction metadata $m=(H(Comm_o)\,|\,addr_s)$, then signs the metadata with private key sk_s^r and sends $sig_r = k_r + m * sk_s^r$ to the sender. The sender also uses its private key sk_s^s to sign the metadata to get a signature $sig_s = k_s + m * sk_s^s$ and uses the aggregated signature $sig = sig_r + sig_s$ as the final signature. At this point, the algorithm will create a blockchain Send transaction include: $\{addr_r, addr_s, Comm_o, Comm_t, Comm_c, sig, k_s * G, k_r * G\}$

The blockchain node first verifies the validity of the signature of the Send transaction, and then verifies the validity of the newly generated commitment by checking the validity of the range proof and the aggregate signature.

Merge. The Merge algorithm describes the process of merging two commitments owned by the user into one commitment. The two commitments input to the algorithm can be the commitments of two one-time addresses, or a commitment of one-time address and a commitment of the user's private account.

Suppose the user intends to merge commitment $Comm_1 = r_1 * G + v_1 * H$ and $Comm_2 = r_2 * G + v_2 * H$, and the output commitment is $Comm_m = r_m * G + v_m * H$, where r_* is the blind factor for commitment and v_* is the balance. The user owns the public and private key pairs of these two commitment addresses. The user who executes the algorithm knows the blind factors of the commitments and the balance contained in the commitments. The user signs the commitments using the private key of the commitment address, and getting the signature sig_1 and sig_2 to prove its ownership of the two commitments. The blockchain node needs to verify the validity of the commitment with range proof, and whether the balance between input and output is equal. The Merge transaction can combine the private balance in the two addresses into one private balance. The merge transaction only operates on two original commitments and a new commitment, and the process will not disclose any critical random numbers and the balance in the commitment, which ensures the privacy of the private balance. This algorithm will create a blockchain Merge transaction include: $\{Comm_m, Comm_1, Comm_2, sig_1, r_{s1} * G, sig_2, r_{s2} * G\}$

The verification of the Merge algorithm also needs to check the validity of the transaction signature, verify the user's ownership of the commitments through commitment signatures, finally check the relationship between the input and output commitments, and verify the validity of the commitment through range proof. The Merge transaction can be marked as a valid transaction after all verifications are successful.

In the BIoT scenario, users can use more efficient public transactions when privacy is not required. In order to meet high privacy requirements, users can transfer the balance of the public account to the private account through the Exchange algorithm, and then generate micro-payment transactions through the Send algorithm. The Merge algorithm can aggregate multiple commitments, and can extract the balance from a one-time address into a private account. If the user intends to withdraw the balance in the private account, redeem transaction can be used to redeem the balance back to the public account. In the BIoT micropayment scenario, the above five algorithms can basically guarantee the user's identity privacy and data privacy.

4.2 Data Interaction

It seems that the amount of data generated in IoT environment is so large that it is impractical to use the blockchain as a storage system for it. Most IoT data should be stored in distributed file systems, such as IPFS while only a small number of short data or data indexes stored in blockchain systems. This part will design a privacy-preserving mechanism for this scenario.

For the data interaction scenario, this mechanism encrypts the short important IoT data or data index and stores the data ciphertext in user's data account $Account_{data}$. The data security and transaction legitimacy are ensured through transaction credential and CA node arbitration mechanism. The schematic illustration of BIoT data interaction is shown in Fig 3. This scheme is composed of three polynomial-time algorithms: $<Upload, Transfer, Arbitration>$.

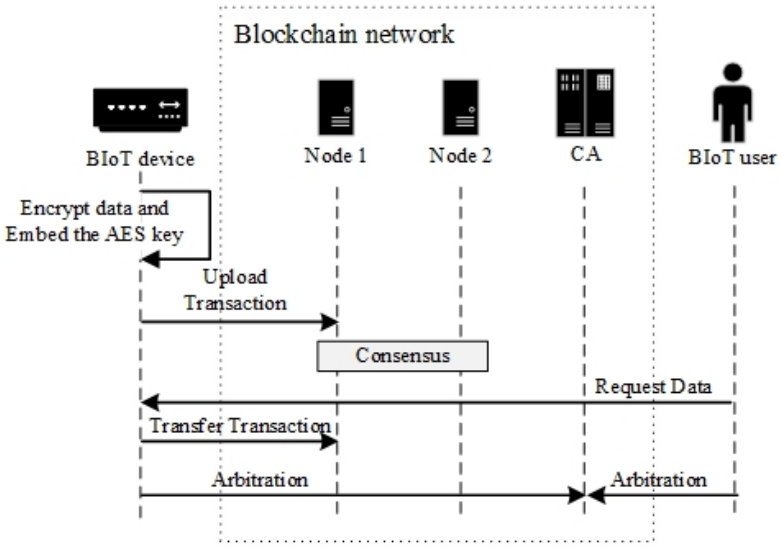

Fig. 3. The schematic illustration of BIoT data interaction

Upload. To ensure the data privacy, the IoT data will be stored as ciphertext. AES is used to encrypt plaintext data. The user keeps the AES key σ, obtains the ciphertext msg_c after encrypting the data, and the plaintext hash e_{msg} is calculated with SHA256. Then the AES key should be encrypted to implement the Transfer and Arbitration algorithm. The AES key is encoded in hexadecimal and then converted to decimal. The obtained decimal number is embedded into the elliptic curve by plaintext embedding algorithm to obtain the corresponding point $Q_\sigma = (x_\sigma, y)$ and abscissa offset θ of the AES key. The BIoT user will use this transaction to store the encrypted information in the user's data account. Finally, this algorithm will create a blockchain Upload transaction include: $\{ msg_c, \theta, e_{msg} \}$.

Transfer. This algorithm generates a transmission transaction. The data owner sends the encryption key to the requester through this transaction. This transaction will be stored in the blockchain and can be used as evidence for future data disputes.

The owner gets the requestor's address and public key off-chain, then selects a random number r_i and generates the encrypted message $(c_1, c_2) = (Q_\sigma + r_i * pk_r, r_i * G)$, which containing the AES key point. This message is encrypted with the requester's public key pk_r. The requester uses his own private key to decrypt the message to obtain the elliptic curve point Q_σ, which is embedded in the AES key. Then the coordinate offset θ corresponding to the message can be obtained by querying the data account of the data owner, and finally recovers the AES key by subtracting the offset from the abscissa of the point.

Finally, the owner will also build a transaction credential in order to guarantee the rights of both parties to the transaction. If the trading party arbitrates the Transfer transaction in the future, the transaction certificate can be used to reconstruct the Transfer transaction to verify whether the trading parties are malicious users. The transaction credential $(\sigma \mid r_i)$ includes the random number r_i that generated the encrypted message for this Transfer transaction and the AES encryption key σ for the IoT data. Assuming that there are n certificate nodes in the consortium blockchain. The owner divides them into m groups, uses the public key of the CA nodes in the group to encrypt the transaction credential sequentially, and obtains m encryption credentials $set_{enc} = \prod_{Enc} \{Enc((\sigma \mid r_i)^{n/m})_j, 0 < j < m\}$. The unique serial number of a set of CA nodes corresponding to each encrypted credential will be written into $set_{Bitest} = \prod \{Bitset_1, ..., Bitset_m\}$. At this time, m encrypted credentials and the corresponding m set of CA nodes will be obtained after calculation. Now, the transfer algorithm will create a blockchain transfer transaction include: $\{ c_1, c_2, set_{enc}, set_{Bitset} \}$

Arbitration. Arbitration will be initiated when the data receiver believes that the key obtained from the Transfer transaction cannot correctly retrieve the IoT data.

The Arbitration mechanism is implemented by the CA nodes. There will be a set of CA nodes decrypting the transaction credential in the transfer transaction to obtain the encryption key and the random number of the transaction. The Transfer transaction includes m encryption credentials and m CA node sets. The first arbitration will be executed by the first CA node set. The $\frac{n}{m}$ nodes contained in the $Bitset_i (0 < i \le m)$ will decrypt the encrypted transaction credential, and obtain the original transaction credentials, which includes the random number and AES key of the Transfer transaction. The CA node then reconstructs the transfer transaction based on the requester's public key and compares it to the on-chain transfer transaction. If the reconstructed transaction is the same as the transfer transaction, the recipient is the fraudster; otherwise, the owner is the fraudster. If the transacting party believes that it has been wrongly convicted as a fraudster, it can use another set of CA node to re-execute the Arbitration mechanism. Eventually, either all CA node sets verify that it is a fraudster, or some CA nodes are found to have security issues that lead to incorrect decisions. The security of CA node is beyond the scope of this paper. According to the result of arbitration, the CA nodes will revoke the fraudster's certificate and construct an Arbitration transaction to transfer the balance of the fraudster's public accounts to the other party.

In the data interaction scenario, IoT data ciphertext is stored in the blockchain through the Upload transaction. Data interaction is mainly realized by Transfer transaction which is stored in the blockchain as a data transfer record to provide proof for future traceability and authentication. If a fraudster constructs a false transaction, which will be verified through the Arbitration mechanism, and the CA node will punish the fraudster and revoke its certificate.

5 Mechanism Analysis

Theorem 1. In the micro-payment scenario, blockchain nodes cannot obtain the information such as private account balances and transaction random numbers based on historical transactions and account storage information.

proof: The blockchain nodes keep all valid historical transactions and current account information. But the node can only get the balance contained in the initial commitment in the Exchange transaction and Redeem transaction, which can be made public. In Send and Merge transactions, blockchain nodes cannot obtain real account information with the commitment based on a one-time address. In terms of data privacy, according to the characteristics of Pederson's commitment, nodes cannot obtain the blinding factor of the commitment through the information contained in the transaction, so they cannot resolve the hidden balance in the commitment. Therefore, in Send and Merge transactions, blockchain nodes cannot obtain the private balance and blinding factors contained in the exchange and cannot infer the identity of the transaction party.

Theorem 2. In the micro-payment scenario, the Send and Merge algorithms cannot generate commitments that exceed the balance contained in the input commitment.

proof: Suppose the malicious user inputs the original commitment $Comm_o = r_o * G + v_o * H$ into the Send algorithm and outputs two new commitments $Comm_1 = r_1 * G + v_1 * H$ and $Comm_2 = r_2 * G + v_2 * H$. If the user outputs more than the original balance, for example $v_2 > v_o$, we subtract the two output commitments from the original commitment first, and get $Diff = Comm_o - Comm_2 - Comm_1 = (r_o - r_2 - r_1) * G + (v_o - v_2 - v_1) * H$. The node will verify that if the $Diff$ is not an elliptic curve point based on G, there will be $v_o \neq v_1 + v_2$, which proves that the transaction is invalid; otherwise, there will be $v_o = v_1 + v_2$ and then $v_1 < 0$. At this point, the node verifying the range proof of the commitment will find that the balance is negative and determine that the transaction is invalid. Merge algorithm is similar to the inverse process of Send algorithm, and the verification method is the same. Therefore, malicious users cannot generate excess balances.

Theorem 3. In the micro payment scenario, if user uses the private account for payment, the flow of the balance cannot be determined by linking the history Send and Merge transactions.

Proof: In a Send transaction, there must be two output commitments. This design can prevent other users from getting the true flow direction of the balance included in the original commitment. Users can generate fake Send transactions to confuse the true balance flow. In this process, the Merge transaction

can cooperate with the Send transaction to generate a commitment containing any balance, but the balance of the commitment will not exceed the sum of the original commitment balance. If users keep using commitments stored in one-time address, then the transaction will never reveal user's identity.

Theorem 4. In the data interaction scenario, blockchain nodes cannot obtain IoT data through historical transaction information.

proof: All BIoT data will be encrypted by AES and stored as ciphertext in blockchain, so malicious users or malicious nodes cannot obtain the plaintext data of IoT devices without the data encryption key. In the Transfer algorithm, the data encryption key will be recorded in the block chain after being encrypted with the public key of the receiver. Malicious users and nodes cannot decrypt the transaction to obtain the data encryption key without the corresponding private key, so the data ciphertext cannot be decrypted. During the Arbitration process, blockchain nodes can only obtain arbitration results from certificate nodes. The arbitration result will only transfer the balance of the public account of the fraudulent party to the other party, and the truly interactive data is not disclosed to the entire blockchain network.

Theorem 5. In the Arbitration algorithm, if the CA nodes in the organization meets high security, the data encryption key will hardly be disclosed.

proof: The data owner will generate m transaction credentials sequentially encrypted by the $\frac{n}{m}$ CA node public keys in the Transfer algorithm. Only when all n / m CA nodes in a group are malicious nodes, the transaction credentials of the Transfer transaction will be disclosed. In the arbitration stage, only one group of m groups executes the arbitration, and only in the condition that all $\frac{n}{m}$ CA nodes in the group participate in the decryption, can the plaintext credentials be finally obtained. In the case where the CA nodes in the organization have high security, the possibility that all randomly selected groups of CA nodes are malicious nodes can be ignored, so transaction credentials are almost impossible to disclose.

6 Experiment and Performance

In this section, we implemented a simple blockchain prototype system to run the privacy-preserving mechanism designed in this paper. The effectiveness of the privacy mechanism was verified by experiments. Then, we analyzed the time overhead and data size of each algorithm in the privacy-preserving mechanism, and compared it with the schemes in other literatures.

6.1 Experiment Setting

We implemented a consortium blockchain prototype system strictly according to the blockchain infrastructure. The system includes a data layer, a network layer, a consensus layer, and an application layer. In practical applications, a smart contract layer should be added below the application layer, and transactions can be written directly as smart contract code to execute automatically. This experiment focuses on the effectiveness and performance of our privacy-preserving mechanism.

To evaluate our prototype system, we set up a test network using four local computers, each of which contained three virtual nodes running in Docker. One of the nodes is a blockchain node that stores the blockchain ledger locally and participates in the consensus process of the blockchain network; the other two nodes continuously send transactions as client nodes. The computer configuration is shown in Table 1.

Table 1. Experiment configuration

No.	Operating system	CPU	Memory	LAN IP
1	Ubuntu 16.04	Intel(R) Core(TM) i7-8750H @ 2.20GHz	16G RAM	192.168.1.236
2	Ubuntu 16.04	Intel(R) Core(TM) i7-8750H @ 2.20GHz	16G RAM	192.168.1.237
3	Deepin 15.10	Intel(R) Core(TM) i5-9300H @ 2.40GHz	8G RAM	192.168.1.238
4	Deepin 15.8	Intel(R) Core(TM) i5-9300H @ 2.40GHz	4G RAM	192.168.1.239

6.2 Experiment Results

Take Send algorithm as an example to demonstrate the effectiveness of the scheme described in this paper. The transactions generated by the Send algorithm are shown in Table 2.

Table 2. The transaction information generated by the send algorithm

Key	Value
TxID	4
TimeStamp	1584274507507535300
InputComm	[“address”: “GaVvF6KYdXVtjRzoqtFia5pUpzELQTRxB”, “commitment”: “113466220299856548520039100827320308151339389671642254574440978714064909558837|507723526372513620061950422477948618173901024891486944359139203552297868447 56”]
OutputComm	[“address”: “Lab7CM4oYv4tweHudT7oSGRk28iDtc9nk”, “commitment”: “674955671579787966843871667451803994496356895194011990289316418268579169182 80|754762439263177026559430402657564706519982797777567306870748099315347233344”], [“address”: “J7mnFdxij8onkCGNsTToyFf6NLu6juv1J”, “commitment”: “125537799355700149526298553382865247611437244063902282758456171106277389052 48|705960325018355910371951605496198021530368161801851566464312272696716011733 48”]
OriginSig	“115034784825872563345151221352903564351894386207802140658352774422545354751588|640399040322479618027725227881214766834275981355116665845554267759157562342 23”
OriginPubKey	“789026590589191895011289170717642775922965307562533177195764452601506205400 6|1103077331146456328640603281033981184604738977950440555483193383273487471689 46”
Signature	“832603398810016324481218906550843404971633895765516841617274880254461583453 0|1329165729978435929521134549712500298905068758645826074769957177689277348017 1”
RandomPoint	“429781827647714563661741677311043261698794614839931686879952543751976345304 36|997599332794898068269896813121177265490084170947660601846998603913715938374 67”
TxHash	“3e8c646834ab9965c083725880948a98ab4777533b208233fe29d4e3324560f1”

We cannot get any information about the transaction balance from the transaction information because the balance is hidden in the commitment of the input InputComm and output OutputComm. Moreover, the one-time address of commitment for the InputComm and OutputComm cannot be found from the account address of the blockchain node user list, so we cannot know which user has this commitment. OriginSig and OriginPubKey can be used to verify the validity of input commitment InputComm.

The Merge algorithm is similar to the Send algorithm. Blockchain nodes and users cannot know the payment relationship and payment balance between users through the historical Send and Merge transactions in the network. Therefore, experiments prove that this scheme can keep the privacy of transactions.

6.3 Experiment Analysis and Comparison

While satisfying the validity of the privacy-preserving mechanism, we should minimize the execution time of the algorithm and reduce the size of the output data. Therefore, we have calculated the implementation efficiency of the scheme in this paper and compared it with existing similar literature schemes.

When calculating the time of execution and the size of the generated data for all the privacy-preserving algorithms in the two scenarios of micropayment and data interaction, we just consider the original output of the algorithm and ignore other information for constructing a complete blockchain transaction, such as timestamp, transaction signatures, etc. The results are shown in Table 3. The data in the table is the average after 100 counts.

Table 3. Performance of our privacy-preserving mechanism

Algorithm	Micro-payment					Data Interaction		
	CreateAccount	Exchange	Redeem	Send	Merge	Upload	Transfer	Arbitration
Time	7ms	273ms	304ms	1274ms	537ms	1.7s	1.2s	6.8s
Size	128B	264B	264B	448B	384B	msg+40B	n/m*(64+4)B+64B	160B

The experimental results show that the algorithm of the micro-payment scenario has a short running time and can almost complete the operation within one second. The Send algorithm requires the sender to communicate with the receiver, inevitably adding gRPC communication time, so the algorithm takes more time to execute. In data interaction scenarios, all algorithm execution time are in seconds. The reasons for time consuming are the off-chain communication task, and lots of computation. The Arbitration algorithm is also time consuming because multiple CA nodes have to decrypt transaction credentials sequentially and multi-party communication is necessary.

The data generated by all the algorithms is very small, because the scheme described in this paper mainly uses Pederson commitment to hide the amount, without generating too much extra proof. The size of the data generated by the Upload algorithm mainly depends on the size of the IoT data(msg). The data generated by Transfer algorithm includes n/m uint32 values used to represent the Bitset of CA node number, and n/m encryption credentials.

Finally, we compare the execution efficiency of the proposed micropayment scenario algorithm with that of the algorithm in literature [16]. The comparison results of the execution time of each algorithm are shown in Fig. 4. The process of CreateAccount generating account local information is similar between the two schemes, but other algorithms of the scheme described in this paper are much less time consuming than similar algorithms in literature [16]. In this paper, Pederson commitment and range proof based on ring signature are used instead of zero-knowledge proof technology which requires a lot of time to generate proofs, thus reducing a lot of computing time. However, the verification time of the scheme in this paper is longer than that in the literature [16], because in the zero-knowledge proof, the proofs generated by other algorithms is just verified directly, while in this scheme, the results of other algorithms need to be verified step by step.

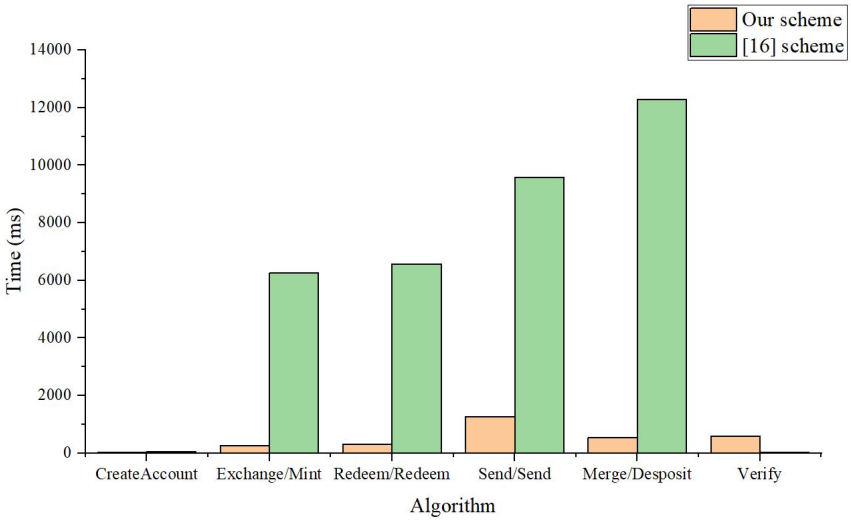

Fig. 4. The time overhead comparison of our scheme and literature [16] in micro-payment scenario

The comparison results of the data generated by each algorithm are shown in Fig. 5. Literature [16] needs to generate additional data such as transaction serial number and constructed zero-knowledge proof, while the scheme in this paper does not have other additional data, so the algorithms in this scheme have less data costs.

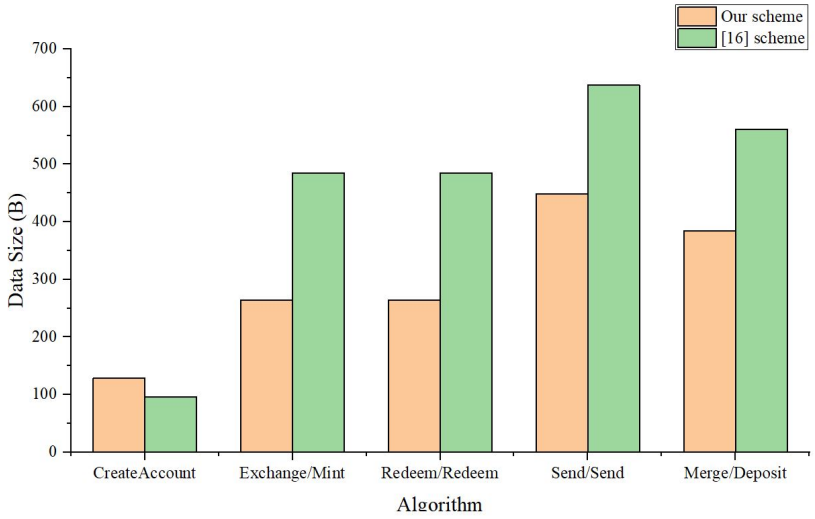

Fig. 5. The amount of generated data comparison of our scheme and literature [16] in micro-payment scenario

7 Conclusion

Firstly, we summarize the current application of BIoT into two categories: micro-payment and data interaction, and discusses privacy issues in these two scenarios. Secondly, this paper designs a universal model of BIoT based on the consortium blockchain based on these two scenarios, and designs a dual-account model to store the user's data and balance. Then, based on the model, the corresponding privacy-preserving mechanisms are proposed for these two application scenarios. According to the micro-payment scenario in the IoT environment, this paper proposes a privacy-preserving mechanism for the consortium blockchain. This mechanism uses Pederson commitment and range proof, etc. to hide the transaction balance in the commitment, and uses the one-time address and commitment merger to confuse the trader's identity, so as to protect the identity privacy and data privacy of the BIoT. After that, a privacy-preserving mechanism combining encryption, transaction credentials and arbitration is proposed to store small IoT data or its index in ciphertext, and keep the privacy of BIoT in data interaction scenarios through transaction credentials and certificate node arbitration. Mechanism analysis proves the privacy and security of the mechanism. Finally, experiments on algorithms show the effectiveness of the proposed mechanisms with low overhead in time consuming and data processing.
In the future, we will try to reduce or even remove the participation of certificate nodes in the process of data interaction under the conditions of keeping security and privacy, and test the efficiency of these mechanism in the actual BIoT application scenario.

Acknowledgements

This work is supported by National Key R&D Program of China (No. 2018YFC0832300; No. 2018YFC0832303).

References

[1] O Said, M. Masud, Towards internet of things: Survey and future vision, International Journal of Computer Networks 5(1)(2013) 1-17.

[2] Ericsson, IoT security-protecting the networked society. <https://www.ericsson.com/en/reports-and-papers/white-papers/iot-security-protecting-the-networked-society>, 2019 (accessed 29.03.20).

[3] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>, 2008 (accessed 29.03.20).

[4] M.S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the Internet of things: a comprehensive survey, IEEE Communications Surveys & Tutorials 21(2)(2018) 1676-1717.

[5] P. Alfonso, T. Nachiket, M. Giovanni M, L Francesco, P Antonio, Blockchain and IoT integration: a systematic survey, Sensors 18(8)(2018) 2575-2611.

[6] N. Kshetri, 1 Blockchain's roles in meeting key supply chain management objectives, International Journal of Information Management 39(2018) 80-89.

[7] N.Z. Aitzhan, D Svetinovic. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams, IEEE Transactions on Dependable and Secure Computing 15(5)(2016) 840-852.

[8] R. Chaabouni, H. Lipmaa, B Zhang, A non-interactive range proof with constant communication, in: Proc. 2012 Financial Cryptography and Data Security, 2012

[9] Y. Yu, Y.N. Li, J.F. Tian, J.W Liu, Blockchain-based solutions to security and privacy issues in the Internet of Things, IEEE Wireless Communications 25(6)(2018) 12-18.

[10] G. Maxwell, CoinJoin: Bitcoin privacy for the real world <https://bitcointalk.org/index.php?topic=279249>, 2013 (accessed 29.3.20).

[11] F.K. Maurer, T. Neudecker, M. Florian, Anonymous CoinJoin transactions with arbitrary values, in: Proc. 2017 IEEE Trustcom/ BigDataSE/ICESS, 2017.

[12] G. Malavolta, D. Schröder, Efficient ring signatures in the standard model, in: Proc. 2017 International Conference on the Theory & Application of Cryptology & Information Security, 2017.

[13] S.L. Ma, Y. Deng, D.B. He, J Zhang, X Xie, An efficient NIZK scheme for privacy-preserving transactions over account-model blockchain, Journal of Latex Class Files 13(9)(2014) 1-12.

[14] I. Miers, C. Garman, M. Green, A.D. Rubin, Zerocoin: Anonymous distributed e-cash from bitcoin, in: Proc. 2012 IEEE Symposium on Security and Privacy, 2013.

[15] E.B. Sasson, A. Chiesa, C. Garman, M. Green, Zerocash: Decentralized anonymous payments from bitcoin, in: Proc. 2014 IEEE Symposium on Security and Privacy, 2014.

[16] Z.S. Guan, Z.G. Wan, Y. Yang, Y. Zhou, B.T. Huang, BlockMaze: an efficient privacy-preserving account-model blockchain based on zk-SNARKs, Cryptology ePrint Archive 2019(2019) 1354.

[17] M. Zhang, S. Wang, P Zhang, L. He, X. Li, S.S. Zhou, Protecting data privacy for permissioned blockchains using identity-based encryption, in: Proc. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, 2019.

[18] J.T. Hao, Y. Sun, H. Luo, A safe and efficient storage scheme based on blockchain and IPFS for agricultural products tracing, Journal of Computer 29(6)(2019) 158-167

[19] S.M. Ali, K. Dolui, F. Antonelli, IoT data privacy via blockchains and IPFS, in: Proc. the 7th International Conference on the Internet of Things, 2017.

[20] G. Zyskind, O. Nathan, A. Pentland, Enigma: Decentralized computation platform with guaranteed privacy. <https://arxiv. org/abs/1506.03471>, 2015.

[21] W.L. Chen, M.J. Ma, Y.J. Ye, Z.B. Zheng, Y.R. Zhou, IoT service based on JointCloud blockchain: the case study of smart traveling, in: Proc. 2018 IEEE Symposium on Service-oriented System Engineering (SOSE), 2018.

[22] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: The case study of a smart home, in: Proc. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, 2017.