A Privacy-Preserving Lightweight Energy Data Sharing Scheme Based on Blockchain for Smart Grid

Xinyang Li[1], Yujue Wang[1], Yong Ding[1,2(✉)], Shiye Ma[3], Bei Xiao[3], Zhihong Guo[3], Xiaorui Kang[3], Xiaohui Ma[3], and Jia Mai[3]

[1] Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China
[2] Cyberspace Security Research Center, Pengcheng Laboratory, Shenzhen, China
stone_dingy@126.com
[3] Shanghai Energy Technology Development Co., Ltd, Shanghai, China

Abstract. As many users join the smart grid system, energy companies need user energy data to manage and improve energy delivery. However, while enjoying the services provided by energy companies, the energy data submitted by users may lead to the risk of privacy leakage. To solve this problem, this paper proposes a privacy-preserving scheme for smart grid based on blockchain and multi-receiver encryption (PPBME). The scheme utilizes lightweight multi-receiver encryption, blockchain, and smart contract technologies to solve the problem of privacy leakage when user data are shared. Also, our PPBME construction employs off-chain storage technology to improve the scalability of the blockchain, so that reducing the storage pressure of the blockchain. This paper also proposes a compensation mechanism for the loss of user privacy. According to the security analysis and computational cost analysis, our PPBME scheme is efficient and secure in supporting smart grid applications.

Keywords: Smart grid · Blockchain · Lightweight · Multi-receiver encryption · Off-chain storage

1 Introduction

Nowadays, as people pay attention to renewable energy power generation, there is an urgent need for an intelligent system that evaluates and prices electricity in real-time, which cannot be offered by the classic power distribution system. The smart grid was introduced in the early 2000s s to integrate the two-way communication infrastructure into the traditional power grid [12], which provides functions such as digital communication between users and suppliers, and monitoring, updating, and reliable distribution of electrical energy for smart meters. It is necessary to provide security mechanism on user data [3].

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 461, pp. 91–110, 2022.
https://doi.org/10.1007/978-3-031-24386-8_6

With the popularization of smart grids and smart home appliances, the data generated and transmitted in smart grids has grown tremendously. Regarding companies responsible for supplying and transmitting electricity, the data collected by smart meters can help them improve system performance and enhance the user experience. Nevertheless, users want a great service experience without their privacy being gained by any entity other than the company they trust [18]. Also, the information transmission between smart meters and service providers faces security and privacy challenges [24]. In the process of data sharing, how to protect the privacy, security and verifiability of user energy data is a problem that smart grids need to solve. Tampering with energy data may mislead service providers and lead to financial losses. In addition, the adversary's long-term analysis of energy data may reveal the user's daily behavior, leading to the leakage of user privacy.

1.1 Our Contributions

To address the privacy and security problems of energy data sharing in smart grid, this paper proposes a privacy-preserving scheme based on blockchain and multi-receiver encryption (PPBME). User data includes the usage data of various types of energy for a period of time, which may contain the user's private information. These data are required for the service providers to adjust resource allocation and regulate prices. Since in the process of data sharing over the public network, user energy data may subject to common known attacks such as eavesdropping and tampering, our PPBME construction is designed to ensure the confidentiality and integrity of the data by utilizing blockchain and data encryption technologies. PPBME also employs off-chain storage technology to store the original data in the cloud. The blockchain stores the relevant credentials for data usage, thereby improving the scalability of the blockchain. Due to the limit computing power of smart meters, the encryption scheme of PPBME in data sharing uses lightweight multi-receiver encryption, which can better achieve fine-grained access control. In order to incentivize users to share their private energy data, the smart contracts are used to generate contract accounts, and service providers will compensate users for privacy leakage by paying to the contract accounts.

The security analysis demonstrates that the proposed PPBME construction can protect the privacy and security of user energy data, as well as resist traditional attacks. Theoretical and experimental analysis show that the computing cost of the proposed PPBME construction is significantly reduced compared with related solutions.

1.2 Related Works

To address the privacy protection issues of communications between suppliers and users in smart grids, many cryptographic schemes have been proposed. Ding et al. [4] presented an efficient metering data aggregation scheme, which supports batch verification of collectors and power service providers, and ensures the privacy and integrity of metering data. Chen et al. [2] proposed a privacy

protection scheme based on the certificateless aggregate signcryption technology, where masking random numbers are used to hide the consumption data of users. Wei et al. [14] presented to apply the blind signature to smart grid to achieve conditional anonymity so that malicious users can be identified. Li et al. [16] further proposed a scheme using conditional anonymous group blind signatures to protect data privacy in smart grids. Yu et al. [25] proposed an EC-ElGamal encryption supporting double trapdoor decryption in smart grid to ensure data privacy.

Due to the limited computing power of smart meter, many studies have proposed lightweight algorithms to protect data privacy at the smart meter side. Liu et al. [17] proposed a lightweight authentication communication scheme, which uses XOR operation to encrypt data and Lagrangian interpolation for identity authentication. However, the XOR operation does not provide sufficient security, so some schemes have been proposed to improve the security of their scheme while ensuring data privacy with only lightweight operations.

Zhang et al. [26] designed a lightweight anonymous authentication key agreement scheme for smart meters and service providers, which simultaneously realizes authentication and key sharing. Moghadam et al. [5] established a low-cost and secure two-way handshake communication using an ECC-based authentication and key exchange protocol. Gope et al. [8,9] used a physically unclonable function (PUF) to provide a smart meter-to-service provider authenticated key exchange protocol. The difference is that [9] solves the security problem of the modeling attack faced by [8]. Cao et al. [1] also used PUF to implement a lightweight privacy-preserving authentication data collection scheme. The experiments show that this scheme has high efficiency and low communication cost.

As an emerging technology, blockchain has the advantages of decentralization, data immutability, and traceability. Zhang et al. [27] used blockchain to realize secure signing of multi-party electronic contracts. Their scheme also uses an identity-based encryption algorithm to ensure the confidentiality of the contract and fairness in signing contracts. Wen et al. [23] constructed a blockchain supervision framework in a multi-party environment, which uses a double-chain structure to supervise the data in the blockchain. Blockchain technology can also be employed to realize privacy-preserving and verifiable billing in smart grid systems [28]. Gao et al. [7] proposed using blockchain technology to monitor smart grid, which can ensure user data privacy and transparently provide users with electricity consumption details. Gai et al. [6] adopted consortium chains to protect the privacy of energy transactions in smart grids.

However, the scalability of the blockchain is still a problem to be considered. If the storage space of all nodes is used to store data, the vast data redundancy will cause a significant burden on the blockchain nodes. Wang et al. [21] proposed an on-chain and off-chain coordination management system based on a consortium blockchain. In this system, the blockchain only stores the hash value and response records of the data, while a large amount of raw data is stored in an off-chain database. Wang et al. [22] designed a blockchain-based smart grid data sharing scheme, which provides immutability and transparency through cloud and blockchain as off-chain storage space and authentication platform,

respectively. Although off-chain storage can solve the blockchain scalability problem, the security provided by on-chain storage cannot be replaced entirely. For example, there is no guarantee that data stored off-chain has not been tampered with.

2 Preliminaries

2.1 Blockchain

Blockchain is a peer-to-peer network technology for building and maintaining a distributed ledger or database of records [20]. Participants in the blockchain need to be verified by a specific consensus mechanism before uploading data to the blockchain for storage. The data blocks on the blockchain are linked in the chronological order of their respective generation in the form of chains and are copied and stored on different nodes. It has the characteristics of decentralization, immutability, and so on. With these characteristics, blockchain can ensure the reliability of recorded data.

Smart contract was first proposed by Nick Szabo [19] in 1994, which is essentially a piece of software code on the blockchain. It usually runs in a virtual environment, and the application can interact with the smart contract through the virtual machine's interface. The smart contract will strictly execute this code to complete the operations defined by the code.

2.2 Mathematical Difficulties on Elliptic Curves

Discrete Logarithm (DL) Problem. Let G be an elliptic curve group of prime order q, and P be a generator of G. Given a tuple (x, xP) with unknown $x \in Z_q^*$, computing x is difficult in any polynomial algorithm.

Decision Diffie-Hellman (DDH) Problem. Let G be an elliptic curve group of prime order q, and P be a generator of G. Given $X = xP$ and $Y = yP$ with unknown $x, y \in Z_q^*$, determining whether $Q = xyP \in G$ holds is difficult in any polynomial algorithm.

3 System Model and Security Requirements

This section presents the system model of PPBME and summarizes its security requirements.

3.1 System Model

As shown in Fig. 1, a PPBME system consists of five types of entities, namely, key generation center (KGC), smart meter (SM), energy company (EC), cloud and blockchain (BC).

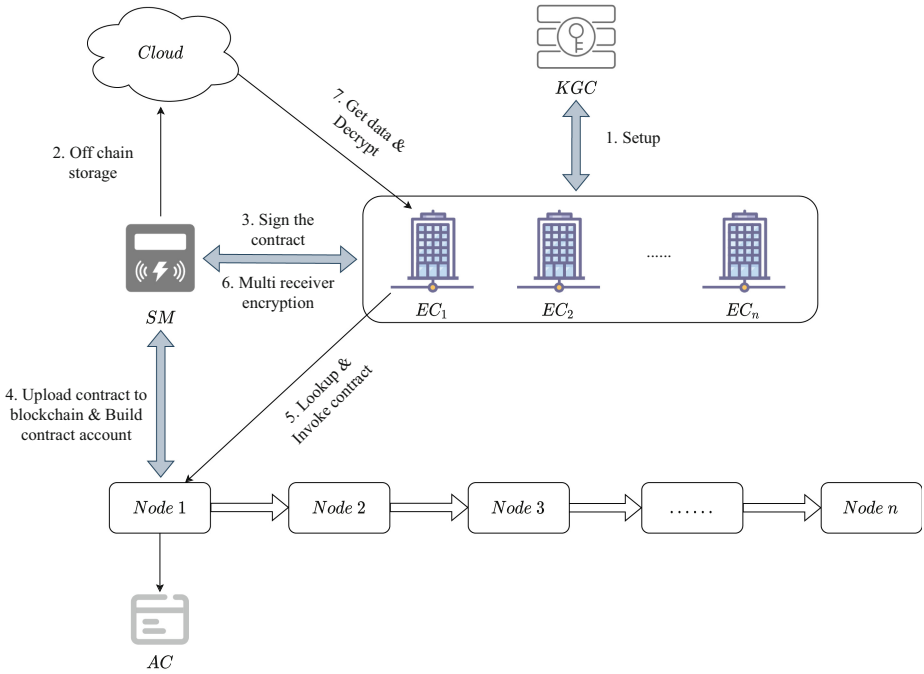

Fig. 1. System model of PPBME.

- **KGC:** KGC is a trusted third party that is mainly responsible for generating and distributing members' public and private keys.
- **SM:** The smart meter belongs to the sender in the system, which has the ability to collect energy data from various household energy-consuming appliances, is responsible for generating data information, and can generate ciphertext using the recipients' identities and public keys. Also, each smart meter can create a contract account AC, so that all nodes in the blockchain can access to pay privacy compensation.
- **EC:** The energy company belongs to the receiver in the system and is an energy service provider in the smart grid. This entity hopes to obtain the data collected by the user's smart meter to improve service quality and reduce costs. It can decrypt the ciphertext with its private key. By paying for contract accounts, energy companies can compensate users for privacy loss when sharing data.
- **Cloud:** The cloud is responsible for storing user energy data in encrypted form, and providing cloud storage services for users to store data off-chain to improve system efficiency.
- **BC:** The blockchain is responsible for storing data such as the transaction between SM and EC. Note that the original data is not stored on blockchain.

3.2 Security Requirements

The PPBME system is a system based on blockchain and multi-receiver encryption technology, which can ensure the privacy of user data during data sharing. It needs to satisfy the following requirements.

- *Privacy of user data*: Before data sharing, SM will store user energy data on the public cloud. Only two entities, the SM and the service provider who purchased the data, during data sharing, can access the user energy data stored on the cloud. No entity can tamper or falsify the stored data.
- *Resist denial and fraud attacks*: During data sharing between smart meter and the service provider, service providers cannot deny or refuse to pay for data purchases.
- *Resist replay attacks*: No malicious entity can obtain some confidential information by replaying the data tuples transmitted between the smart meter and the service provider.
- *Resist privileged-insider attack*: When other entities apply to KGC for keys, malicious users inside KGC cannot infer the private key of any entity based on the data tuples sent or received in the secure channel at this stage.
- *Access control*: Unauthorized service providers cannot obtain user energy data in any way.

3.3 System Framework

A PPBME construction consists of eight procedures, namely, Setup, $\text{KeyGen}_{\text{EC}}$, $\text{KeyGen}_{\text{SM}}$, OffChainSto, TransApp, TransPro, MulEnc, and Dec.

- $\mathbf{Setup}(1^{\lambda}) \rightarrow \Omega$: The system setup procedure is executed by KGC, which takes the security parameter 1^{λ} as input and outputs the system public parameters Ω.
- $\mathbf{KeyGen_{EC}}(\Omega, ID_i) \rightarrow (pk_i, sk_i, d_{EC_i}, X_{EC_i})$: The key generation procedure of EC is executed by KGC, which takes the system public parameters Ω and EC's identity ID_i as input, and outputs the public-private key pair (pk_i, sk_i) and signing key pair (d_{EC_i}, X_{EC_i}) for EC.
- $\mathbf{KeyGen_{SM}}(\Omega, ID_{SM}) \rightarrow (d_{SM}, X_{SM})$: The key generation procedure of SM is executed by KGC, which takes the system public parameters Ω and SM's identity ID_{SM} as input and outputs the signing key pair (d_{SM}, X_{SM}) for SM.
- $\mathbf{OffChainSto}(\Omega, M) \rightarrow (k, CK, site, D)$: Off-chain storage procedure is executed by SM, which takes the system public parameters Ω and user energy data M as input and outputs symmetric key k, ciphertext CK of data M under symmetric key k, the storage transaction D and the storage location *site*.
- $\mathbf{TransApp}(\Omega, ID_i, contract_i) \rightarrow (Y_i, O)$: The transaction application procedure is executed by EC and SM. EC takes the system public parameter Ω, EC's identity ID_i and purchase contract $contract_i$ as input, outputs the transaction request data Y_i, and sends Y_i to SM. SM takes the system public

parameter Ω and the transaction request data Y_i of EC as input and outputs transaction data O.

- **TransPro**$(\Omega, O) \rightarrow (\{L_i\}_{i=1}^n, AC, \delta_{com})$: The transaction processing procedure is executed by SM, which takes the system public parameters Ω and transaction data O as input. SM initiates a smart contract through the transaction data O, and the smart contract creates the transaction $\{L_i\}_{i=1}^n$ of both parties. When all nodes successfully verify the transaction through the consensus algorithm, the transaction will be recorded in the blockchain. At the same time, SM creates a contract account AC that all nodes in the blockchain can access. EC pays enough contract currency to contract account AC as "compensation", AC closes the payment channel, changes the status of the contract account, and returns a payment completion flag δ_{com}.
- **MulEnc**$(\Omega, \{ID_i\}_{i=1}^n, \{pk_i\}_{i=1}^n, k) \rightarrow CT$: The multi-receiver encryption procedure is executed by SM, which takes the system public parameters Ω, symmetric key k, the identities of all participant ECs $\{ID_i\}_{i=1}^n$ and their public keys $\{pk_i\}_{i=1}^n$ as input and outputs the ciphertext CT under the symmetric key k and a multi-receiver encryption scheme.
- **Dec**$(\Omega, CT, site, pk_i, sk_i) \rightarrow (k, M)$: The decryption procedure is executed by EC, which takes the system public parameters Ω, ciphertext CT, EC's public-private key pair (pk_i, sk_i) and the data storage location $site$ in the cloud as input. The EC decrypts the ciphertext CT using the multi-receiver encryption scheme to obtain the symmetric key k, then finds the storage transaction D according to the storage location $site$ in the cloud, and finally decrypts the ciphertext CK in D with the symmetric key k to obtain the user energy data M.

A correct PPBME construction should satisfy the following conditions: If all participants faithfully follow the procedures, then

- Each EC can successfully validate the key pair generated by KGC.
- Each EC can successfully decrypt the ciphertext generated by SM.

4 PPBME Construction

This section introduces a PPBME construction.

4.1 System Initialization

Setup. KGC takes the security parameter 1^λ as input and outputs $p, q, E, G, G_p, G_q, P, Q$, where p, q are two distinct prime integers, E is an elliptic curve defined on F_p, G is the additive group on the elliptic curve E, G_p is a subgroup of G with prime order p, G_q is a subgroup of G with prime order q, $P \in G_p$ is a generator of G_p, and $Q \in G_q$ is a generator of G_q.

KGC chooses a random integer $x \in Z_p^*$ as the master private key, and calculates $P_{pub} = x \cdot P$ as the corresponding public key. KGC selects five collision-resistant hash functions $H_i : \{0,1\}^* \rightarrow Z_p^*$ for $i = 1, 2, 3$, $H_4 : \{0,1\}^* \rightarrow Z_q^*$ and $H_5 : \{0,1\}^* \rightarrow \{0,1\}^{l_1}$, where l_1, l_2 are determined by the security parameter λ. KGC chooses a secure multi-receiver encryption algorithm I, a secure symmetric encryption scheme $\Pi = (KeyGen, Enc, Dec)$ (e.g., AES) and a function $F(\cdot)$ that maps from point to value [13]. KGC publishes the system public parameters $\Omega = \{p, q, E, G, G_p, G_q, P, P_{pub}, Q, H_1, H_2, H_3, H_4, H_5, I, \Pi, F, l_1, l_2\}$.

KeyGen$_{\mathbf{EC}}$. Each EC_i randomly selects an integer $t_i \in Z_p^*$, and calculates

$$T_i = t_i \cdot P$$

Then, EC_i sends dataset $set_1 = \{T_i, ID_i\}$ to KGC, where ID_i is EC_i's identity.

KGC computes the encryption and signing keys for EC_i according to the dataset set_1 and the system parameters Ω. KGC randomly selects integers $r_i, v_i \in Z_p^*$, $d_{EC_i} \in Z_q^*$, calculates

$$\begin{aligned} R_i &= r_i \cdot P \\ V_i &= v_i \cdot T_i \\ X_{EC_i} &= d_{EC_i} \cdot Q \\ b_i &= H_1(R_i \| V_i \| ID_i) \\ c_i &= r_i + b_i x \pmod{p} \end{aligned}$$

and sends the dataset $set_2 = \{R_i, V_i, c_i, v_i, d_{EC_i}, X_{EC_i}\}$ to EC_i through a secure channel.

After receiving the dataset set_2, EC_i first verifies it by checking the following equality

$$R_i + H_1(R_i \| V_i \| ID_i) P_{pub} \stackrel{?}{=} c_i \cdot P \tag{1}$$

If it holds, set_2 is valid, and then EC_i calculates $u_i = t_i \cdot v_i$. Thus, the public key of EC_i is $pk_i = (R_i, V_i)$, and the private key is $sk_i = (c_i, u_i)$. At the same time, EC_i obtains the public-private key pair (d_{EC_i}, X_{EC_i}) for signing.

KeyGen$_{\mathbf{SM}}$. Similarly, SM applies to KGC for a pair of signing keys, sends the SM's identity ID_{SM} to KGC.

KGC randomly selects an integer $d_{SM} \in Z_q^*$, calculates

$$X_{SM} = d_{SM} \cdot Q$$

and sends the dataset $set_4 = \{d_{SM}, X_{SM}\}$ to SM through a secure channel.

Therefore, SM obtains the public-private key pair (d_{SM}, X_{SM}) for signing.

4.2 Off-Chain Storage

With the secure symmetric encryption scheme Π, SM generates the symmetric key k, and encrypts the user energy data M to obtain ciphertext CK as follows.

$$k \leftarrow \Pi.KeyGen(1^{\lambda})$$
$$CK = \Pi.Enc_k(M)$$

SM generates current timestamp $\mathcal{T}_1$, selects a random integer $\dot{a} \in Z_q^*$, and calculates a signature tuple $\dot{\sigma}_{SM} = (\dot{A}, \dot{B})$ as follows.

$$\dot{A} = F(\dot{a} \cdot Q) \pmod q$$
$$\dot{B} = \dot{a} - H_4(CK \| \mathcal{T}_1) \cdot d_{SM} \pmod q$$

SM outputs a storage transaction $D = \{CK, \mathcal{T}_1, \dot{\sigma}_{SM}\}$. Then, SM stores D on the public cloud and records the data storage location *site*.

4.3 Transaction Processing

TransApp. Each EC_i generates a purchase contract $contract_i$ and current timestamp $\mathcal{T}_{2,i}$, selects a random integer $a_i \in Z_q^*$, and calculates a signature tuple $\sigma_{EC,i} = (A_i, B_i)$ as follows.

$$A_i = F(a_i \cdot Q) \pmod q$$
$$B_i = a_i - H_4(contract_i \| \mathcal{T}_{2,i} \| ID_i \| X_{EC_i}) \cdot d_{EC_i} \pmod q$$

EC_i sends a request data Y_i to SM, where

$$Y_i = \langle contract_i, \mathcal{T}_{2,i}, ID_i, X_{EC_i}, \sigma_{EC,i} \rangle$$

After receiving the request data Y_i, SM first checks whether $\mathcal{T}_{now} - \mathcal{T}_{2,i} \leq \varepsilon$ holds, where $\mathcal{T}_{now}$ is the current time, and ε is the maximum transmission delay. Then, if SM approves the purchase contract of EC_i, it returns an approval response and generates a timestamp $\mathcal{T}_3$. Next, SM selects a random integer $\ddot{a} \in Z_q^*$, and calculates a signature tuple $\ddot{\sigma}_{SM} = (\ddot{A}, \ddot{B})$ as follows.

$$\ddot{A} = F(\ddot{a} \cdot Q) \pmod q$$
$$\ddot{B} = \ddot{a} - H_4((Y_1, Y_2, \cdots, Y_n) \| \mathcal{T}_3 \| X_{SM}) \cdot d_{SM} \pmod q$$

Then, SM outputs the transaction data O, where

$$O = \langle Y_1, Y_2, \cdots, Y_n, \mathcal{T}_3, X_{SM}, \ddot{\sigma}_{SM} \rangle$$

TransPro. SM initiates a smart contract through O, as shown in Algorithm 1. The smart contract verifies the signature in O as follows

$$F(\ddot{B} \cdot Q + H_4((Y_1, Y_2, \cdots, Y_n) \| \mathcal{T}_3 \| X_{SM}) \cdot X_{SM}) \stackrel{?}{=} \ddot{A} \tag{2}$$

and, for each Y_i, checks the signature as follows.

$$F(B_i \cdot Q + H_4(contract_i \| \mathcal{T}_{2,i} \| ID_i \| X_{EC_i}) \cdot X_{EC_i}) \stackrel{?}{=} A_i \tag{3}$$

If all are satisfied, the transaction L_i between EC_i and SM is constructed and recorded in the blockchain. At the same time, a contract account AC is created, and all nodes in the blockchain can access the contract account.

Algorithm 1. Create Transaction and Contract Accounts

Input: $O = \langle Y_1, Y_2, \cdots, Y_n, \mathcal{T}_3, X_{SM}, \ddot{\sigma}_{SM} \rangle$
Output: $L_i; AC; \delta_{com}$
1: **if** $Time.now() - \mathcal{T}_3 \leq \varepsilon$ **Then**
2: **Verify** $\langle Y_1, Y_2, \cdots, Y_n \| \mathcal{T}_3 \| X_{SM} \rangle = Ver_{X_{SM}}(\ddot{\sigma}_{SM})$ is true
3: **end if**
4: $T_{trans} \leftarrow Time.now(); value = 0; status = 1$
5: **While** $i \leq n$ **do**
6: $i := i + 1$
7: **Verify** $(contract_i \| \mathcal{T}_{2,i} \| ID_i \| X_{EC_i}) = Ver_{X_{EC_i}}(\sigma_{EC,i})$ is true **Then**
8: $L_i \leftarrow (Y_i, value_{contract_i}, T_{trans})$
9: $AC \leftarrow (value, status)$
10: $p \leftarrow AC.getpayment()$
11: $AC.updata(value = value + p)$
12: **if** $Time.now() - T_{trans} \leq \varepsilon$ **Then**
13: L_i completed
14: **Verify** $value \geq \sum value_{contract_i}$ is true **Then**
15: Receive δ_{com} from Blockchain
16: **if Verify** $\delta_{com} = true$ **Then**
17: $AC.updata(status = 0)$
18: **end if**

EC_i synchronizes blockchain data and detects the status of contract account AC. EC_i needs to pay the purchase amount in the purchase contract $contract_i$ to the contract account AC. After AC receives the purchase amount, it determines whether the amount is sufficient. If satisfied, it changes the status of the AC and returns a payment completion flag δ_{com}.

4.4 Data Sharing

SM synchronizes blockchain data and checks the status and identity of the contract account AC. When all ECs have executed the contract, the SM encrypts the symmetric key k using a multi-receiver encryption scheme.

MulEnc. With the identities $\{ID_i\}_{i=1}^n$ of $\{EC_i\}_{i=1}^n$ and their public keys $\{pk_i\}_{i=1}^n$, SM encrypts the symmetric key k, and shares the ciphertext and cloud storage address with participating ECs as follows. SM randomly chooses $\omega \in \{0,1\}^{l_2}$, and computes

$$s = H_2(k\|\omega)$$
$$S = s \cdot P$$

For each EC_i, SM calculates

$$b_i = H_1(R_i\|V_i\|ID_i)$$
$$U_i = s \cdot (R_i + b_i P_{pub} + V_i)$$
$$\mu_i = H_3(U_i\|ID_i\|R_i\|V_i)$$

randomly chooses an integer $e \in Z_p^*$, and computes a polynomial $f(y)$ with degree n as follows.

$$f(y) = \prod_{i=1}^{n}(y - \mu_i) + e \pmod p$$

Next, SM calculates

$$\gamma = (k\|\omega) \oplus e$$
$$C = H_5(S\|e) \oplus H_5(k\|\omega)$$

and uploads ciphertext $CT = (S, f, \gamma, C)$ and the data storage location *site* to blockchain.

Dec. After EC_i obtains $\{CT, site\}$ from the blockchain, it finds the cloud storage transaction D through *site*. Then, EC_i verifies the timestamp on the transaction D, and checks the following equality

$$F(\dot{B} \cdot Q + H_4(CK\|\mathcal{T}_1) \cdot X_{SM}) \stackrel{?}{=} \dot{A} \tag{4}$$

If it holds, it indicates that D is a storage transaction created by SM; otherwise, D is an invalid transaction. After EC_i has successfully verified the signature, EC_i computes

$$U_i' = (c_i + u_i) \cdot S$$
$$\mu_i' = H_3(U_i'\|ID_i\|R_i\|V_i) \tag{5}$$
$$e = f(\mu_i') \pmod p$$

and checks the following equality

$$H_5(e \oplus \gamma) \stackrel{?}{=} H_5(S\|e) \oplus C \tag{6}$$

If it is not satisfied, the decryption process aborts; otherwise, EC_i outputs the symmetric key k.

EC_i decrypts the ciphertext CK in D and outputs the SM energy data M as follows.

$$M = \Pi.Dec_k(CK)$$

Theorem 1. *The proposed PPBME construction is correct.*

Proof. To prove the correctness of the proposed PPBME construction, it is necessary to prove that the Eqs. (1)–(6) are satisfied.

For the dataset set_2 issued by KGC, equality (1) satisfies as follow:

$$\begin{aligned} & R_i + H_1(R_i \| V_i \| ID_i) P_{pub} \\ =& r_i \cdot P + H_1(R_i \| V_i \| ID_i) x \cdot P \\ =& (r_i + H_1(R_i \| V_i \| ID_i) x) \cdot P \\ =& c_i \cdot P \end{aligned}$$

For the signature $\ddot{\sigma}_{SM}$ from SM, equality (2) satisfies as follow:

$$\begin{aligned} & F(\ddot{B} \cdot Q + H_4((Y_1, Y_2, \cdots, Y_n) \| \mathcal{T}_3 \| X_{SM}) \cdot X_{SM}) \\ & = F((\ddot{a} - H_4(Y_1, Y_2, \cdots, Y_n) \cdot d_{SM}) \cdot Q \\ & + H_4((Y_1, Y_2, \cdots, Y_n) \| \mathcal{T}_3 \| X_{SM}) \cdot X_{SM}) \\ & = F(\ddot{a} \cdot Q) \\ & = \ddot{A} \end{aligned}$$

The Eqs. (3) and (4) can be proved similar to the Eq. (2)

In order to decrypt the energy data M from ciphertext CK, the Eq. (5) satisfies as follow:

$$\begin{aligned} U_i' =& (c_i + u_i) \cdot S \\ =& (c_i + u_i) \cdot s \cdot P \\ =& s \cdot (c_i \cdot P + u_i \cdot P) \\ =& s \cdot ((r_i + b_i x) \cdot P + t_i \cdot v_i \cdot P) \\ =& s \cdot (r_i \cdot P + b_i x \cdot P + v_i \cdot T_i) \\ =& s \cdot (R_i + b_i P_{pub} + V_i) = U_i \end{aligned}$$

For U_i calculated by SM and U_i' calculated by EC_i, we have

$$f(\mu_i) = f(H_3(U_i \| ID_i \| R_i \| V_i)) = e \pmod{p}$$

and

$$f(\mu_i') = f(H_3(U_i' \| ID_i \| R_i \| V_i)) = e \pmod{p}$$

Therefore, $k \| \omega = e \oplus \gamma$, which means EC can correctly decrypt the symmetric key k and further decrypt ciphertext CK.

In order to verify the correctness of the ciphertext CT decryption, the Eq. (6) satisfies as follow:

$$H_5(e \oplus \gamma) = H_5(k \| \omega) = H_5(S \| e) \oplus C$$

5 System Security Analysis

This section analyzes the security and performance of the proposed PPBME construction.

Theorem 2. *Suppose the symmetric encryption scheme Π is secure. In the PPBME construction proposed in this paper, any entity (including the cloud) except the key owner cannot obtain or infer the original data stored on the cloud. Also, the key owner cannot maliciously modify the data stored in the cloud.*

Proof. In the off-chain storage phase of PPBME, only SM with the symmetric key k can decrypt the ciphertext stored in the cloud. Only by obtaining the symmetric key k and the data storage location *site* can the attacker decrypt the private data through the decryption algorithm $\Pi.Dec_k(\cdot)$. At the same time, SM needs to sign the ciphertext and the current timestamp and store them in the cloud together. If the encryption party maliciously modifies the data, it will be detected and traced in time.

Theorem 3. *In the proposed PPBME construction, the transaction application and processing between SM and EC can effectively resist denial and fraud attacks.*

Proof. In PPBME, transaction processing between EC and SM is performed by smart contracts in a prescribed manner. EC's transaction and data usage are publicly recorded in the blockchain ledger, which means EC cannot deny or refuse to pay compensation. If the EC has fraudulent or false transactions, the real identity of EC will be discovered and traced. Therefore, the PPBME system proposed in this paper can effectively resist denial and fraud attacks.

Theorem 4. *The proposed PPBME construction is resistant to replay attacks. Any adversary cannot extract some valuable information by replaying the request data Y or transaction data O transmitted between EC and SM.*

Proof. The PPBME construction is proposed in a synchronous environment, and the massages between EC and SM are processed with the current timestamp. Thus, any adversary cannot efficiently perform replay attacks on the system. If the adversary tries to launch a replay attack by pretending to be a participant, when the smart contract creates transaction information and contract accounts, it will detect whether there are old messages through the timestamp attached to the messages.

Theorem 5. *The proposed PPBME construction can resist the privileged insider attack.*

Proof. In the KeyGen phase, the internal attacker may know the dataset $set_2 = \{R_i, V_i, c_i, v_i\}$ sent by KGC to EC. However, it cannot obtain r_i, v_i, t_i and the master private key x if the DL assumption holds in probabilistic polynomial time. Also, the attacker cannot obtain u_i in EC's private key $sk_i = (c_i, u_i)$, without knowing t_i and v_i. Thus, the PPBME construction can resist privileged insider attacks.

6 Comparison and Analysis

In this section, we compare the proposed PPBME scheme with the ones proposed by Kumar et al. [15], He et al. [11], and Guan et al. [10] in terms of security, functional characteristics, and computational cost.

6.1 Comparison on Security and Functionality Features

In Table 1, the schemes in [10,11,15] and our PPBME scheme are compared under five attributes such as simulated attack, denial and fraud attacks, replay attack, privileged insider attack and access control. The analysis shows that only our PPBME scheme can resist all these attacks and support access control mechanism, thus, our PPBME construction is more secure than other related schemes in [10,11,15].

Table 1. Security and functional features.

Properties	[15]	[11]	[10]	PPBME scheme
Simulated attack	Yes	Yes	Yes	Yes
Denial and fraud attacks	No	No	Yes	Yes
Replay attack	Yes	No	No	Yes
Privileged insider attack	—	Yes	Yes	Yes
Access control	No	Yes	Yes	Yes

In Kumar et al.'s scheme [15], a session key is generated for each pair of participants in the smart grid who wish to trade to ensure the confidentiality of the communication between two parties. Their scheme also uses timestamps to prevent replay attacks. However, this scheme does not support supervision on transaction party and allow the trusted third party to register for the two parties, thus, malicious users may conduct fake transactions, or launch denial and fraud attacks. Also, in one-to-many transactions, the scheme of Kumar et al. [15] cannot achieve access control well.

He et al.'s scheme [11] achieves system privacy protection through a multi-receiver encryption scheme, where the original data was directly encrypted and transmitted. For big data, it would be a major problem for devices with limited computational resources such as smart meters. Also, their scheme cannot resist denial and fraud attacks among users.

Guan et al.'s scheme [10] provides users with fine-grained access control through ciphertext-policy attribute-based encryption (CP-ABE). The service provider can determine whether the data is the one to be purchased through an access policy verification. However, their scheme is designed in bilinear groups, which seriously decreases the efficiency of data encryption and decryption.

The PPBME scheme proposed in this paper can solve the shortcomings of [10,11,15]. Our PPBME scheme uses the blockchain to store transaction

information and smart contracts to control the completion of transactions, preventing the appearance of fraudulent or false transactions by participants and effectively resisting denial and fraud attacks. Also, we consider the scalability of the blockchain and store the encrypted data on the public cloud for off-chain storage. During transaction processing, the smart contract will authenticate the transaction parties and the transaction content to ensure that the transaction is valid, prevent the simulated attack and replay attack, and protect the security of transactions. Moreover, our PPBME scheme uses lightweight multi-receiver encryption technology to achieve access control and data privacy protection during transmission. Therefore, our PPBME scheme provides more security protection mechanisms than related one in smart grid.

6.2 Theoretical Analysis

As shown in Table 2, we analyze and compare the computing cost of our PPBME construction and the schemes in [10,11,15], where G_1, G_2 are multiplicative cyclic groups, and $\hat{G}$ is additive groups on nonsingular elliptic curves.

Suppose n ECs are wishing to obtain energy data from SM. In order to achieve energy data sharing, Kumar et al.'s scheme [15] requires SM to perform $3n$ multiplication operations on $\hat{G}$, $5n$ hash operations, $3n$ symmetric encryption operations and n XOR encryption operations. Therefore, the transaction and encryption time on the SM side is $3nT_{sm-\hat{G}} + 5nT_{gh} + 3nT_{Enc/Dec} + nT_{sc}$. The EC side needs to implement 5 multiplication operations on $\hat{G}$, 8 hash operations, one XOR operation and 3 symmetric encryption operations. Therefore, the transaction and decryption time on EC side is $5T_{sm-\hat{G}}+8T_{gh}+2T_{sc}+3T_{Enc/Dec}$.

In He et al.'s scheme [11], the SM side needs to perform $3n+1$ multiplication operations on $\hat{G}$, n addition operations on $\hat{G}$, $4n+2$ hash operations, n XOR encryption operation and one symmetric encryption operation. Therefore, the transaction and encryption time on SM side is $(3n+1)T_{sm-\hat{G}} + nT_{add-\hat{G}} + (4n+2)T_{gh} + nT_{sc} + T_{Enc/Dec}$. The EC side needs to implement 3 multiplication operations on $\hat{G}$, 7 hash operations, two XOR operations and one symmetric encryption operation. Therefore, the transaction and decryption time on EC side is $3T_{sm-\hat{G}} + 7T_{gh} + 2T_{sc} + T_{Enc/Dec}$.

In Guan et al.'s scheme [10], the SM side needs to perform $n+2$ exponentiation operations on G_1, one bilinear pair matching operation and two exponentiations on G_2. Therefore, the transaction and encryption time on SM side is $(n+2)T_{exp-G_1} + T_{bp} + 2T_{exp-G_2}$. EC side needs to implement $2n+3$ bilinear pairing operations, n exponentiation operations on G_2, one exponentiation operation on G_1, one hash operation and one ECDSA signing operation. Therefore, the transaction and decryption time on EC side is $(2n+3)T_{bp} + nT_{exp-G_2} + T_{exp-G_1} + T_{gh} + T_{sig}$.

Table 2. Calculation cost comparison.

Schemes		Smart meter side	Service provision side
[15]	K_1	—	$2T_{sm-\hat{G}} + T_{gh}$
	K_2	$2nT_{sm-\hat{G}} + 3nT_{gh} + nT_{Enc/Dec} + nT_{sc}$	$T_{sm-\hat{G}} + 4T_{gh} + T_{Enc/Dec}$
	K_3	$nT_{sm-\hat{G}} + 2nT_{gh} + 2nT_{Enc/Dec}$	$2T_{sm-\hat{G}} + 2T_{Enc/Dec} + 3T_{gh} + T_{sc}$
[11]	K_1	—	$T_{sm-\hat{G}}$
	K_2	$(3n+1)T_{sm-\hat{G}} + 4nT_{gh} + nT_{add-\hat{G}} + nT_{sc}$	—
	K_3	$2T_{gh} + T_{Enc/Dec}$	$2T_{sm-\hat{G}} + 7T_{gh} + 2T_{sc} + T_{Enc/Dec}$
[10]	K_1	—	$T_{gh} + T_{sig}$
	K_2	$(n+2)T_{exp-G_1}$	—
	K_3	$T_{bp} + 2T_{exp-G_2}$	$(2n+3)T_{bp} + nT_{exp-G_2} + T_{exp-G_1}$
PPBME scheme	K_1	—	$2T_{sm-\hat{G}} + T_{add-\hat{G}} + T_{gh}$
	K_2	$T_{Enc/Dec} + 2T_{sig}$	T_{sig}
	K_3	$(2n+1)T_{sm-\hat{G}} + 2nT_{add-\hat{G}} + (2n+3)T_{gh} + 2T_{sc}$	$T_{sm-\hat{G}} + 3T_{gh} + 2T_{sc} + T_{Enc/Dec} + T_{ver}$

Notes: K_1: System initialization, K_2: Off-chain storage and data processing, K_3: Data sharing. T_{bp}: Bilinear mapping time ($e : G_1 \times G_1 \rightarrow G_2$), T_{exp-G_1}: Exponentiation operation time on group G_1, T_{exp-G_2}: Exponentiation operation time on group G_2, $T_{sm-\hat{G}}$: Multiplication operation time of group $\hat{G}$, $T_{add-\hat{G}}$: Addition operation time of group $\hat{G}$, T_{gh}: Hash operation time, T_{sc}: XOR operation time, $T_{Enc/Dec}$: Symmetric encryption/decryption time, T_{sig}: ECDSA signature operation time, T_{ver}: ECDSA authentication operation time.

In our PPBME scheme, the SM side needs to perform $2n+1$ multiplication operations on $\hat{G}$, $2n$ addition operations on $\hat{G}$, $2n+3$ hash operations, two XOR encryption operations, one symmetric encryption operation and two ECDSA signing operations. Therefore, the transaction and encryption time on SM side is $(2n+1)T_{sm-\hat{G}} + 2nT_{add-\hat{G}} + (2n+3)T_{gh} + 2T_{sc} + T_{Enc/Dec} + 2T_{sig}$. EC side needs to implement 3 multiplication operations on $\hat{G}$, one addition operation on $\hat{G}$, 4 hash operations, two XOR operations, one ECDSA signing operation, one ECDSA verification operation and one symmetric encryption operation. Therefore, the transaction and decryption time on EC side is $3T_{sm-\hat{G}} + T_{add-\hat{G}} + 4T_{gh} + 2T_{sc} + T_{sig} + T_{ver} + T_{Enc/Dec}$.

6.3 Experimental Analysis

In this section, we evaluated the experimental performance of the proposed PPBME construction in an environment with a quad-core Xeon processor, 16G memory, and the 2021.3.3 version of the Goland software. The PBC and Crypto libraries are used to support crypto operations, where the elliptic curve is of Type E ($y^3 = x^3 + ax + b$) such that q and the element size in G are all 256 bits, and the encryption algorithm is AES-256-CBC. All hash functions are SHA-256. The experimental results of key operations are summarized in Table 3. According to Table 2 and Table 3, it is easy to get the running time at the smart meter side and the service provider side of these schemes, which are shown in Table 4.

Table 3. Running time of key operations.

Notations	Runtime (milliseconds)
T_{bp}	28.353
T_{exp-G_1}	1.352
T_{exp-G_2}	1.725
$T_{sm-\hat{G}}$	2.806
$T_{add-\hat{G}}$	0.016
T_{gh}	0.002
T_{sc}	0.001
$T_{Enc/Dec}$	0.042
T_{sig}	2.945
T_{ver}	5.871

Table 4. Running time (milliseconds).

Scheme	Smart meter side	Service provider side
Kumar et al. [15]	$8.555n$	14.174
He et al. [11]	$8.443n + 2.852$	8.476
Guan et al. [10]	$1.352n + 34.507$	$58.431n + 89.358$
PPBME scheme	$5.648n + 8.746$	17.302

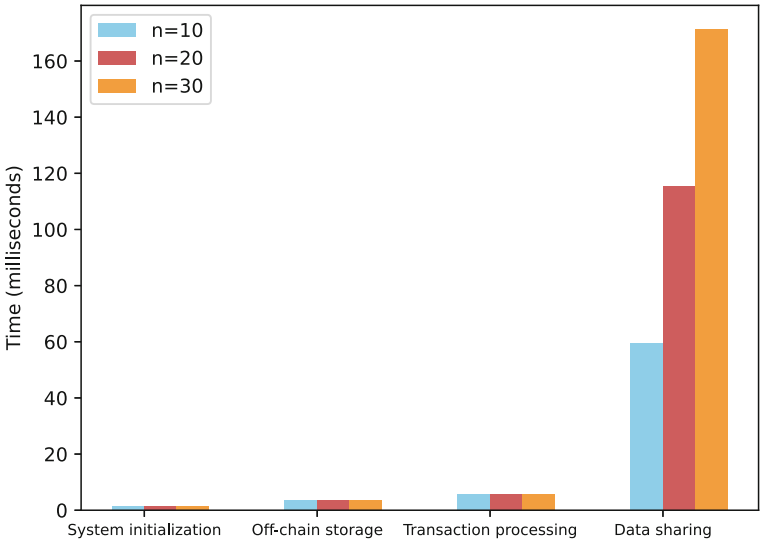

Fig. 2. Time cost at smart meter side.

We evaluated the time spent at the smart meter side of the proposed PPBME scheme in the system initialization phase, off-chain storage phase, transaction processing phase, and data sharing phase when the number of service providers $n = 10, 20, 30$, respectively, which are shown in Fig. 2. These results are con-

sistent with those in Table 4. Therefore, the proposed PPBME scheme enjoys higher efficiency than other related schemes.

7 Conclusion

In order to solve the privacy leakage problem caused by data sharing in smart grid systems, this paper proposed a privacy protection system based on blockchain and multi-receiver encryption. The elliptic curve encryption and off-chain storage technologies are used to improve the system's efficiency in processing data. Before data sharing, energy companies deposit the compensation to the contract account to compensate for the leakage of user data in the future. Users can achieve control access on energy data for energy companies through smart meters with a multi-recipient encryption technology. The security and performance analysis showed that the proposed PPBME construction is secure and efficient.

Acknowledgment. This article is supported in part by the National Key R&D Program of China under project 2020YFB1006004, the Guangxi Natural Science Foundation under grants 2019GXNSFFA245015 and 2019GXNSFGA245004, the National Natural Science Foundation of China under projects 62162017, 62172119, 61862012 and 61962012, and the Peng Cheng Laboratory Project of Guangdong Province PCL2021A09, PCL2021A02 and PCL2022A03.

References

1. Cao, Y.N., Wang, Y., Ding, Y., Zheng, H., Guan, Z., Wang, H.: A PUF-based lightweight authenticated metering data collection scheme with privacy protection in smart grid. In: 2021 IEEE International Conference on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 876–883 (2021). https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00124
2. Chen, J., Ren, X.: A privacy protection scheme based on certificateless aggregate signcryption and masking random number in smart grid. In: Proceedings of the 2016 4th International Conference on Mechanical Materials and Manufacturing Engineering, pp. 10–13. Atlantis Press, October 2016. https://doi.org/10.2991/mmme-16.2016.3
3. Colak, I.: Introduction to smart grid. In: 2016 International Smart Grid Workshop and Certificate Program (ISGWCP), pp. 1–5 (2016). https://doi.org/10.1109/ISGWCP.2016.7548265
4. Ding, Y., Wang, B., Wang, Y., Zhang, K., Wang, H.: Secure metering data aggregation with batch verification in industrial smart grid. IEEE Trans. Industr. Inf. **16**(10), 6607–6616 (2020). https://doi.org/10.1109/TII.2020.2965578
5. Farhdi Moghadam, M., Mohajerzdeh, A., Karimipour, H., Chitsaz, H., Karimi, R., Molavi, B.: A privacy protection key agreement protocol based on ECC for smart grid. In: Choo, K.-K.R., Dehghantanha, A. (eds.) Handbook of Big Data Privacy, pp. 63–76. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38557-6_4

6. Gai, K., Wu, Y., Zhu, L., Qiu, M., Shen, M.: Privacy-preserving energy trading using consortium blockchain in smart grid. IEEE Trans. Industr. Inf. **15**(6), 3548–3558 (2019). https://doi.org/10.1109/TII.2019.2893433
7. Gao, J., et al.: GridMonitoring: secured sovereign blockchain based monitoring on smart grid. IEEE Access **6**, 9917–9925 (2018). https://doi.org/10.1109/ACCESS.2018.2806303
8. Gope, P., Sikdar, B.: Privacy-aware authenticated key agreement scheme for secure smart grid communication. IEEE Trans. Smart Grid **10**(4), 3953–3962 (2019). https://doi.org/10.1109/TSG.2018.2844403
9. Gope, P., Sikdar, B.: A privacy-aware reconfigurable authenticated key exchange scheme for secure communication in smart grids. IEEE Trans. Smart Grid **12**(6), 5335–5348 (2021). https://doi.org/10.1109/TSG.2021.3106105
10. Guan, Z., Lu, X., Yang, W., Wu, L., Wang, N., Zhang, Z.: Achieving efficient and privacy-preserving energy trading based on blockchain and ABE in smart grid. J. Parallel Distrib. Comput. **147**, 34–45 (2021)
11. He, D., Wang, H., Wang, L., Shen, J., Yang, X.: Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices. Soft. Comput. **21**(22), 6801–6810 (2016). https://doi.org/10.1007/s00500-016-2231-x
12. Kabalci, E., Kabalci, Y.: Introduction to smart grid architecture. In: Kabalci, E., Kabalci, Y. (eds.) Smart Grids and Their Communication Systems. Energy Systems in Electrical Engineering, pp. 3–45. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1768-2_1
13. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. CRC Press, Boca Raton (2020)
14. Kong, W., Shen, J., Vijayakumar, P., Cho, Y., Chang, V.: A practical group blind signature scheme for privacy protection in smart grid. J. Parallel Distrib. Comput. **136**, 29–39 (2020)
15. Kumar, P., Gurtov, A., Sain, M., Martin, A., Ha, P.H.: Lightweight authentication and key agreement for smart metering in smart energy networks. IEEE Trans. Smart Grid **10**(4), 4349–4359 (2019). https://doi.org/10.1109/TSG.2018.2857558
16. Li, X., Sun, X., Li, F.: A group blind signature scheme for privacy protection of power big data in smart grid. In: Tan, Y., Shi, Y., Zomaya, A., Yan, H., Cai, J. (eds.) DMBD 2021. CCIS, vol. 1453, pp. 23–34. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-7476-1_3
17. Liu, Y., Cheng, C., Gu, T., Jiang, T., Li, X.: A lightweight authenticated communication scheme for smart grid. IEEE Sens. J. **16**(3), 836–842 (2016). https://doi.org/10.1109/JSEN.2015.2489258
18. Su, Z., Wang, Y., Xu, Q., Fei, M., Tian, Y.C., Zhang, N.: A secure charging scheme for electric vehicles with smart communities in energy blockchain. IEEE Internet Things J. **6**(3), 4601–4613 (2019). https://doi.org/10.1109/JIOT.2018.2869297
19. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997). https://doi.org/10.5210/fm.v2i9.548, https://firstmonday.org/ojs/index.php/fm/article/view/548
20. Underwood, S.: Blockchain beyond bitcoin. Commun. ACM **59**(11), 15–17 (2016). https://doi.org/10.1145/2994581
21. Wang, K., Yan, Y., Guo, S., Wei, X., Shao, S.: On-chain and off-chain collaborative management system based on consortium blockchain. In: Sun, X., Zhang, X., Xia, Z., Bertino, E. (eds.) ICAIS 2021. CCIS, vol. 1423, pp. 172–187. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78618-2_14

22. Wang, Y., et al.: SPDS: a secure and auditable private data sharing scheme for smart grid based on blockchain. IEEE Trans. Industr. Inf. **17**(11), 7688–7699 (2021). https://doi.org/10.1109/TII.2020.3040171
23. Wen, B., Wang, Y., Ding, Y., Zheng, H., Liang, H., Wang, H.: A privacy-preserving blockchain supervision framework in the multiparty setting. Wirel. Commun. Mob. Comput. **2021** (2021). https://doi.org/10.1155/2021/5236579
24. Yang, L., Xue, H., Li, F.: Privacy-preserving data sharing in smart grid systems. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 878–883 (2014). https://doi.org/10.1109/SmartGridComm.2014.7007759
25. Zhan, Y., Zhou, L., Wang, B., Duan, P., Zhang, B.: Efficient function queryable and privacy preserving data aggregation scheme in smart grid. IEEE Trans. Parallel Distrib. Syst. **33**(12), 3430–3441 (2022). https://doi.org/10.1109/TPDS.2022.3153930
26. Zhang, L., Zhao, L., Yin, S., Chi, C.H., Liu, R., Zhang, Y.: A lightweight authentication scheme with privacy protection for smart grid communications. Future Gener. Comput. Syst. **100**, 770–778 (2019)
27. Zhang, T., Wang, Y., Ding, Y., Wu, Q., Liang, H., Wang, H.: Multi-party electronic contract signing protocol based on blockchain. IEICE Trans. Inf. Syst. **E105.D**(2), 264–271 (2022). https://doi.org/10.1587/transinf.2021BCP0011
28. Zhao, M., Ding, Y., Tang, S., Liang, H., Wang, H.: A blockchain-based framework for privacy-preserving and verifiable billing in smart grid. Peer-to-Peer Netw. Appl. (2022). https://doi.org/10.1007/s12083-022-01379-4