applied sciences

MDPI

Article

Cloud-Assisted Privacy Protection Energy Trading Based on IBS and Homomorphic Encryption in IIoT

Huajie Wang, Yao Xiao, Yong Feng *, Qian Qian, Yingna Li and Xiaodong Fu

Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming 650500, China

* Correspondence: fybraver@kust.edu.cn

Abstract: The decentralized and tamper-proof features of blockchain technology can solve the problems of low compatibility, poor flexibility, and single point of failure in the traditional Industrial Internet of Things (IIoT). However, the transparency of the blockchain ledger makes the privacy disclosure of users a huge security risk. Given the privacy leakage problem exposed in the existing energy trading scheme based on the blockchain, this paper creatively proposes a privacy protection scheme for IIoT energy trading based on an identity-based signature (IBS) and homomorphic encryption. On the premise of satisfying the transaction traceability and verifiability, this scheme uses IBS technology to provide an anonymous mechanism for energy trading nodes and utilizes Paillier homomorphic encryption to prevent the disclosure of transaction amounts. To meet the high-concurrency and high-throughput energy trading requirements in IIoT, moreover, the proposed scheme combines the off-chain storage with cloud assistance and the off-chain transaction based on PCN to reduce redundant data written into the blockchain and to improve the concurrent trading efficiency, respectively. The security analysis and performance evaluation results show that the proposed scheme can realize the dual privacy protection of identities and transaction amounts in the trading process at the cost of reasonable calculation.

Keywords: cloud computing; blockchain; Industrial Internet of things; payment channel network; identity-based cryptography; homomorphic encryption; privacy protection energy trading

Citation: Wang, H.; Xiao, Y.; Feng, Y.; Qian, Q.; Li, Y.; Fu, X. Cloud-Assisted Privacy Protection Energy Trading Based on IBS and Homomorphic Encryption in IIoT. *Appl. Sci.* **2022**, *12*, 9509. https://doi.org/10.3390/app12199509

Academic Editor: David Megías

Received: 22 August 2022
Accepted: 19 September 2022
Published: 22 September 2022

1. Introduction

The IIoT, which has ubiquitous interconnection, integrated perception, intelligent optimization, and security protection, is regarded as a key technology in the era of Industry 4.0 [1]. However, it is a significant issue for industrial systems to meet the rising energy demand of IIoT applications due to the expanding number of IIoT nodes and performance requirements [2]. A vast amount of energy trading data are exposed to security threats, which could result in significant economic issues, due to the dynamic and expansive character of the IIoT system.

In the traditional energy Internet, transactions are often verified, stored, and managed by a trusted center, which also records all transaction data. According to [3], this centralized operation mode has some drawbacks and can easily become the target of network attackers. In the context of the IIoT, energy trading is currently moving from a centralized to a distributed model [4]. Neither the producer nor the consumer of energy can be regarded as a trustworthy entity in the dispersed P2P trade environment [5]. Building confidence in the trading platform is therefore essential for its acceptance in the consumer sector. Decentralization, distrust, distributed sharing, and tamper-proof transaction records are all benefits of the P2P system powered by blockchain, which has been investigated and used in energy trading recently [1,4–10].

The IIoT energy trading combined with blockchain technology mainly faces the following problems: first, some existing research schemes send all private transactions to the

blockchain for execution, which greatly limits the scalability of energy trading. Second, the block will grow bigger and take longer to construct if the transaction data is entirely recorded on the blockchain. It is recommended to write the least amount of information feasible when taking into account the throughput and transaction delay of the blockchain [11]. Finally, the existing transaction models generally ignore some important practical constraints, such as privacy disclosure caused by frequent transactions [12]. The analysis of this data over time may reveal the user's trading habits and other private data. In the energy trading process, it is best to conceal or limit the transaction information.

Given the above problems, this paper proposes cloud-assisted privacy protection energy trading based on IBS and homomorphic encryption in IIoT. The proposed scheme uses PCN [13] to realize off-chain transactions and cloud-assisted off-chain storage—that is, not all data are saved on the blockchain, only indexes or abstracts, and the actual data are saved by the cloud service provider (CSP). In addition, the proposed scheme combines an identity-based signature with Paillier homomorphic encryption. On the one hand, it provides pseudonyms for nodes participating in blockchain network transactions. It creates a pseudonym for each node and uses that pseudonym for transactions to maintain anonymity. On the other hand, the confidentiality of the transaction amount is realized by Paillier [14] homomorphic encryption. The contributions of this paper are as follows:

- We propose a uniform energy trading framework built on the blockchain in light of the IIoT energy trade scenarios now in use. This design reduces duplicate data transmitted to the blockchain and increases transaction efficiency by combining PCN-based off-chain transactions and cloud-assisted off-chain storage.
- Using IBS and Paillier homomorphic encryption, we propose a privacy protection mechanism and apply it to off-chain transactions on the blockchain, allowing transactions using pseudonyms to protect node identity privacy and transaction data information security.
- We have carried out simulation experiments, and the experimental results prove the effectiveness and feasibility of the scheme.

The rest of this paper is organized as follows. Section 2 reviews the relevant work; Section 3 introduces the relevant background knowledge required by the scheme; Section 4 describes the proposed scheme in detail; in Section 5, performance evaluation and experimental analysis will be conducted; Section 6 contains some discussion; Section 7 concludes the paper.

2. Related Work

In this section, the current developments in energy trading and blockchain privacy protection are outlined.

2.1. Blockchain-Based Energy Trading Systems

According to Abdella et al. [15], the deployment of distributed energy trading systems has security issues such as information confidentiality, message integrity, and availability attacks. Most existing energy trading schemes based on blockchain, however, lack a privacy protection mechanism, in which the transparent records published on the blockchain can easily lead to the violation of users' privacy.

Li et al. [1] proposed the Fenechain scheme. Yahaya et al. [10] proposed a P2P energy trading scheme using the alliance blockchain, and an electric vehicle energy trading scheme based on privacy protection blockchain was proposed by Baza et al. [8] in 2021. The aforementioned plan [1,8,10] stores a large amount of data on the blockchain, making it impossible to scale and maintain user privacy. A blockchain-based energy trading system for the V2V environment was proposed by Kim et al. [6]. This scheme reduces the data stored on the blockchain, so it has good scalability, but there are hidden dangers in data security. Chen et al. [16] put forth a reliable framework for selling energy that integrates distributed optimization and blockchain technology. However, their algorithms rely too much on consensus mechanisms.

Potential pertinent research has provided some remedies in light of the aforementioned issues. Gai et al. [9] conducted research on privacy protection in blockchain-based energy trading. The proposed method can successfully hide the account characteristics of active and inactive users, and use account mapping technology to prevent attackers from illegally obtaining private data. The suggested technique can simultaneously protect the seller's energy sales distribution and prevent traders' privacy information from being breached by mining various energy trading volumes, but it still lacks scalability. Lu et al. [17] presented encryption based on ciphertext policy attributes (CP-ABE) as the primary technique for reconstructing the transaction model in order to address the privacy protection issues that the majority of blockchain-based transaction models experience. The suggested technique creates the PP-BCTS universal distributed transaction model (Privacy Protection Blockchain Transaction Scheme). It can accomplish ciphertext-based transaction arbitration for fine-grained access control. With this layout, the security and dependability of the transaction model may be considerably enhanced while also maximizing the protection of sensitive data. Pop et al. [18] implemented demand response plans on the open blockchain in a decentralized manner, and combined zero knowledge proof and smart contracts to protect users' privacy data. However, these two schemes [17,18] do not consider the storage limit on the blockchain. An inventive multi-blockchain energy trading architecture was put forth by Huang et al. [7], which realized efficient energy trading by using the side chain mechanism but stored the transaction data on the main chain, which had the risk of privacy disclosure.

We originally investigated the problem of IIoT energy trading based on PCN and put out a secure energy trading scheme [19]. This research suggests an effective and secure IIoT energy trading platform as an enhancement and extension of earlier work, in which we have rewritten the whole article to enhance each part and provided more details of the proposed trading framework. To thoroughly assess the performance of the scheme, we conducted experiments.

2.2. Privacy Protection of Blockchain Transaction Data

The blockchain ledger is ready to provide transaction traceability and verifiability. Although the ledger is transparent, this also makes the identities of blockchain users and the confidentiality of transaction amounts visible.

The cross-domain IIoT device authentication method based on blockchain that Shen et al. [20] developed makes use of IBS technology to provide entities with anonymous identities. This scheme provides a revocable anonymous identity and cross-management domain authentication mechanism for entities and realizes the protection of entity identity privacy information. However, the possibility of privacy leaking during entity contact is not taken into account.

A private, decentralized, token-based energy trading system was suggested by Aitzhan et al. [21]. Using blockchain and cryptography technologies to safeguard privacy and ensure secure transactions, this method enables peers to negotiate energy pricing in an anonymous manner, but there is a problem with repeated communication messages.

An enhanced group signature mechanism is employed by Yang et al. [22] to safeguard nodes' identities during blockchain transactions. This system ensures node anonymity and forward security, has an excellent signature and verification efficiency, and has non-forgeable signature features. This approach, however, solely makes use of the signature algorithm to achieve anonymity for both users' identities and is unable to achieve privacy protection for transaction amounts, failing to satisfy users' expectations for both identity and transaction amount privacy.

Wang et al. [23] propose a framework to hide the amount of money by encrypting and decrypting using a homomorphic encryption cryptosystem. The scheme sets up a dummy account, which can receive the same amount of bitcoin in each transaction but cannot consume it. This ensures that the hidden amount is always positive However, since the scheme sets

up an additional account, and the encryption requires I cycles, the efficiency of transaction encryption and verification is low, and it cannot meet the user's needs for identity privacy.

In order to improve the security of the transmission data, Subramaniyaswamy et al. [24] proposed a new method to securely encrypt the sensor signal value of the IoT. This method uses an improved lightweight algorithm based on homomorphic encryption ring learning with errors to encrypt the transmission data. For big data analysis in IoT applications, Sasikumar et al. [25] developed a joint consensus algorithm based on blockchain and artificial intelligence, and claimed that the algorithm reduced energy consumption and solved current security problems.

3. Preliminary Knowledge

3.1. Payment Channel Network

To overcome this scalability issue, the PCN proposed by Poon J et al. [13] can take most of the transactions off-chain. Users only need to enter the initial and final balances for each channel during the transaction. The typical PCN transaction is shown in Figure 1. In PCN, a set quantity of electronic money is deposited into a shared account by both parties to the transaction. Through several hops, the sender can transfer money to the recipient. The intermediary node in the payment channel path will assess the transfer fee when the payment travels through its payment channel. The blockchain is updated with the details of both parties' balances when the transaction is completed [19].

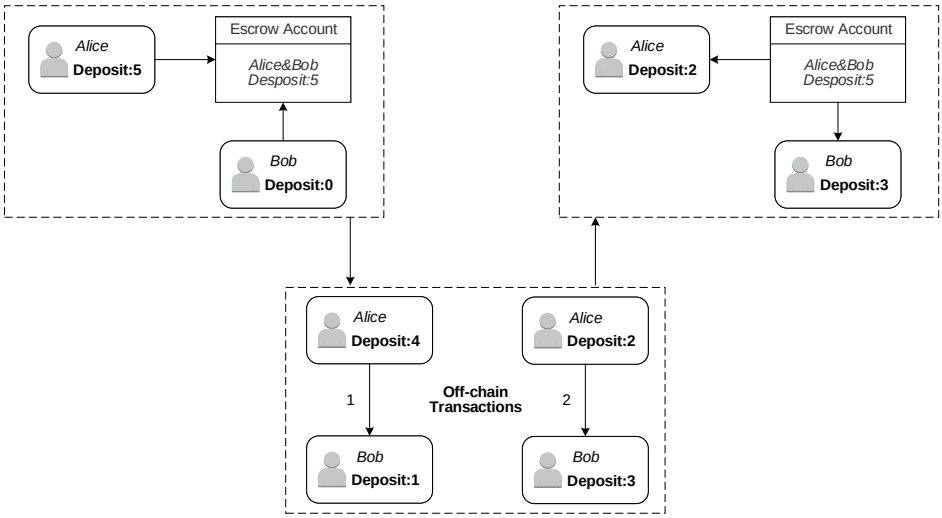

Figure 1. An example of off-chain payment channel transaction.

Figure 2 depicts a particular payment procedure. In the hash time lock contract (HTLC), the maximum level of user tolerance for transaction time is specified. Hash lock and time lock make up HTLC. The sender creates a random number R and receives its hash value H through the hash lock from the receiver. The hash value H in the transaction contract is stored in the sender or each intermediate node. The transfer fund can only be withdrawn after the forwarder has given the secret number R, which guarantees that the payment receipt cannot be disputed. Each transaction under time locking has a deadline for completion. The sent monies will be returned to the sender if they do not obtain the random number R within the allotted time.

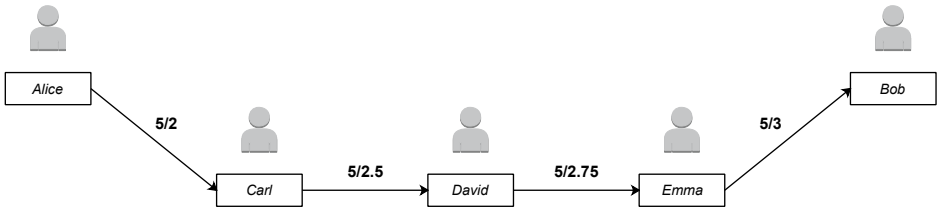

Figure 2. A demonstration of payment channel capacity. Alice sends 2 bitcoin to Bob through 3 nodes.

3.2. Identity-Based Signature

To solve the defects of power concentration and vulnerability caused by the certificate mechanism, Shamir [26] put forth an identity-based cryptosystem (IBC). In this system, the user's public key can be a string containing information about the user's identity, whereas the user's private key is produced by the key generation center (KGC). Boneh and Franklin [27], in 2001, proposed the first identity-based encryption (IBE) method. The identity-based signature (IBS) derived from IBE is used for device authentication. In this paper, the IBS technology specified in the Chinese national standard SM9 [28] issued by the State Password Administration of China is used to ensure the identity anonymity of the node [29]. The algorithm is described as follows:

- System master key establishment: KGC selects a random number $s \in [1, N-1]$ as the main private key and calculates the element $P_{Pub-s} = [s]P_2$ in G_2 as the main public key and KGC keeps s and discloses P_{Pub-s};
- User key generation: assuming the identity identifier of the user is ID_A, KGC calculates t_1 on the finite field F.

$$t_1 = H_1(ID_A||hid, N) + s, \tag{1}$$

If t_1 is 0, it is necessary to recalculate the master private key and update all users' private keys. If t_1 is not equal to 0, calculate t_2:

$$t_2 = [s]{t_1}^{-1}, \tag{2}$$

The user's private key is $d_A = [t_2]P_1$. The user's public key pk_A is

$$pk_A = H_1(ID_A||hid) + P_{Pub-s}, \tag{3}$$

So far, the key pair of the user is (d_A, pk_A);

- Signature generation: assuming there is a message M, the signer calculates $g = e(P_1, P_{Pub-s})$, selects the random number $r \in [1, N-1]$, calculates $w = g^r$, then calculates $h = H_2(m||w)$, $l = (r-h) mod N$, and finally calculates $S = [l]d_x$.
  The signer's signature value (h, S) for the message can be obtained.
- Signature verification: assuming the message received by the verifier and its signature are $m^{'}$ and $(h^{'}, S^{'})$, respectively, the verifier first verifies whether both $h^{'} \in [1, N-1]$ and $S^{'} \in G_1$ are correct. If one of them is not correct, the verification fails. Otherwise, P, u, and $w^{'}$ are calculated.

$$P = [H_1(ID_A||hid, N)]P_2 + P_{Pub-s}, \tag{4}$$

$$u = e(S^{'}, P), \tag{5}$$

$$w^{'} = g^{h^{'}}, \tag{6}$$

Finally, $h_2 = H_2(m^{'}||w^{'}, N)$ is calculated and compared with $h^{'}$. If it is consistent, the verification passes [30].
During the above calculation, N is the order of elliptic curve used by the SM9 digital signature algorithm. P_1, P_2 are the generators of the N-order cyclic subgroup G_1 and G_2. hid is the signature private key generation function identifier. H_1 and H_2 are hash functions.

3.3. Paillier Homomorphic Encryption

Rivest et al. [31] made the initial proposal for homomorphic encryption in 1978. Semi-homomorphic encryption and fully homomorphic encryption are the two categories into which it can be separated. Complete homomorphic encryption is extremely expensive and unable to handle high transaction throughput demands. The classical probabilistic homomorphic encryption technique employed in this study, the Paillier algorithm, is more

sophisticated than other methods and has been extensively used in the Internet of Things, cloud computing, and other applications. The following is a description of the Paillier homomorphic encryption algorithm:

- Key generation: arbitrarily select large prime numbers p, q, and calculate $n = p \times q$ and $\lambda = lcm(p-1, q-1)$; If an integer is arbitrarily selected and $gcd(L(g^{\lambda} mod n^2), n) = 1$ is satisfied, $pk_P = (n, g)$ and $sk_P = (p, q)$ are the public key and private key, respectively.
- Encryption: arbitrarily select $r < n$, and for plaintext $m < n$, the encrypted ciphertext $c = g^m r^n mod n^2$.
- Decrypt: assuming the ciphertext is c, the decrypted ciphertext

$$m = \frac{L(c^{\lambda} mod n^2)}{L(g^{\lambda} mod n^2)} mod n, \tag{7}$$

- Additive homomorphism: for two plaintext Msg_1 and Msg_2, Paillier homomorphic encryption operation is expressed as: $Enc(\cdot)$. Following Paillier homomorphic encryption, the ciphertext is, accordingly, $Enc(Msg_1)$ and $Enc(Msg_2)$, then $Enc(Msg_1)$ and $Enc(Msg_1)$ satisfy the following requirements:

$$Enc(Msg_1) \times Enc(Msg_2) = Enc(Msg_1 + Msg_2), \tag{8}$$

  During the above calculation, $lcm(\cdot, \cdot)$ is defined as the least common multiple of 2 parameters $L(u) = \frac{(u-1)}{n}$.

4. The Proposed Scheme

The processes of the energy trading scheme based on IBS and Paillier homomorphic encryption are described in detail in this section.

4.1. System Structure

For the energy trading scenario in the IIoT, this research proposes a Paillier homomorphic encryption and IBS-based energy trading scheme. The following role information is included in the scheme:

- Energy nodes: the users can become energy nodes after registering, and the KGC then distributes the key pair. Energy nodes can engage in energy transactions to complete the blockchain's point-to-point transaction information transfer procedure.
- Transaction broadcast nodes (TBN): as full nodes in the blockchain, they do not directly participate in transactions and are only responsible for broadcasting transaction information to other energy nodes.
- Energy station (ES): the energy station is accountable for powering energy nodes and has the authority to choose qualified participants in the energy trading system.
- Cloud service provider (CSP): stores complete transaction information to reduce data writing on the blockchain.
- Key generation center (KGC): for system nodes, the KGC serves as the authority for key generation and distribution. It also offers services for pseudonym generation, identity authentication, and transaction correctness checking.

The privacy protection of transactions in this scheme is reflected in two aspects: user anonymity and encryption of transaction amounts. The system architecture is shown in Figure 3.

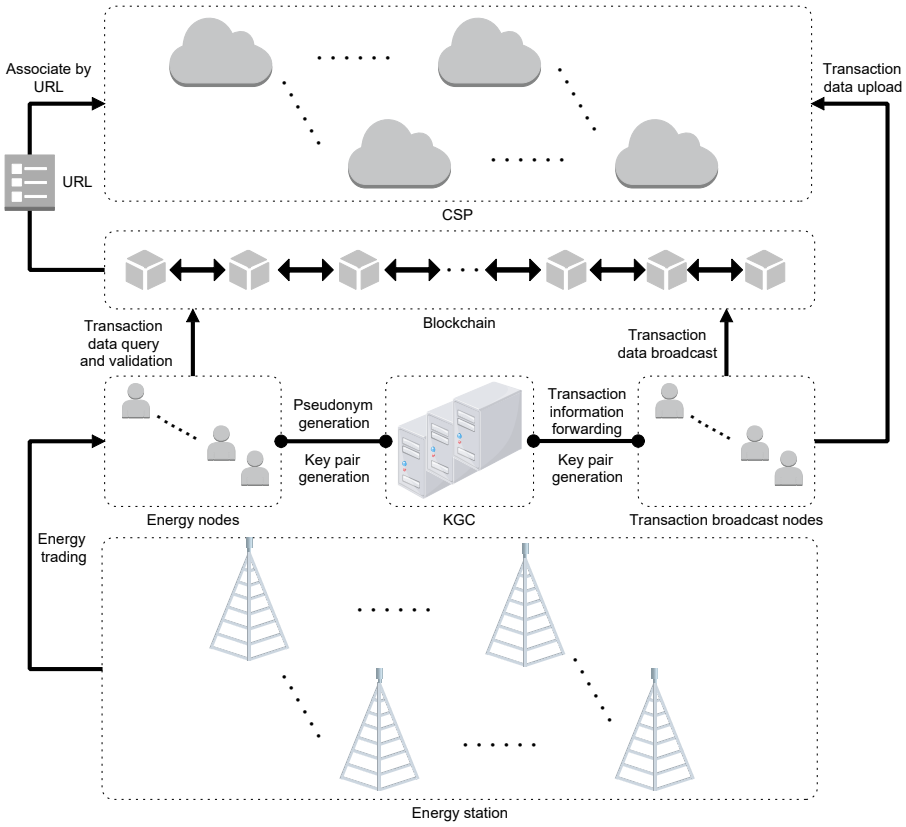

Figure 3. System architecture.

4.2. Privacy-Preserved Security Energy Trading

4.2.1. Initialization

PCN is represented as the digraph $G = (V, E)$, where vertex V corresponds to the set of accounts and vertex E corresponds to the set of currently available installment channels [19]. The amount of additional bitcoin the sender can pay the receiver is determined by the load on each coordinated side. The energy node $U \in V$, where U_0 and U_s address the payer and the collector separately. On either side, there is an HTLC tolerance—that is to say, the maximum hanging time for accommodating the irregular number R_{num}. In order for payment to be successful, the following limitations on viability must be satisfied:

- If, for any link in the path $U_0 \to U_1 \to U_2 \to \ldots \to U_t \to U_s$, the initial balance of u0 is the sum of the payment amount and the accumulation of fees charged by all intermediate nodes,

$$\beta_{(0,1)} = \alpha + \phi_{(1,t)}, \tag{9}$$

  where U_i and U_j's transfer amount is shown by $\beta_{(i,j)}$. α denotes the amount of bitcoin that U_s will eventually obtain. $\phi_{(i,j)}$ denotes the total transaction fee charged by intermediate nodes from U_i to U_j.
- The balance of any channel on the path should be no less than the total amount of the transfer fee and the fees charged for all subsequent links, which are expressed as a backward balance constraint:

$$\beta_{(i,i+1)} \geq \alpha + \phi_{(i+1,t)}. \tag{10}$$

- At the same time, the current channel balance should also be more than the balance after the total expenditure charged by all the intermediate nodes on the previous link, which is expressed as a forward balance constraint:

$$\beta_{(i,i+1)} \geq \beta_{(0,1)} - \phi_{(1,i-1)}. \tag{11}$$

- The transfer fee on any path must be less than the available capacity λ of the current channel—that is, it needs to satisfy the following condition:

$$\lambda_{(i,i+1)} \geq \beta_{(i,i+1)}. \tag{12}$$

- The transfer needs to be completed within the HTLC tolerance ξ on each channel in the path. ξ_i is the HTLC tolerance, which means the maximum tolerance time of using the current channel.

$$\xi_{i-1} = \xi_i + \Delta, \tag{13}$$

KGC generates the signature master key pair (s, P_{Pub-S}), and the Paillier encryption key (s_P, P_{Pub-P}). Alice and Bob join the blockchain with the real identities ID_{Alice} and ID_{Bob}, and KGC generates public–private key pairs for them based on their real identities.

4.2.2. Pseudonym Generation

Alice sends a pseudonym application PA_{Alice} to KGC and attaches the signature of the application with Alice's private key:

$$Msg_1 = PA_{Alice}||Sig_{Alice}(PA_{Alice}), \tag{14}$$

The SM9 digital signature algorithm is shown in Algorithm 1. After receiving Msg_1, KGC first verifies it. If the verification passes, KGC connects Alice's real identity with the current timestamp and takes a hash value, which will be used as Alice's pseudonym:

$$Pid_{Alice} = H(ID_{Alice}||Timestamp). \tag{15}$$

The SM9 signature algorithm is shown in Algorithm 2. Similarly, Bob sends a pseudonym application message Msg_2 to KGC and obtains a pseudonym Pid_{Bob}:

$$Msg_2 = PA_{Bob}||Sig_{Bob}(PA_{Bob}), \tag{16}$$

$$Pid_{Bob} = H(ID_{Bob}||Timestamp). \tag{17}$$

Algorithm 1 Digital signature.

Input: P_{Pub-S}, System Paramater, Msg
Output: (h, S)
$g = e(P_1, P_{Pub-s}) \in G_T$;
$l = 0$;
while $l == 0$ **do**
    select $r \in [1, N-1]$;
    $w = g^r \in G_T$;
    Transform w's data type into a bit string;
    $h = H_2(m||w)$;
    $l = (r - h) mod N$;
end while
return (h, S)

Algorithm 2 Signature verification.

Input: P_{Pub-S}, System Parameter, $Msg^{'}$, $(h^{'}, S^{'})$
Output: *True* or *False*
Transform $h^{'}$ to integer type;
Transform $S^{'}$'s data type to an elliptic curve point;
if $h^{'} \in [1, N-1]$ **then**
    if $S^{'} \in G_1$ **then**
        $g = e(P_1, P_{Pub-s}) \in G_T$;
        $t = g^{h^{'}} \in G_T$;
        $h_1 = H_1(ID_x||hid, N)$;
        $P = [h_1]P_2 + P_{Pub-s} \in G_2$;
        $w^{'} = ut \in G_T$;
        Transform $w^{'}$'s data type into a bit string;
        $h_2 = H_2(m^{'}||w^{'}, N)$;
        if $h_2 == h^{'}$ **then return** true;
        end if
    end if
end if

4.2.3. Open Channel

In this phase, Alice will choose t strings χ_i at random, after which she will lock the x bitcoin on HTLC. Algorithm 3 allows for the calculation of the (χ_i, ω_i) forwarded to other nodes in a subsequent transaction procedure.

Before Alice makes payment to Bob, both parties need to create a third-party common account. Alice and Bob use the Paillier homomorphic encryption public key P_{Pub-P} to encrypt the amount deposited in the common account to obtain the ciphertext $M_1 = Enc(\beta_0^{Alice})$ and $M_2 = Enc(\beta_0^{Bob})$, where $Enc(m)$ means that m is encrypted with the Paillier encryption public key P_{Pub-P}.

Alice and Bob send M_1 and M_2 to KGC, respectively, and then KGC obtains the public account balance through calculation:

$$\beta = M_1 \times M_2 = Enc(\beta_0^{Alice}) \times Enc(\beta_0^{Bob}) = Enc(\beta_0^{Alice} + \beta_0^{Bob}). \tag{18}$$

Common account ACC is expressed as:

$$ACC = (Address_{Alice}, Address_{Bob}, Pid_{Alice}, Pid_{Bob}, \beta, \xi, \lambda). \tag{19}$$

β represents the common account balance and λ represents the capacity available on the path. $Address_{Alice}$ and $Address_{Bob}$ represent the account addresses of Alice and Bob, respectively. KGC calculates the balance of the common account β according to the Paillier homomorphic property.

Algorithm 3 HTLC.

Input: χ, ξ_i
Output: (χ_i, ω_i)
for $i = 1 \rightarrow t$ **do**
    if $\forall i \in [t]$ **then**
        $\chi_i \in 0, 1^*$
        $\omega_i = H(\oplus_{j=1}^{t} \chi_j), j \geq i$
    end if
return $((\chi_1, \omega_1) \ldots (\chi_t, \omega_t))$
end for

Before the first off-chain transaction, Alice encrypts the transfer amount α and the account balance β_A after the transfer to obtain the ciphertext $M_3 = Enc(\alpha)$, $M_4 = Enc(\beta_A)$. Alice sends M_3 and M_4 to KGC to verify the validity of its initial transfer—that is, to verify the formula

$$M_1 = Enc(\alpha) \times Enc(\beta_A) = Enc(\alpha + \beta_A). \tag{20}$$

If the equation holds, it means that Alice meets the balance requirements for the initial account. KGC outputs a channel identifier $\sigma_{(U_0,U_t)}$ after passing the verification.

4.2.4. Transaction

It is assumed that a channel exists between Alice and Bob that satisfies the criteria. Alice initially determines the entire cost ζ in the payment path before sending the payment:

$$\zeta = \sum_{i=0,j=1}^{i=t-1,j=t} \zeta_{(i,j)}. \tag{21}$$

If Alice does not have enough bitcoin, she will give up the payment, or send the contract to each transferor in $\sigma_{(U_0,U_t)}$. According to Algorithm 4, each node in the channel $\sigma_{(U_0,U_t)}$ must confirm the ensuing channel capacity. If the verification is unsuccessful, the transaction is put on hold; if it is successful, contracts will be issued for more nodes.

Algorithm 4 Transactions for intermediate nodes.

Input: $\sigma_{(U_i,U_{i+1})}$
Output: Decision
if Decision = Forward **then**
  if $\beta_{(i,i+1)} \leq \lambda_{(i,i+1)}$ **then**
    $\xi_{i+1} = \xi_i - \Delta$
    $\lambda_{(i,i+1)} = \lambda_{(i,i+1)} - \beta_{(i,i+1)}$
    HTLC $(U_i, U_{i+1}, \omega_{i+1}, \xi_{i+1}, \beta_{(i,i+1)})$
  else
    Send$(U_{i-1}, (\omega_i, \sigma_{(U_{i-1},U_i)}, \beta_{(i-1,i)}))$
  end if
else if Decision = Abort **then**
  $\lambda_{(i,i+1)} = \lambda_{(i,i+1)} + \beta_{(i,i+1)}$
  Send$(U_{i-1}, (\omega_i, \sigma_{(U_{i-1},U_i)}, \beta_{(i-1,i)}), Abort)$
else if Decision = Accept **then**
  Send$(U_{i-1}, (\chi_{i+1} \oplus \chi_i, \omega_i, \sigma_{(U_{i-1},U_i)}, \beta_{(i-1,i)}), Accept)$
end if

If every node in the channel $\sigma_{(U_0,U_t)}$ has timely completed the contract, Bob can release x_t to collect the locked α bitcoins from the contract. Following Bob's publication of x_t, other nodes in the channel $\sigma_{(U_0,U_t)}$ can calculate the corresponding random number to determine their channel transmission fees. The specific procedure is shown in Algorithm 5. The core process of the PCN-based trading scheme is shown in Figure 4.

Algorithm 5 Process of contract completion.

Input: $\sigma_{(U_{i-1},U_i)}$
Output: Decision
if $H(\chi_i) = \omega_i and \xi_i = \xi_{now} + \Delta$ **then**
  Send$(U_{i-1}(\chi_i, \omega_i, \sigma_{(U_{i-1},U_i)}, \beta_{(i-1,i)}), Accept)$
else
  Send$(U_{i-1}(\omega_i, \sigma_{(U_{i-1},U_i)}, \beta_{(i-1,i)}), Abort)$
end if

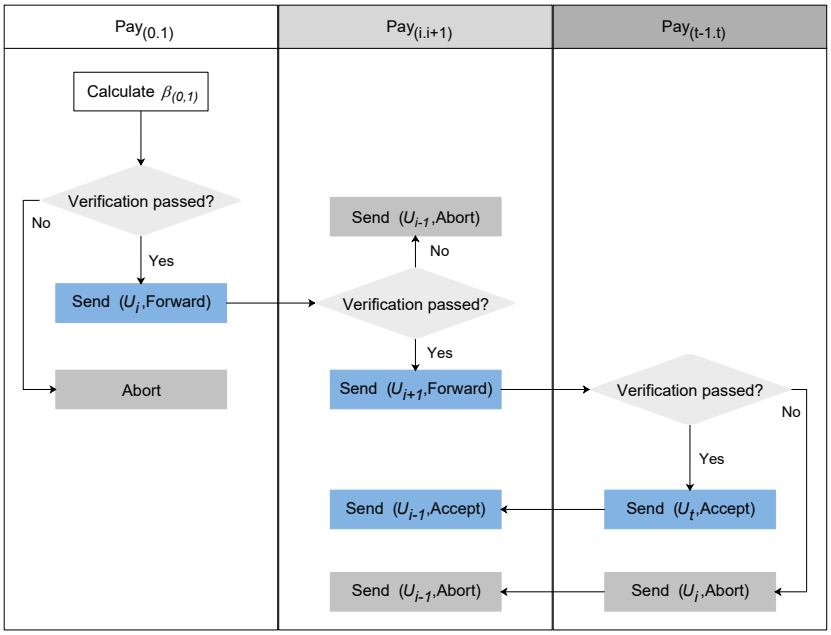

Figure 4. Core process of PCN-based trading scheme.

4.2.5. Close Channel

At the completion stage, the current account balances of Alice and Bob are β_A and β_B, respectively. Alice encrypts the value of $\beta_A + \zeta$ with homomorphic encryption public key P_{Pub-P} to obtain M_5 and sends it to KGC. Similarly, Bob encrypts β_B to obtain M_6 and sends it to KGC. KGC verifies the formula:

$$M_5 \times M_6 = Enc(\beta_A + \zeta) \times Enc(\beta_B) = \beta. \tag{22}$$

If the equation is true, KGC returns the identification of the successful transaction to Alice and Bob, and then forwards the encrypted transaction data to the TNB for broadcasting in the blockchain.

The encrypted transaction data are then uploaded to the CSP with a signature from TBN. The data format stored in the CSP is:

$$Data_{CSP} = Enc(Data_{TX})||Pid_{Alice}||Pid_{Bob}||Sig_{TBN}(Enc(Data_{TX})||Pid_{Alice}||Pid_{Bob}). \tag{23}$$

After being signed by TBN, the transaction data summary is published to the blockchain. The summary information contains a uniform resource locator (URL), and the corresponding file hosted on the cloud service can be obtained through the URL. Figure 5 depicts the connection between the encrypted data kept in CSP and the data digest on the blockchain. By contrasting the summary data with the cloud data's hash value, the integrity and validity of the transaction information may be confirmed. The overall flow of transactions is shown in Figure 6. The format of the transaction information finally written into the blockchain is:

$$Data_{block} = H(Data_{CSP})||URL||Sig_{TBN}(H(Data_{CSP})||URL). \tag{24}$$

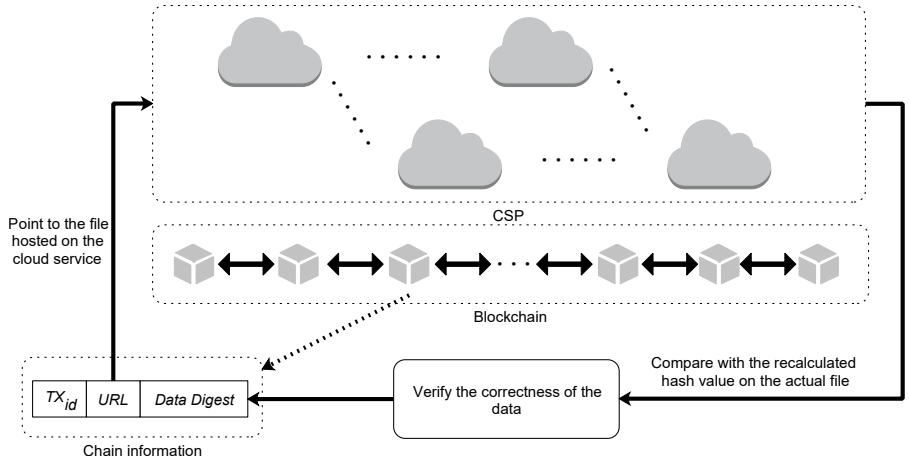

Figure 5. Relationship between chain data and cloud data.

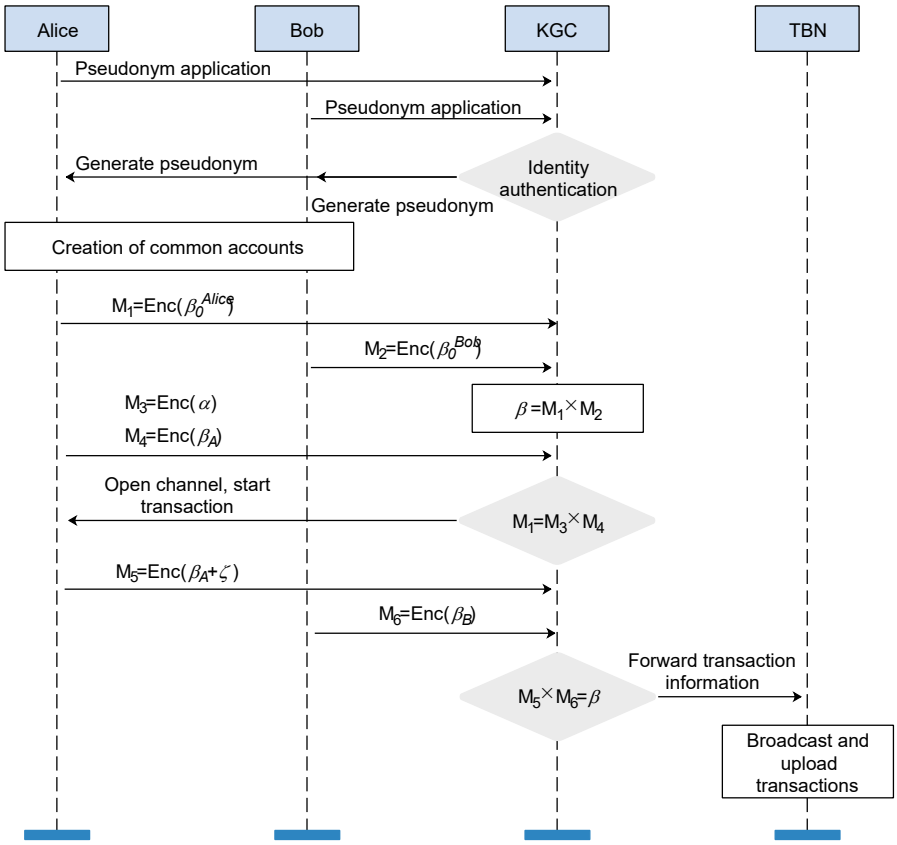

Figure 6. Overall flow of transactions.

5. Performance Analysis

5.1. Safety Analysis

The scheme put forward in this study is designed based on IBS and Paillier homomorphic encryption. Among them, Paillier homomorphic encryption is based on the complex residual problem. During the whole transaction process, both parties first encrypt the amount deposited in the common account and send it to KGC. KGC calculates the balance $\beta = M_1 \times M_2$ of the common account. Next, Alice encrypts the current balance and the current transfer amount and sends them to KGC. Similarly, KGC verifies the feasibility of the initial transaction by calculating whether $M_1 = M_3 \times M_4$ is established. Finally, both parties encrypt the account balance and send it to KGC. After passing the verification $\beta = M_5 \times M_6$, KGC forwards the encrypted transaction data to the transaction broadcast-

ing node, and the transaction is completed. The above process does not require the private keys of both parties to decrypt any encrypted data. When the homomorphic decryption key is unknown, the transaction amount and account balance cannot be decrypted

5.2. Anonymity Analysis

After registering with the energy station, all users can become legitimate energy nodes, and all energy nodes can request a pseudonym from KGC before engaging in trading. The node utilizes the pseudonym that KGC creates for it to carry out energy transactions. The method put forward in this study makes use of IBS technology specified in the Chinese national standard SM9 [28] issued by the China National Cryptology Administration to generate pseudonyms for nodes. The identity password security of bilinear pairs serves as the foundation for the algorithm's security, including the problem of solving discrete logarithms on elliptic curves. The calculation process of anonymous identities includes a one-way algorithm of hash functions. Therefore, it is not feasible to crack the password. The anonymity of the proposed scheme is guaranteed.

5.3. Traceability of Pseudonyms

In the process of pseudonym application, the KGC stores the agreement between the node's true identity and its pseudonym in the local list. If any dispute occurs in the transaction process, the disputing party can send an arbitration application to the KGC. The KGC can query the local list to resolve the real identity of the node and gradually trace the transaction process.

5.4. Scheme Comparison

The comparison between the plan put forward in this study and previous works is shown in Table 1. Li et al. [1] and Yahaya et al. [10] have built their own transaction frameworks through the alliance blockchain. These two systems make the guarantee that the data are safe, but saving all the trading data on the chain will lead to large storage overhead. The confidentiality and transparency of energy trading are guaranteed by the blockchain-based energy trading privacy protection method developed by Baza et al. [8], but the system's overall computation and storage overhead are high. In the scheme of [20], the blockchain-based cross-domain Industrial Internet of Things device authentication scheme proposed, they use IBS technology to provide anonymous identitied for entities, but it does not take into account the privacy leakage problem that may be caused in the process of entity interaction. In the scheme of [21], through the system's integration of blockchain and multi-signature technologies, peers can securely conduct transactions while negotiating energy rates in an anonymous manner. However, due to the redundancy of communication messages and insufficient routing, the system has scalability problems. In the scheme of [22], the nodes in the blockchain transaction have their identities protected using an upgraded identity-based group signature technique; however, the privacy disclosure of transaction data is also not taken into account. Ref. [23] proposes a framework to hide the amount by encrypting and decrypting using a homomorphic encryption cryptosystem, which can realize the encryption of the transaction amount, but cannot meet the needs of users for identity privacy. In this study, users' identity privacy and transaction information privacy are protected using a transaction mechanism based on IBS and Paillier homomorphic encryption. Both the node identity and transaction amount privacy are safeguarded during the transaction procedure. Additionally, with the help of the cloud, the amount of redundant data written to the blockchain is decreased.

Table 1. Scheme comparison.

Scheme	Confidentiality of Transaction Amount	PCN	Cloud Assistance	Anonymity of Nodes
[1]	yes	no	no	yes
[8]	yes	no	no	yes
[10]	no	no	no	yes
[20]	no	no	yes	yes
[21]	no	no	no	yes
[22]	no	no	no	yes
[23]	yes	no	no	no
Proposed	yes	yes	yes	yes

5.5. Computation Overhead

The scheme suggested in this study makes use of the SM9 digital signature technique to create an anonymous identity for nodes and an anonymous identity signature private key for nodes, guaranteeing the confidentiality of the identities of both parties to the transaction. Based on the node's genuine identification, the KGC creates a special ID for it, and it stores the link between the pseudonym and the real identity of the node in the local list. The proposed scheme also leverages the Paillier additive homomorphic encryption algorithm to safeguard the nodes taking part in the transaction from unauthorized users reading sensitive data by encrypting the transaction amount. Additionally, the blockchain's distributed storage system and consensus algorithm guarantee the data's integrity, stop unauthorized users from secretly altering the data on the chain, and successfully fend off DDoS attacks.

The SM9 digital signature algorithm and the Paillier encryption scheme's encryption and decryption of plaintext and ciphertext processes account for the majority of the computation overhead in the approach suggested in this work. Therefore, the simulation in this paper is mainly aimed at the running time of the SM9 digital signature algorithm and the Paillier encryption algorithm involved in the trading process. We use Java language and jPBC library to complete the simulation. The simulation environment is a Windows 10 system with AMD Ryzen 7 4800H CPU and 8.00 GB memory. In order to be closer to the IIoT environment, we use VMware Workstation 15 Pro to deploy virtual machines. The virtual machine environment is Ubuntu 20.04 with 2 GB of memory. Table 2 gives a description of the symbols.

Table 2. Symbol description.

Entity	Operation
$Sig()$	represents signature operation
$Ver()$	represents signature verification operation
$Enc()$	represents encryption operation
$Dec()$	represents decryption operation
$Mul()$	represents ciphertext multiplication

Table 3 displays the operating hours of each party in the suggested scheme. It is worth noting that the generation process of the system master key pair and the real identity key pair of each node is regarded as pre-calculation and is not discussed in the performance analysis.

Table 3. Main operations of the proposed scheme.

Entity	Operation
Alice	$Sig() + 4Enc()$
Bob	$Sig() + 2Enc()$
KGC	$2Ver() + 3Mul()$
TBN	$2Sig()$
CSP	$Ver()$

The computation overhead of the pseudonym mechanism based on IBS is shown in Table 4.

Table 4. Computation overhead of pseudonym mechanism based on IBS.

Entity	Computation Overhead
Alice	644.9301 ms
Bob	659.9820 ms
KGC	1501.5812 ms
TBN	1475.2068 ms
CSP	801.2629 ms

The key size is connected to the relevant procedures in the Paillier homomorphic encryption technique. Figure 7 and Table 5 shows the relationship between relevant operations and key size in the Paillier homomorphic encryption algorithm.

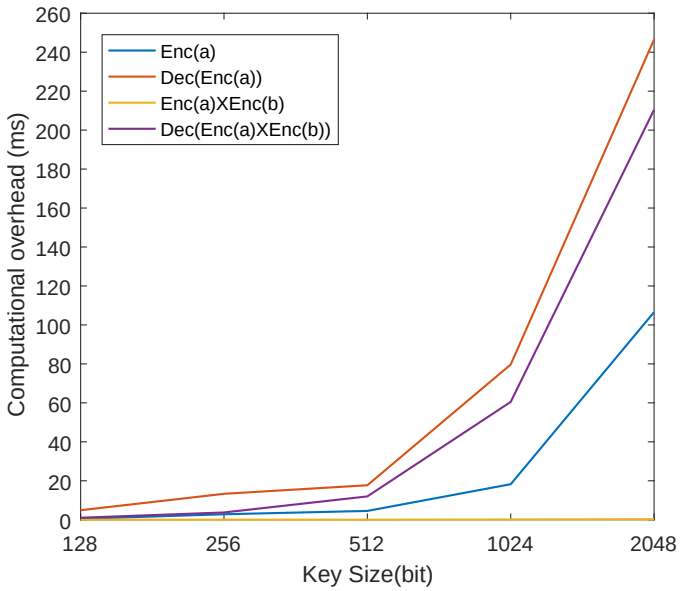

Figure 7. The effect of key size on the computation overhead of Paillier homomorphic encryption algorithm.

Table 5. The effect of key size on the computation overhead of Paillier homomorphic encryption algorithm.

Operation	128 bits	256 bits	512 bits	1024 bits	2048 bis
$Enc(a)$	0.4764 ms	2.8156 ms	4.5277 ms	18.2366 ms	106.5709 ms
$Dec(Enc(a))$	4.8908 ms	13.3131ms	17.7056 ms	79.6857 ms	246.5112 ms
$Enc(a) \times Enc(b)$	0.0319 ms	0.0409 ms	0.0412 ms	0.1439 ms	0.2872 ms
$Dec(Enc(a) \times Enc(b))$	1.0243 ms	3.7358 ms	11.9766 ms	60.4985 ms	210.4611 ms

As can be seen from Figure 7 and Table 5, when the key size is greater than 1024 bits, the computation overhead of $Enc(a)$, $Dec(Enc(a))$, and $Dec(Enc(a) \times Enc(b))$ rises sharply.

$Enc(a) \times Enc(b)$ is hardly affected by the key size. In the proposed scheme, the *keysize* = 512 bit is selected. The total computation overhead of each part is shown in Figure 8. Figure 9 displays the comparison between the overall computation overhead of the approach suggested in this research and that of [8]. Figure 9 illustrates the advantages of the scheme described in this paper in terms of computing overhead.

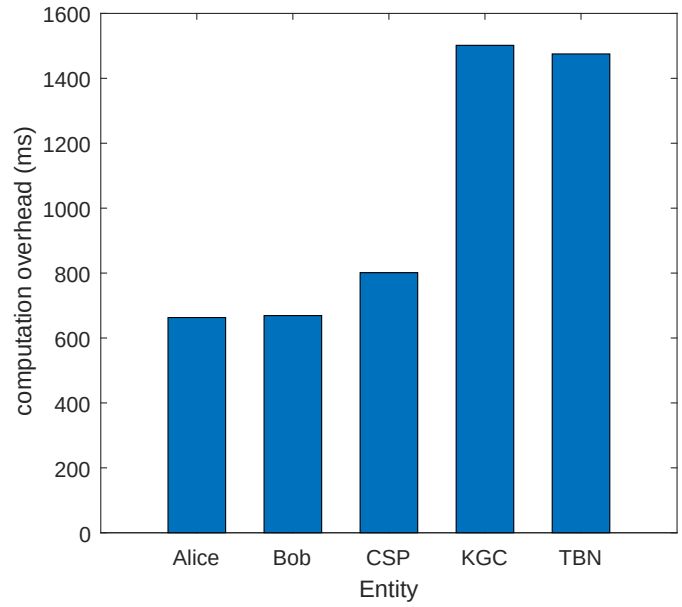

Figure 8. Computation overhead of each entity.

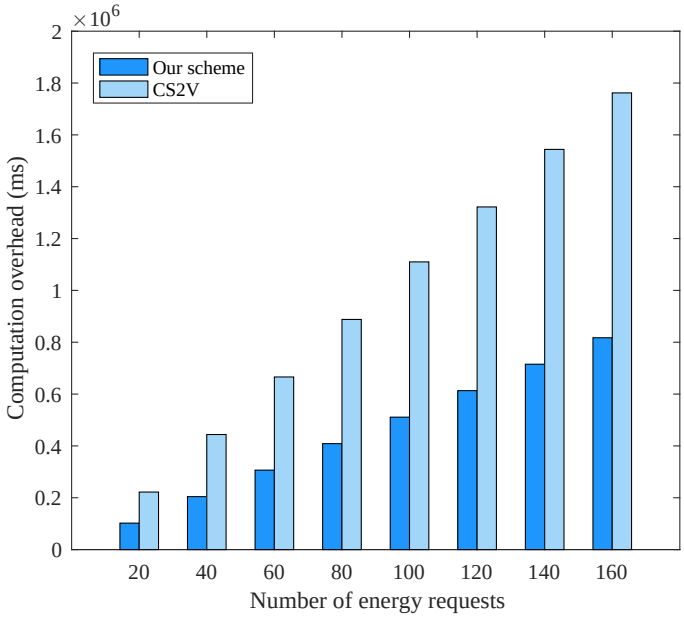

Figure 9. Computation overhead comparison with CS2V [8].

5.6. Communication Overhead and Storage Overhead

At the stage of applying for a pseudonym, Alice and Bob send the pseudonym request information and signature to KGC: $|PA_{Alice}| + |Sig_{Alice}(PA_{Alice})| = 0.195$ KB, $|PA_{Bob}| + |Sig_{Bob}(PA_{Bob})| = 0.195$ KB. KGC returns the pseudonym to Alice and Bob: $|Pid_{Alice}| + |Pid_{Bob}| = 0.125$ KB. In the transaction stage, Alice conducted four homomorphic encrypted information transmissions: $|M_1| + |M_3| + |M_4| + |M_5| = 1.199$ KB. Meanwhile, Bob conducted two homomorphic encrypted transmissions: $|M_2| + |M_6| = 0.599$ KB. Then, KGC calculated the ciphertext product three times: $|Enc(\beta_0^{Alice})x \times Enc(\beta_0^{Bob})| +$

$|Enc(\alpha) \times Enc(\beta_A)| + |Enc(\beta_A + \zeta) \times Enc(\beta_B)| = 0.902$ KB. At the transaction completion stage, TBN uploads and broadcasts $Data_{CSP}$ and $Data_{block}$: $|Data_{CSP}| + |Data_{block}| = 1.037$ KB. It should be noted that the number of times the HTLC contract is sent depends on the number of nodes in the payment channel, so it is not discussed here. The communication overhead of each entity is shown in Table 6.

Table 6. Communication overhead of each entity.

Entity	Communication Overhead
Alice	1.394 KB
Bob	0.794 KB
KGC	1.027 KB
TBN	1.037 KB

We evaluated the storage overhead of the on-chain storage and the off-chain storage to assess the impact of the off-chain storage technique suggested in this paper. Figure 10 depicts the outcomes of our comparison between the storage overhead in [8] and that in this system. Figure 10 shows that the strategy suggested in this research has benefits in terms of storage overhead.

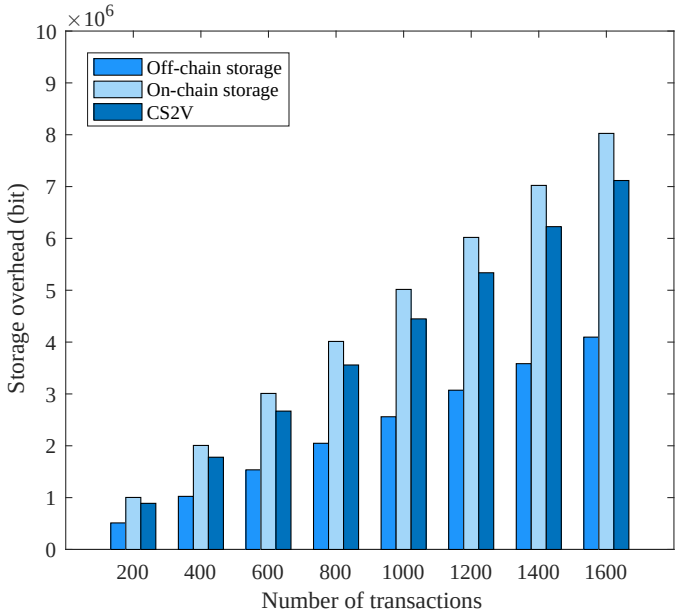

Figure 10. Storage overhead comparison with CS2V [8].

6. Discussion

The scheme presented in this study introduces the SM9 digital signature algorithm and Paillier homomorphic encryption algorithm in order to effectively protect the nodes' privacy and information security in energy trading. Additionally, cloud-assisted PCN-based off-chain transactions offer a more effective and practical method for transactions. The scheme is feasible, according to the simulation findings.

The method outlined in this paper has some limitations. The computation overhead will rise as a result of the frequent data interchange between KGC and nodes, as explained in Section 3. However, the transaction cost is lowered as a result of the off-chain transaction method, which is acceptable. Future studies must find ways to lessen KGC's computational requirements as much as possible. In addition, the security and performance of the SM9 algorithm have been proven in [28], so this article does not repeat it. Finally, the scheme outlined in this study has the potential to theoretically address the security issue associated with blockchain-based energy trading. Blockchain technology is now made up of a significant number of heterogeneous blockchains due to its ongoing development. In the future,

our scheme needs to be tested on a variety of blockchains, and its latency and throughput should be improved according to specific energy trading scenarios.

7. Conclusions

In this study, we present a privacy protection solution for IIoT energy trading that is based on IBS and Paillier homomorphic encryption. In order to protect the confidentiality and security of the nodes' identification information during the trading process, IBS is especially utilized to offer anonymous IDs for energy trading nodes. The transaction value is encrypted using the Paillier homomorphic encryption technique, ensuring the privacy of both parties' transaction data. Additionally, PCN's off-chain transaction mechanism and cloud-assisted off-chain storage mechanism minimize the resource restrictions caused by protracted confirmation periods and high transaction costs on the blockchain network.

The security analysis and simulation results demonstrate that the proposed method in this study offers benefits over [8] in terms of computation overhead and storage overhead and that the proposed scheme's communication cost is tolerable. In addition, this scheme realizes the dual privacy protection of user identity and transaction information that is not realized in most current energy trading schemes. In future research work, the scheme needs to be further tested on a variety of blockchains, and the computation overhead of each entity needs to be reduced.

Author Contributions: Conceptualization, Y.F., Y.L. and X.F.; methodology, Y.F., X.F., H.W. and Y.X.; software, Y.X. and H.W.; validation, H.W. and Q.Q.; formal analysis, H.W. and Y.X.; investigation, Y.X. and Q.Q.; writing—original draft preparation, H.W. and Y.X.; writing—review and editing, Y.F., Y.L., Q.Q. and H.W.; visualization, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Yunnan Key Laboratory of Blockchain Application Technology under Grant 202105AG070005 (YNB202111) and by the National Natural Science Foundation of China under Grant 62062047 and Grant 61662042.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Li, M.; Hu, D.; Lal, C.; Conti, M.; Zhang, Z. Blockchain-Enabled Secure Energy Trading with Verifiable Fairness in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6564–6574. [CrossRef]
2. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3690–3700. [CrossRef]
3. Dong, Z.; Luo, F.; Liang, G. Blockchain: A secure, decentralized, trusted cyber infrastructure solution for future energy systems. *J. Mod. Power Syst. Clean Energy* **2018**, *6*, 958–967. [CrossRef]
4. Guan, Z.; Lu, X.; Wang, N.; Wu, J.; Du, X.; Guizani, M. Towards secure and efficient energy trading in IIoT-enabled energy internet: A blockchain approach. *Future Gener. Comput. Syst.* **2020**, *110*, 686–695. [CrossRef]
5. Chowdhury, M.J.M.; Usman, M.; Ferdous, M.S.; Chowdhury, N.; Harun, A.I.; Jannat, U.S.; Biswas, K. A cross-layer trust-based consensus protocol for peer-to-peer energy trading using fuzzy logic. *IEEE Internet Things J.* **2021**, *9*, 14779–14789. [CrossRef]
6. Kim, M.; Lee, J.; Oh, J.; Park, K.; Park, Y. Blockchain based energy trading scheme for vehicle-to-vehicle using decentralized identifiers. *Appl. Energy* **2022**, *9*, 119445. [CrossRef]
7. Huang, X.; Zhang, Y.; Li, D.; Han, L. A Solution for Bilayer Energy-Trading Management in Microgrids Using Multiblockchain. *IEEE Internet Things J.* **2022**, *9*, 13886–13900. [CrossRef]
8. Baza, M.; Sherif, A.; Mahmoud, M.M.E.A.; Bakiras, S.; Alasmary, W.; Abdallah, M.; Lin, X. Privacy-Preserving Blockchain-Based Energy Trading Schemes for Electric Vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9369–9384. [CrossRef]
9. Gai, K.;Wu, Y.; Zhu, L.;Qiu, M.; Shen, M. Privacy-Preserving Energy Trading Using Consortium Blockchain in Smart Grid. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3548–3558. [CrossRef]
10. Yahaya, A.S.; Javaid, N.; Almogren, A.; Ahmed, A.; Gulfam, S.M.; Radwan, A. A Two-Stage Privacy Preservation and Secure Peer-to-Peer Energy Trading Model Using Blockchain and Cloud-Based Aggregator. *IEEE Access* **2021**, *9*, 143121–143137. [CrossRef]

11. Zhang, Y.; Yang, D.; Xue, G. CheaPay: An Optimal Algorithm for Fee Minimization in Blockchain-Based Payment Channel Networks. In Proceedings of the 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
12. Cavoukian, A.; Polonetsky, J.; Wolf, C. Smartprivacy for the smart grid: Embedding privacy into the design of electricity conservation. *Identity Inf. Soc.* **2010**, *3*, 275–294.
13. Poon, J.; Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016. Available online: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf (accessed on 10 August 2022).
14. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the 1999 International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; pp. 223–238.
15. Abdella, J.; Shuaib, K. Peer to peer distributed energy trading in smart grids: A survey. *Energies* **2018**, *11*, 1560. [CrossRef]
16. Chen, S.; Zhang, L.; Yan, Z.; Shen, Z. A Distributed and Robust Security-Constrained Economic Dispatch Algorithm Based on Blockchain. *IEEE Trans. Power Syst.* **2021**, *37*, 691–700.
17. Lu, X.; Guan, Z.; Zhou, X.; Wu, L.; Du, X.; Guizani, M. An Efficient and Privacy-Preserving Energy Trading Scheme Based on Blockchain. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6
18. Pop, C.D.; Antal, M.;Cioara, T.; Anghel, I.; Salomie, I. Blockchain and Demand Response: Zero-Knowledge Proofs for Energy Transactions Privacy. *Sensors* **2020**, *20*, 5678. [CrossRef] [PubMed]
19. Feng, Y.; Xiao, Y.; Li, D.; Fu, X. PCN-based Secure Energy Trading in Industrial Internet of Things. In Proceedings of the 2020 International Conference on Blockchain and Trustworthy Systems, Dali, China, 6–7 August 2020; pp. 305–318.
20. Shen, M.; Liu, H.; Zhu, L.; Xu, K.; Yu, H.; Du, X.; Mohsen, G. Blockchain-assisted secure device authentication for cross-domain industrial IoT. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 942–954.
21. Aitzhan, N.Z.; Svetinovic, D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 840–852.
22. Yang, Y.; Cai, J.; Zhang, X.; Yuan, Z. Privacy preserving scheme in block chain with provably secure based on SM9 algorithm. *J. Softw.* **2019**, *30*, 1692–1704.
23. Wang, Q.; Qin, B.; Hu, J.; Xiao, F. Preserving transaction privacy in bitcoin. *Future Gener. Comput. Syst.* **2020**, *107*, 793–804.
24. Subramaniyaswamy, V.; Jagadeeswari, J.; Indragandhi, V.; Jhaveri, R.H.; Vijayakumar, V.; Kotecha, K.; Ravi, L. Learning with Error Algorithm for Faster Encryption of IoT Sensor Signal-Based Edge Devices. *Secur. Commun. Netw.* **2020**, *107*, 793–804.
25. Sasikumar, A.; Ravi, L.; Kotecha, K.; Saini, R.J.; Varadarajan, V.; Subramaniyaswamy, V. Sustainable Smart Industry: A Secure and Energy Efficient Consensus Mechanism for Artificial Intelligence Enabled Industrial Internet of Things. *Comput. Intell. Neurosci.* **2022**, *2022*, 1419360.
26. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Paris, France, 9–11 April 1984; pp. 47–53.
27. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2001; pp. 213–229.
28. Cheng, Z. The sm9 Cryptographic Schemes. Cryptology ePrint Archive. 2017. Available online: https://eprint.iacr.org/2017/117 (accessed on 10 August 2022).
29. Liu, S.G.; Liu, R.; Rao, S.Y. Secure and efficient two-party collaborative SM9 signature scheme suitable for smart home. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 4022–4030. [CrossRef]
30. Yang, X.; Yuan, S.; Zhou, H.; Ding, B. A proxy-protected proxy signature based on SM9. In Proceedings of the 2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC), Shenzhen, China, 9–11 October 2021; pp. 166–170.
31. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On data banks and privacy homomorphisms. *Found. Secur. Comput.* **1978**, *4*, 169–180.