# SIFChain: A Decentralized Framework for Secure Storage Sharing and Dynamic Access Control in Virtual Power Plants

Xiaochuan Xu[1]*, Xiao Xin[2], Jie Liu[3], Ruiqi Fang[4], Dekai Liu[5], Zhixin Li[6]
School of Cyber Science and Engineering, Southeast University, Nanjing, 211118, China[1, 4]
Shandong Future Group Co., Ltd, Jinan, 250100, China[2, 3, 6]
School of Information Science and Engineering, University of Jinan, Jinan, 250022, China[5]

***Abstract*—Virtual Power Plants (VPPs) face significant challenges in secure data management and sharing, including risks of centralized control, single points of failure, and dynamic access requirements among multiple stakeholders. To address these issues, this study proposes SIFChain, a decentralized framework that integrates Hyperledger Fabric, the InterPlanetary File System (IPFS), and a revocable Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme with collaborative key generation. Unlike existing solutions such as Filecoin or Storj, SIFChain introduces a dual-channel blockchain architecture that separates public operational data from sensitive attribute information, mitigating privacy leakage and access policy exposure. The framework achieves fine-grained, dynamic access control with forward and backward security through an enhanced CP-ABE mechanism. Experimental evaluation demonstrates that SIFChain provides scalable performance: data upload/download times increase linearly from 1 MB to 1 GB, blockchain transaction latency remains under 5 ms for typical operations (registration, access requests, policy updates), and attribute-based encryption/decryption overhead scales linearly with policy complexity. These results confirm the practicality of SIFChain for secure, cross-organizational data sharing in Virtual Power Plant ecosystems.**

***Keywords*—*Virtual Power Plant; access control; CP-ABE; IPFS; blockchain; distributed energy; privacy protection***

## I. Introduction

With the rapid advancement of renewable energy integration and smart grid technologies, Virtual Power Plants (VPPs) have emerged as critical components in modern energy systems [1]. VPPs aggregate distributed energy resources (DERs) such as solar panels, wind turbines, battery storage systems, and flexible loads into unified, controllable entities that participate in electricity markets and grid services [2]. These systems generate massive volumes of operational data daily, including generation forecasts, real-time power measurements, market bids, grid interactions, and participant settlement information [21].

According to energy industry analysis [3], well-managed VPPs can improve grid stability by 20-30% while reducing operational costs by 15-25%, significantly enhancing energy efficiency and renewable integration [22]. However, current VPP data management systems face multiple security and governance challenges. From a technical perspective, traditional VPP architectures often rely on centralized control systems, creating single points of failure and exposing critical energy infrastructure to risks of cyber-attacks, data tampering, and unauthorized access [4], [15]. The threat model includes malicious insiders, external adversaries targeting communication links, and compromised DER devices that could leak sensitive operational data. These vulnerabilities make it difficult to ensure data integrity and reliability in multi-stakeholder energy environments involving generators, aggregators, grid operators, regulators, and consumers.

Furthermore, from an access control standpoint, VPP data management is typically controlled by single entities or limited consortia, leaving other legitimate stakeholders without verifiable and open access to necessary operational and financial data. As energy markets become increasingly decentralized and dynamic, with frequent changes in participation, ownership structures, and regulatory requirements, the need for secure, fine-grained, and adaptable access control mechanisms has become paramount. The evolving nature of VPP collaborations—where participants may join or leave, assets may change ownership, and access privileges must adapt to real-time market conditions—requires fundamentally new approaches to data security and sharing.

Blockchain technology offers promising solutions to these challenges through its distributed ledger architecture, which authenticates and records data via consensus algorithms without requiring trusted third parties [5]. The immutable nature of blockchain records provides strong guarantees against data tampering, while smart contracts enable automated execution of energy agreements, settlements, and access policies. Hyperledger Fabric, as a permissioned blockchain framework [6], is particularly suitable for VPP applications due to its efficient consensus mechanisms, multi-channel architecture for data isolation, and support for complex organizational structures typical in energy ecosystems.

The InterPlanetary File System (IPFS) complements blockchain technology by providing decentralized storage capabilities, addressing the inherent storage limitations of blockchain systems for large-scale energy data [7]. By storing encrypted VPP operational data on IPFS and maintaining only cryptographic references on the blockchain, energy organizations can achieve both scalability and security while preserving data availability. Additionally, Attribute-Based Encryption (ABE) technology [8], particularly Ciphertext-Policy Attribute-

*Corresponding author.

Based Encryption (CP-ABE) [9], enables fine-grained access control by allowing data owners to define access policies based on user roles, organizational affiliations, and operational contexts rather than specific identities.

Despite these advancements, existing decentralized storage systems such as Filecoin and Storj rely on proof-of-storage mechanisms that introduce high computational overhead and lack fine-grained access control tailored to multi-stakeholder VPP environments. Moreover, typical IPFS-Ethereum integrations suffer from on-chain storage bottlenecks and lack privacy-preserving attribute management. Many systems still rely on centralized key management authorities or lack efficient mechanisms for access privilege revocation and updates as market conditions change [10], [11], [16], [17]. These limitations create security vulnerabilities and administrative overhead in practical VPP deployments where access requirements frequently evolve due to changing market participants [12], regulatory compliance needs, asset ownership transfers, and real-time operational conditions.

This study addresses these gaps by proposing SIFChain, a comprehensive framework for secure storage sharing and dynamic access control in Virtual Power Plants. Our solution integrates Hyperledger Fabric, IPFS, and an enhanced revocable CP-ABE scheme with collaborative key generation, which mitigates the risk of single points of failure in key distribution. The main contributions of this work are as follows:

- We propose SIFChain, an efficient dynamic storage sharing framework for VPP consortium blockchains that enables granular and flexible access control for sensitive energy data through an enhanced Ciphertext attribute-based encryption algorithm.
- We design a multi-channel architecture comprising public channels and attribute channels to address access policy and attribute leakage challenges in on-chain attribute encryption. This architecture enables separate storage of different data types (operational, financial, regulatory), ensuring enhanced privacy management through dedicated attribute nodes representing different VPP stakeholders.
- We modify the Certificate Authority (CA) structure to enable collaborative attribute key generation between the CA and attribute authorization nodes, enhancing the security of user attribute private key generation and distribution in distributed energy environments.
- We implement a comprehensive security strategy that combines symmetric encryption for IPFS-stored VPP data with attribute-based encryption for blockchain-managed access control, ensuring data security while facilitating efficient storage and sharing across organizational boundaries in energy ecosystems.

The remainder of this study is organized as follows: Section II reviews related woks. Section III details the specific implementation of the SIFChain framework in VPP contexts. Section IV presents experimental results and performance analysis. Finally, Section V concludes the study and outlines directions for future research.

## II. Related Work

Decentralized storage and access control for energy systems have attracted significant research interest. Filecoin and Storj provide blockchain-based storage markets but focus on proof-of-storage rather than fine-grained access policies, making them less suitable for dynamic multi-stakeholder VPPs. IPFS combined with Ethereum has been proposed for IoT data sharing [13], yet on-chain storage costs and lack of attribute-based privacy controls limit scalability. In the context of VPPs, consortium blockchains like Hyperledger Fabric have been explored for energy trading [11] and data management [14], but these works often assume static access rights and do not address attribute revocation or collaborative key generation.

CP-ABE schemes with revocation capabilities have been studied extensively [8], [9]. However, most implementations rely on a single authority for key distribution, creating a central point of trust. Collaborative key generation between multiple authorities, as in SIFChain, reduces this vulnerability. Furthermore, existing VPP data management frameworks seldom integrate IPFS for scalable storage alongside dynamic attribute-based access control [18]. Our work bridges this gap by proposing a holistic architecture that combines Fabric's permissioned channels, IPFS's content-addressed storage, and a revocable CP-ABE scheme with distributed key generation, thereby advancing the state-of-the-art in secure VPP data sharing.

## III. SIFChain Framework for Virtual Power Plants

### A. System Architecture and Stakeholders

Our solution enhances the storage and sharing of VPP data among multiple stakeholders, including distributed energy resource owners, VPP aggregators, grid operators, electricity market participants, regulators, and consumers. The system consists of four main entities: the Fabric consortium blockchain, the IPFS distributed file system, data owners (typically VPP operators or asset owners), and data consumers (various authorized stakeholders).

We enhance dynamic access control through the revocable Ciphertext-Policy Attribute-Based Encryption (CP-ABE) mechanism, where decryption key parameters are generated collaboratively by the Fabric Certificate Authority (Fabric-CA) and the attribute authorities $AA_l$ representing different stakeholder groups. This approach reduces the risk of data leakage and enhances security in the distributed energy environment.

For the consortium blockchain, we employed a dual-channel architecture that distinctly separates public operational data (e.g., aggregated power forecasts, market prices) from sensitive attribute data (e.g., individual asset performance, participant identities, financial settlements), encryption policies, and other privacy-related information, thereby ensuring enhanced data security. This solution also utilizes the IPFS distributed file system for secure and efficient storage of VPP operational data, including high-frequency sensor data, detailed asset performance logs, and historical market interactions. Data owners encrypt their sensitive VPP data and upload it to IPFS, creating a dedicated, secure IPFS network that guarantees data confidentiality while improving storage efficiency for large-scale energy datasets.

### B. Data Flow and Access Control Mechanism

In terms of data storage, IPFS only stores the ciphertext of the original VPP data, while the blockchain maintains the address in IPFS, along with symmetric keys and hash values of the original data. When an endorsing node in Hyperledger Fabric executes a smart contract for data access, it combines the data user attribute set with the channel anchor point to verify the ciphertext decryption. If the attribute set meets the established access policy (e.g., "Grid Operator AND Real-time Operations"), the consumer can successfully decrypt the ciphertext and gain access to the data; otherwise, the endorsing node will return a failure indicator, thereby denying the access request. This design facilitates secure data storage and fine-grained attribute-based access control tailored to VPP requirements [13].

### C. Relevant Symbols

To facilitate the discussion, we first present the required parameters and relevant symbols in Table I.

TABLE I. Notation for SIFChain in VPP Context

| Notation | Description |
|---|---|
| $PP$ | System common parameter |
| $ID_i$ | ID of user $i$ in the VPP system |
| $AA_l$ | Attribute authorization node $l$ representing a stakeholder group |
| $S$ | Collection of attributes (roles, organizations, access levels) |
| $PK$ | Attribute encryption public key in the system |
| $MSK$ | Attribute encryption master key in the system |
| $\omega$ | Attribute set of user $ID_i$ in the system |
| $SK_{ID_i,\omega}$ | Attribute private key of user $ID_i$ |
| $m$ | Original sensitive VPP data (sensor readings, bids, etc.) |
| $H_1$ | Hash value of the original sensitive data |
| $CT_m$ | Symmetrically encrypted ciphertext of the original data |
| $URL$ | Address where IPFS stores encrypted data |
| $k$ | Symmetric key used to encrypt data |
| $m_1$ | New data formed by combining $URL$, $k$, and $H_1$ |
| $R_l$ | Revocation lists maintained in the system |
| $P_\omega^*$ | Access policy for attribute encryption |
| $CT_{ABE}$ | Ciphertext encrypted with data $m_1$ attribute |
| $CT_{ABE}^{`}$ | Recalculated attribute encrypted ciphertext |
| $leaf_j$ | Access the leaf node $j$ in the policy |
| $Root$ | Access the root node in the policy |
| $x$ | Accessing non-leaf nodes in a policy |
| $z$ | Accessing child nodes of non-leaf nodes |
| $H_2$ | Hash value computed by the data user for verification |
| $DO$ | Data owner in the system (VPP operator/asset owner) |
| $DU$ | Data users in the system (various stakeholders) |
| $CA$ | Certificate Management Center in Fabric |

### D. System Model for VPP Applications

The proposed storage sharing system for Virtual Power Plants is illustrated in Fig. 1. The Fabric consortium blockchain features two core channels: a public channel for operational data and an attribute channel for sensitive information [14], each with its own distinct ledger.

We classify nodes into two categories: public nodes (maintaining the public ledger with grid and market data), and attribute nodes (joining the attribute channel to manage attribute data for different stakeholder groups).

Before VPP data is uploaded to the blockchain, the data owner predefines the access policy based on energy market rules and regulatory requirements. The sensitive VPP data is encrypted using a symmetric algorithm and subsequently stored in IPFS. The IPFS address, symmetric encryption key, and data hash are then uploaded to the blockchain using attribute encryption. In the blockchain, the ciphertext and revocation lists are maintained as public data within the public channel, while sensitive information—such as encryption policies, participant attributes, and financial details—is stored in the attribute channel, managed by attribute nodes representing different stakeholder interests. Data within the same channel is synchronized in real-time across all relevant nodes.

When a data user (e.g., grid operator, regulator, market participant) requests access to VPP data, the attribute node evaluates the user's attribute set within the attribute channel utilizing Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to verify whether the consumer satisfies the access policy. If verification succeeds, the system performs decryption and grants access; otherwise, a failure message is returned. Access rights in VPPs may change frequently due to market conditions, regulatory updates, or participant status changes. If a consumer loses privileges, the system uses the revocation list to re-encrypt the ciphertext with a random value, replacing the original to ensure both forward and backward secrecy [19].

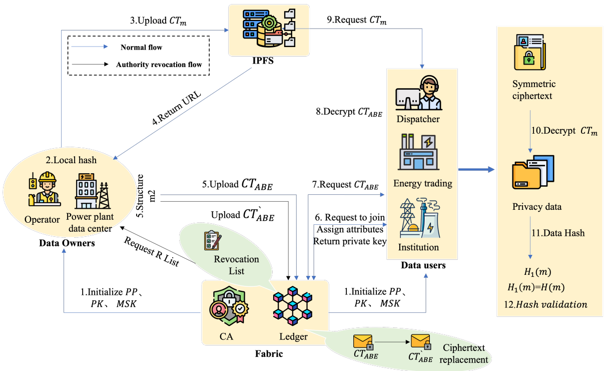


Fig. 1. SIFChain system model for Virtual Power Plants.

### E. Components of the Solution

This solution consists of six stages: system initialization, key generation, data encryption and storage, data encryption upload, attribute revocation, and decryption. The specific operations are detailed as follows:

*1) System initialization:* The Fabric-CA selects a bilinear mapping $e : G_1 \times G_1 \rightarrow G_2$, where $G_1, G_2$ are two cyclic groups of order $q$. Then, it selects generators $g, h, v \in G_1$ and random numbers $\alpha, \beta \in Z_q$, and computes $Z = e(g, g)^\beta, v^\beta$. Finally, it outputs the system public parameters $PP = (G_1, G_2, e, g, h, v, Z)$.

For each authenticated VPP participant $ID_i$ in the system, the CA selects a unique secret parameter $\lambda_i \in Z_q$ and securely sends $(\lambda_i, \beta)$ to the user, while sending $(\alpha, h)$ to the attribute authorization node $AA_l$ in Fabric, where $l \in \{1, \ldots, n\}$ is the index of each attribute authorization node, and $n$ is the total number of attribute authorization nodes in the system.

Define the attribute domain $S$ representing VPP roles and permissions (e.g., "Grid Operator", "Market Participant", "Asset Owner", "Regulator"). For any attribute $j \in S$, if the attribute authorization node $AA_l$ holds the key distribution authority for it, i.e., $j \in AA_l$, then $AA_l$ selects a secret random number $t_j \in Z_q^*$, computes $T_j = g^{t_j}$, and finally outputs the attribute public key $PK = (\{T_j\}_{j\in AA_l})_{l\in(1,2,...,n)}$ and the attribute master secret key $MSK = (\{t_j\}_{j\in AA_l})_{l\in(1,2,...,n)}$.

*2) Key generation:* A VPP participant $ID_i$ with attribute set $\omega$ (e.g., $\omega = \{\text{Asset Owner}, \text{Solar Farm}, \text{North Region}\}$) sends $\lambda_i$ to the attribute authority node $AA_l$ to request the corresponding private key. For any attribute $j \in \omega \cap AA_l$, the computation is conducted as follows:

$$S_{i,j} = g^{\frac{\alpha\lambda_i}{t_j}} h^{\frac{\alpha}{t_j}}, \quad W_i = g^{\alpha\lambda_i} \tag{1}$$

After receiving $S_{i,j}$ and $W_i$, the user computes $W_i^* = W_i \cdot g^{-\beta}$. The final private key for VPP participant $ID_i$ is:

$$SK_{ID_i,\omega} = (\lambda_i, W_i^*, S_{i,1}, \ldots, S_{i,\|\omega\|}) \tag{2}$$

In this process, since each attribute authority node $AA_l$ cannot obtain the secret parameter $\beta$, and the Certificate Authority (CA) cannot access the attribute key $t_j$, it is ensured that untrusted attribute authority nodes and the system center cannot decrypt the user's ciphertext.

*3) Data encryption and storage:* The VPP data owner employs the Secure Hash Algorithm 256 (SHA-256) hash function to hash the original data $m$ (e.g., power generation data, market bids), denoted as $H(m) \rightarrow H_1$, where $H_1$ is utilized for subsequent verification. The data owner symmetrically encrypts the data as $Enc(k, m) \rightarrow CT_m$, where $k$ is the encryption key, and $CT_m$ is the resulting ciphertext. $CT_m$ is uploaded to IPFS for storage.

Upon receiving $CT_m$, IPFS evaluates the size of the data. If the size exceeds 256 KB, it performs chunking on the data and stores the chunks across nodes located around the world. This process generates a unique hash value based on the content, referred to as the storage address, represented as a $URL$. Finally, the $URL$ is returned to the data owner.

The data owner executes a concatenation operation, specifically $m_1 = (URL\|k\|H_1)$, where $m_1$ represents the new data, $URL$ is the unique address returned by IPFS, $k$ is the symmetric encryption key, and $H_1$ is the hash value.

*4) Data encryption and upload:* Obtain the latest attribute revocation list from all attribute authority nodes $AA_l$, represented as $R_l = \{(j, (\lambda_{rev}, L_{j,rev})_{rev\in List_j})\}$. The data owner selects random numbers $s, r \in Z_q^*$ and calculates:

$$C_0 = m_1 \cdot Z^{sr}, \quad C^* = g^{sr} \tag{3}$$

The data owner selects an access tree policy $P_{\omega^*}$ that defines which attributes are required to access the VPP data. Starting from the root node, an $n_x - 1$ degree polynomial $q_x$ is selected for each internal node from top to bottom, as detailed in the access policy generation algorithm shown in Fig. 2.

For each leaf node $leaf_j$ in the access tree $P_{\omega^*}$, compute:

$$C_{1,j} = T_j^{r\cdot q_{leaf_j}(0)}, \quad C_{2,j} = L_j^{r\cdot q_{leaf_j}(0)} \tag{4}$$

where, $L_j = v \cdot g^{t_j}$.

For attributes that have been revoked (i.e., $j \in \omega^*, List_j \neq \emptyset$), compute:

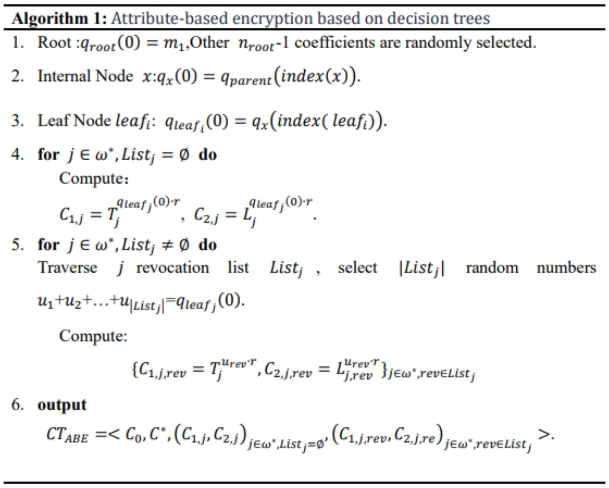
$$C_{1,j,rev} = T_j^{r\cdot(q_{leaf_j}(0)-u_{rev})}, \quad C_{2,j,rev} = L_j^{r\cdot(q_{leaf_j}(0)-u_{rev})} \tag{5}$$

where. $u_{rev}$ is a random number associated with the revoked user.

The final attribute-based encrypted ciphertext is:

$$\begin{aligned} CT_{ABE} = \langle C_0,\, C^*,\, \{C_{1,j},\, C_{2,j}\}_{j\in\omega^*,\, \text{List}_j=\emptyset}, \\ \{C_{1,j,\text{rev}},\, C_{2,j,\text{rev}}\}_{j\in\omega^*,\, \text{rev}\in\text{List}_j}\rangle \end{aligned} \tag{6}$$

Once the data owner encrypts $m_1$, they upload $CT_{ABE}$, along with the encryption policy $P_{\omega^*}$, to the blockchain. The attribute-encrypted ciphertext is stored in the public channel, while the access policy and user attributes are securely stored in the attribute channel. The public channel also holds the revocation list for managing access permissions. The comprehensive architecture for this data encryption and upload process is illustrated in Fig. 3.



```
Algorithm 1: Attribute-based encryption based on decision trees
1. Root :$q_{root}(0) = m_1$,Other $n_{root}$-1 coefficients are randomly selected.
2. Internal Node $x$:$q_x(0) = q_{parent}(index(x))$.
3. Leaf Node $leaf_i$: $q_{leaf_i}(0) = q_x(index(leaf_i))$.
4. for $j \in \omega^*, List_j = \emptyset$ do
       Compute:
       $C_{1,j} = T_j^{q_{leaf_j}(0)\cdot r}$, $C_{2,j} = L_j^{q_{leaf_j}(0)\cdot r}$.
5. for $j \in \omega^*, List_j \neq \emptyset$ do
   Traverse $j$ revocation list $List_j$ , select $|List_j|$ random numbers
   $u_1$+$u_2$+…+$u_{|List_j|}$=$q_{leaf_j}(0)$.
       Compute:
           $\{C_{1,j,rev} = T_j^{u_{rev}\cdot r}, C_{2,j,rev} = L_{j,rev}^{u_{rev}\cdot r}\}_{j\in\omega^*,rev\in List_j}$
6. output
    $CT_{ABE}$ =< $C_0, C^*, (C_{1,j}, C_{2,j})_{j\in\omega^*,List_j=\emptyset}, (C_{1,j,rev}, C_{2,j,re})_{j\in\omega^*,rev\in List_j}$ >.
```

Fig. 2. Access policy generation algorithm for VPP data.

*5) Attribute revocation:* When a VPP participant loses decryption privileges or access rights change (e.g., a market participant leaves the VPP consortium), the data owner can re-encrypt the ciphertext without updating the decryption keys. The system supports two revocation scenarios:

*a) Case 1: Attribute not previously revoked:* ($j \in \omega^*, List_j = \emptyset$) Select a random number $u_{rev}^*$ and compute the re-encrypted ciphertext $CT^{'}_{ABE}$:

$$C_{1,j,rev} = C_{1,j} \cdot T_j^{-u_{rev}^*\cdot r} \tag{7}$$

$$C_{2,j,rev} = C_{2,j} \cdot L_j^{-u_{rev}^*\cdot r} = L_j^{(q_{leaf_j}(0)-u_{rev}^*)\cdot r} \tag{8}$$

$$C_{1,j,rev^*} = T_j^{u_{rev}^*\cdot r}, \quad C_{2,j,rev^*} = L_j^{u_{rev}^*\cdot r} \tag{9}$$

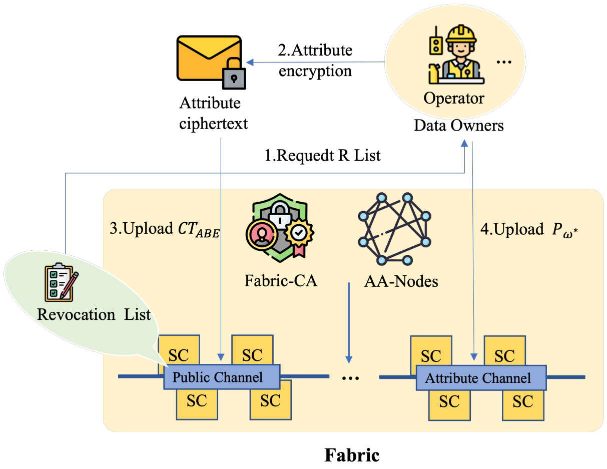


Fig. 3. Data encryption and upload architecture for VPP.

*b) Case 2: Attribute already revoked:* ($j \in \omega^*, List_j \neq \emptyset$) Select a random number $u^*_{rev}$ and compute the re-encrypted ciphertext $CT^{`}_{ABE}$:

$$C_{1,j,rev} = C_{1,j,rev} \cdot T_j^{-u^*_{rev} \cdot r} = T_j^{(u_{rev} - u^*_{rev}) \cdot r} \tag{10}$$

$$C_{2,j,rev} = C_{2,j,rev} \cdot L_j^{-u^*_{rev} \cdot r} = L_j^{(u_{rev} - u^*_{rev}) \cdot r} \tag{11}$$

$$C_{1,j,rev^*} = T_j^{u^*_{rev} \cdot r}, \quad C_{2,j,rev^*} = L_j^{u^*_{rev} \cdot r} \tag{12}$$

The original attribute-encrypted ciphertext should be replaced with the newly computed ciphertext $CT^{`}_{ABE}$ on the blockchain.

*6) Data decryption:* The decryption algorithm operates recursively on the access tree $P_{\omega^*}$. For each leaf node $leaf_j$, query the attribute revocation list $R$.

If the user's attribute has not been revoked ($j \in \omega \cap \omega^*, List_j = \emptyset$), compute:

$$\begin{aligned} \text{DecryptNode}(CT_{ABE}, SK_{ID,\omega}, \text{leaf}_j) &= \frac{e(S_{i,j} \cdot v^{\beta}, C_{1,j})}{e(g, C_{2,j})} \\ &= e(g,g)^{\alpha r \lambda_i \cdot q_{\text{leaf}_j}(0)} \end{aligned} \tag{13}$$

If the user's attribute has been revoked ($j \in \omega \cap \omega^*, List_j \neq \emptyset$), compute:

$$\begin{aligned} &\text{DecryptNode}(CT_{ABE}, SK_{ID,\omega}, \text{leaf}_j) \\ &= \prod_{rev \in \text{List}_j} \left( \frac{e(S_{i,j} \cdot v^{\beta}, C_{1,j,rev})}{e(g, C_{2,j,rev})} \right)^{\frac{\lambda_i}{\lambda_i - \lambda_{rev}}} \\ &= e(g,g)^{\alpha r \lambda_i \cdot q_{\text{leaf}_j}(0)} \end{aligned} \tag{14}$$

For a non-leaf node $x$ in the access tree $P_{\omega^*}$ where the child node is $z$, call the result of $DecryptNode(CT_{ABE}, SK_{ID,\omega}, z)$ and denote it as $F_Z$.

$$\begin{aligned} F(x) &= \prod_{z \in S_x} F(z)^{\Delta_{i,S_x^{(0)}}} \\ &= \prod_{z \in S_x} \left(e(g,g)^{\alpha r \lambda_i \cdot q_z(0)}\right)^{\Delta_{i,S_x^{(0)}}} \\ &= e(g,g)^{\alpha r \lambda_i \cdot q_x(0)} \end{aligned} \tag{15}$$

The user is able to recursively compute $F(root) = e(g,g)^{\alpha r \lambda_i s}$, using the Lagrange interpolation method if and only if their attribute set $\omega$ satisfies the access policy $P_{\omega^*}$.

The plaintext $m_1$ is ultimately recovered through the subsequent calculation:

$$\begin{aligned} m_1 &= \frac{C_0 \cdot e(C^*, W_i^*)}{F(root)} \\ &= \frac{m_1 \cdot Z^{sr} \cdot e(g^{sr}, g^{\alpha \lambda_i - \beta})}{e(g,g)^{\alpha \lambda_i rs}} \\ &= \frac{m_1 \cdot e(g,g)^{\beta sr} \cdot e(g^{sr}, g^{\alpha \lambda_i - \beta})}{e(g,g)^{\alpha \lambda_i rs}} \end{aligned} \tag{16}$$

The data user decomposes the retrieved data $m_1 = (URL \| k \| H_1)$ into three distinct components: the unique address $URL$, the symmetric key $k$, and the hash value $H_1$. The user will locally store $k$ and $H_1$, and use the $URL$ to query IPFS and download the encrypted data $CT_m$.

The data user performs the decryption algorithm $Decrypt(CT_m, k) \rightarrow m$, where $CT_m$ is the ciphertext downloaded from IPFS, $k$ is the symmetric key provided by the data owner, and $m$ is the original VPP data that the user intends to access. After obtaining $m$, the user performs a hash operation on it, resulting in $H_2$. By comparing $H_2$ with $H_1$, if the two are found to be consistent, the data can be confirmed as authentic and valid. The data can then be utilized for various VPP purposes, such as a grid operator making dispatch decisions, a market participant submitting bids, or a regulator auditing VPP operations.

## IV. Implementation and Performance Evaluation

We conducted the experiments on a server equipped with an 11th-generation 64-bit Intel i7-11800H processor (2.30 GHz, 8 GB RAM). The Hyperledger Fabric blockchain network was implemented on Ubuntu 22.04 using Fabric version 1.4.12 and Fabric CA version 1.4.12. All nodes operated on virtual machines via VMware Workstation. We selected a Java-based pairing cryptography library version 2.0.0 with an A-type curve for bilinear pairing operations relevant to VPP security requirements.

### A. Data Upload and Download Performance

We evaluated data sizes typical for VPP applications: 1 MB (configuration files), 5 MB (hourly operational reports), 10 MB (daily aggregates), 50 MB (market bidding data), 100 MB (high-frequency sensor logs), 500 MB (historical performance data), and 1 GB (comprehensive datasets). The results, shown in Fig. 4, demonstrate linear scaling suitable for VPP data volumes.

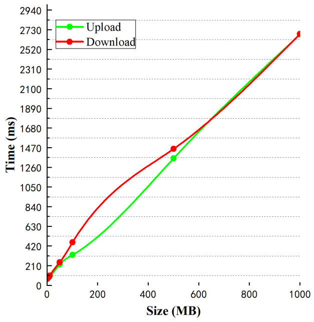
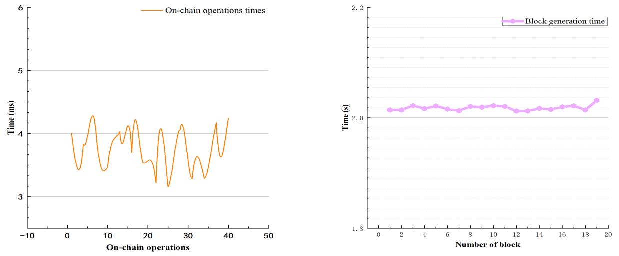


Fig. 4. Data upload and download performance for VPP data on IPFS.

### B. Blockchain Performance for VPP Operations

We measured blockchain operations relevant to VPP scenarios: participant registration (2.8 ms), data access requests (3.1 ms), policy updates (3.5 ms), and revocation operations (4.0 ms). As depicted in Fig. 5, the block generation time remained stable at 2.02 seconds for VPP transaction batches, highlighting the system's operational efficiency. Each block contains VPP-specific metadata: operation type, stakeholder identifiers, timestamp, previous/current hashes, and data hashes.



(a) VPP on-chain operation times. (b) VPP block generation times.

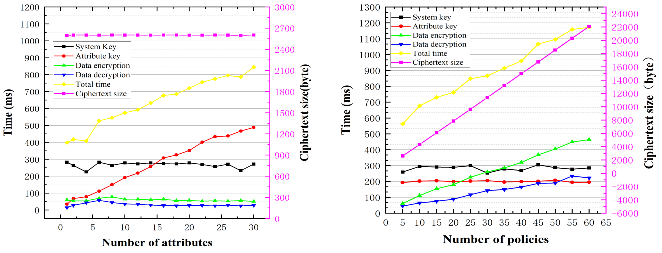
Fig. 5. Blockchain performance for VPP operations.

### C. Access Control Performance Analysis

We evaluated system performance under VPP-specific attribute configurations: stakeholder roles (owner, operator, regulator, consumer), access levels (real-time, historical, financial), and organizational affiliations. With 5 policies and 10 attributes fixed, we measured initialization time (stabilizing at 255 ms), encryption time (proportional to policy complexity), decryption time (meeting real-time VPP requirements), and ciphertext size (scaling linearly with policies). The performance impacts of increasing attributes and policies are presented in Fig. 6. These results confirm the system's suitability for dynamic VPP environments where policies frequently evolve.

### D. Discussion on Practical Deployment

While the experimental results demonstrate the feasibility of SIFChain, several practical considerations warrant discussion beyond the performance analysis presented above.



(a) VPP attribute scaling performance. (b) VPP policy scaling performance.

Fig. 6. SIFChain performance under VPP parameter variations.

First, the experiments were conducted on a server-grade processor, whereas real VPP deployments involve numerous low-power IoT devices at the edge. The computational overhead of CP-ABE operations, particularly pairing and exponentiation, may be significant on such constrained hardware. However, in practice, encryption is performed by data owners (typically VPP operators with sufficient computational resources), while decryption is carried out by consumers such as grid operators and regulators, who also possess adequate capacity. For resource-constrained edge nodes, lightweight ABE variants or offloading cryptographic operations to gateway devices could be considered as mitigations.

Second, our implementation uses Hyperledger Fabric v1.4.12, which, although stable, has been superseded by newer versions (2.x and 3.x) with different performance characteristics. The framework design is version-agnostic, and we expect similar trends in newer releases, but actual latency and throughput may vary. Future work should evaluate SIFChain on updated Fabric versions to validate its performance in modern production environments.

Third, the revocation mechanism requires communication between the data owner and attribute authorities to update the ciphertext. In a wide-area network, this may introduce additional latency proportional to the number of revoked attributes. Our experiments measured revocation operation time at 4.0ms in a local network; however, in geographically distributed settings, network delays could increase this to tens or hundreds of milliseconds. Nevertheless, revocation is an infrequent operation in VPPs, and the impact on overall performance is acceptable given the security benefits.

Finally, the evaluation was conducted in a controlled environment without a geographically distributed VPP consortium. Realistic deployment would involve multiple sites with heterogeneous network conditions and IoT devices, which could affect both performance and security. We plan to conduct field trials in a pilot VPP testbed as future work to validate the framework under real-world conditions.

### E. Security Analysis

The security of SIFChain relies on standard cryptographic assumptions, including the bilinear Diffie-Hellman (BDH) assumption for the CP-ABE scheme and the collision resistance of SHA-256. We provide an informal security analysis covering the following guarantees:

*1) Data confidentiality:* Through the combination of symmetric encryption for IPFS-stored data and CP-ABE for access control, sensitive VPP data remains confidential even if IPFS nodes or blockchain channels are compromised.

*2) Forward and backward security:* The revocable CP-ABE scheme ensures that when a VPP participant's access privileges are revoked, they cannot decrypt newly encrypted data (forward security) nor previously encrypted data that they could previously access (backward security). Even if a user caches old symmetric keys, the re-encryption of the ciphertext with fresh randomness invalidates those keys, preserving backward secrecy.

*3) Collusion resistance:* Even if multiple VPP participants collude by combining their attribute keys, they cannot decrypt ciphertext unless at least one of them individually satisfies the access policy.

*4) Decentralized trust:* The collaborative key generation between CA and attribute authorization nodes eliminates single points of trust, making the system resilient to compromise of any single authority.

*5) Data integrity:* The use of cryptographic hashes ensures that any tampering with VPP data stored on IPFS can be detected during the decryption and verification process.

*6) Privacy leakage:* Although the dual-channel architecture isolates sensitive attributes, blockchain transactions themselves may leak metadata such as access patterns. An adversary monitoring the network could infer relationships between data owners and consumers. To mitigate this, future iterations could incorporate obfuscation techniques or zero-knowledge proofs [20]; however, this remains an open challenge.

We emphasize that a full formal proof of the CP-ABE scheme under bilinear assumptions is beyond the scope of this study and is left as future work. Nevertheless, the design follows established practices from prior art [8], [9].

## V. Conclusion

This study presents SIFChain, a decentralized data management framework for Virtual Power Plants that integrates Fabric blockchain, IPFS, and enhanced revocable CP-ABE to ensure secure storage and dynamic access control. The key innovation is the collaborative key generation mechanism between CA and attribute nodes, reducing centralization vulnerabilities in distributed energy systems. The dual-channel architecture preserves confidentiality while supporting multi-stakeholder VPP operations. The revocable CP-ABE scheme enables fine-grained access control adapting to dynamic VPP environments. By storing encrypted data on IPFS, we achieve scalable security without semi-trusted third parties. Experimental results demonstrate linear scalability and low-latency operations suitable for VPP workloads. We also discussed practical considerations for IoT hardware, newer Fabric versions, and network overhead, and identified privacy leakage as an area for future enhancement. Future work will focus on decentralized attribute encryption, integration with real VPP systems, enhanced key management for large-scale energy applications, and formal security proofs.

## References

[1] P. D. Lund, J. Lindgren, J. Mikkola, and J. Salpakari, "Review of energy system flexibility measures to enable high levels of variable renewable electricity," Renewable and Sustainable Energy Reviews, vol. 45, pp. 785–807, 2015.

[2] E. Mengelkamp, J. Gärtner, K. Rock, et al., "Designing microgrid energy markets: A case study: The Brooklyn Microgrid," Applied Energy, vol. 210, pp. 870–880, 2018.

[3] M. Andoni, V. Robu, D. Flynn, et al., "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," Renewable and Sustainable Energy Reviews, vol.100,pp.143–174,2019.

[4] C. Zhang, J. Wu, C. Long, and M. Cheng, "Review of existing peer-to-peer energy trading projects," Energy Procedia, vol. 105, pp. 2563–2568, 2017.

[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[6] E. Androulaki, A. Barger, V. Bortnikov, et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in Proc. 13th EuroSys Conf., 2018, pp. 1–15.

[7] J. Benet, "IPFS - content addressed, versioned, P2P file system," arXiv preprint arXiv:1407.3561, 2014.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in 2007 IEEE Symp. on Security and Privacy (SP'07), 2007, pp. 321–334.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. on Comput. and Commun. Security, 2006, pp. 89–98.

[10] S. Wang, A. F. Taha, J. Wang, et al., "Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids," IEEE Trans. Syst., Man, Cybern., Syst., vol. 49, no. 8, pp. 1612–1623, 2019.

[11] Z. Li, J. Kang, R. Yu, et al., "Consortium blockchain for secure energy trading in industrial internet of things," IEEE Trans. Ind. Informat., vol. 14, no. 8, pp. 3690–3700, 2018.

[12] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in 2017 IEEE Int. Conf. on Pervasive Comput. and Commun. Workshops (PerCom Workshops), 2017, pp. 618–623.

[13] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the internet of things," Peer-to-Peer Netw. and Appl., vol. 10, no. 4, pp. 983–994, 2017.

[14] X. Luo, B. Zhang, and Z. Zhang, "A secure and efficient data sharing scheme based on blockchain in vehicular social networks," IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 5826–5835, 2020.

[15] M. Al-Shatri, A. Al-Fuqaha, and H. S. Al-Raweshidy, "Virtual power plants: Challenges, opportunities, and profitability assessment in current energy markets," Electricity, vol. 5, no. 2, pp. 370–384, 2024.

[16] A. Bedi, J. Ramprabhakar, R. Anand, V. P. Meena, and I. A. Hameed, "Empowering net zero energy grids: A comprehensive review of virtual power plants, challenges, applications, and blockchain integration," Discover Applied Sciences, vol. 7, Art. no. 252, 2025.

[17] C. Feng, J. Cheng, and S. Chen, "Blockchain-based transaction mechanism in virtual power plant: Considering users' privacy and reputation," Electric Power Systems Research, vol. 249, Art. no. 111988, 2025.

[18] X. Wang, L. Zhang, and J. Wu, "Hyperledger Fabric-based multi-channel structure for data exchange in internet of vehicles," Electronics, vol. 14, no. 3, Art. no. 572, 2025.

[19] J. Xu, W. Susilo, and J. K. Liu, "An efficient traceable and revocable access control scheme for smart grids," Symmetry, vol. 17, no. 2, Art. no. 294, 2025.

[20] Q. Li, H. Wang, and B. Cao, "Advancing user privacy in virtual power plants: A novel zero-knowledge proof-based distributed attribute encryption approach," Electronics, vol. 13, no. 7, Art. no. 1283, 2024.

[21] W. Luan, L. Tian, B. Zhao, and Q. Ai, "A multi-timescale blockchain-based virtual power plant trading framework for building integrated photovoltaic prosumers," Applied Energy, vol. 398, Art. no. 126422, 2025.

[22] M. Song, X. Xu, C. Gao, et al., "Two-stage stochastic scheduling of virtual power plant based on transactive control," CSEE Journal of Power and Energy Systems, vol. 11, no. 3, pp. 1442–1453, 2025.