

Article

LOR-A2ABE: Lightweight and Revocable Attribute-Anonymous ABE with Outsourced Decryption in Centralized IoT

Dan Gao ^{1,*}, Huanhuan Xu ^{1,2} and Shuqu Qian ³¹ Department of Information Engineering, Anshun Technical College, Anshun 561000, China² College of Computer Science and Technology, Guizhou University, Guiyang 550025, China; huanh_xu@163.com³ School of Mathematics and Computer Science, Anshun University, Anshun 561000, China; shuquqian@163.com

* Correspondence: dan_gao_2025@163.com; Tel.: +86-159-1450-1375

Abstract

Due to the rapid proliferation and evolution of the Internet of Things (IoT) in industrial and smart city applications, concerns over sensitive data security have become increasingly prominent. This is especially true in resource-constrained “cloud–terminal” centralized architectures, where ensuring privacy protection for downlink data and implementing fine-grained access control have become critical. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) serves as an effective solution due to its fine-grained access control capability. Nevertheless, conventional CP-ABE approaches face notable limitations when deployed in these practical settings, including the lack of an efficient and lightweight client-side revocation mechanism, excessive decryption overhead on terminal devices, and the practical difficulty in balancing security with performance. To address these issues, this paper proposes LOR-A2ABE, a Lightweight, Outsourced, and Revocable Anonymous Attribute-Based Encryption scheme. The scheme achieves lightweight client-side revocation through partial updates by embedding version numbers and timestamps into keys and ciphertexts via hash mapping. Furthermore, it employs outsourcing to offload the majority of computations to the cloud, allowing client-side decryption with only constant, low-complexity operations, thereby significantly reducing the computational burden on resource-constrained terminals. Considering the practical context where client devices are typically resource-limited sensors or microcontrollers and downlink data often require real-time processing, our scheme adopts a practical security model optimized for IoT constraints. This model prioritizes forward security and efficient revocation—the most critical requirements for operational IoT systems—while maintaining provable security under the Decisional Linear (DLIN) assumption within a bounded collusion model, achieving IND-CPA security and anonymity. Theoretical analysis and experimental simulations show that LOR-A2ABE incurs acceptable and controllable overhead in the key issuance and encryption phases, while outperforming most existing schemes in decryption and revocation efficiency, making it particularly suitable for “cloud–terminal” centralized IoT environments where terminal devices are resource-constrained and require frequent decryption operations.

Keywords: “cloud–terminal” centralized architecture; CP-ABE; lightweight revocation; outsourced decryption; anonymity



Academic Editor: Xiaogang Qi

Received: 10 January 2026

Revised: 4 February 2026

Accepted: 5 February 2026

Published: 6 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The Internet of Things (IoT), built upon robust networks of real-time sensors and microcontrollers, is widely applied in domains such as smart cities [1,2] and the Industrial IoT [3,4]. Currently, key constraints on IoT development include the limited computational resources of terminal devices [3,5,6] and the security and privacy concerns associated with the massive amounts of sensitive data generated by IoT devices [7,8]. The issue of constrained computational resources is often addressed or mitigated by integrating with cloud computing [9–11], while the security and privacy of sensitive data are tackled through cryptographic techniques [12–16].

Moreover, compared to distributed networks, centralized networks offer advantages such as lower initial deployment and maintenance costs, more straightforward and unified overall management, configuration, and policy implementation [17], greater convenience in data aggregation and large-scale analysis [18] and enhanced performance in resource scheduling and load balancing [19]. These advantages are especially decisive in IoT applications where security, real-time response, and consistent policy enforcement are paramount.

These practical advantages of centralized architectures become particularly critical in security-sensitive IoT applications demanding real-time, fine-grained access control. Consider a smart factory predictive maintenance system: vibration and temperature data from hundreds of machines are streamed to a central analytics platform. The access policy must be dynamic, granting engineers from specific maintenance teams access only to data from their assigned production lines and only during their scheduled shifts. In such a scenario, a distributed access control system would face inherent challenges: (1) Policy Synchronization Latency: Propagating and reaching consensus on immediate attribute revocation (e.g., when an engineer's shift ends) across all devices would introduce unacceptable delay, creating security windows. (2) Communication Overhead: Frequent, fine-grained policy updates would generate massive coordination traffic among resource-constrained sensors, contradicting the goal of energy efficiency. In contrast, a centralized authority, as assumed in our work, can enforce such complex, dynamic policies instantly and consistently, making it the natural architectural fit for these “cloud–terminal” IoT environments where data aggregation and real-time control are paramount.

This example underscores the critical need for fine-grained access control within the “cloud–terminal” centralized IoT paradigm. Ensuring that downlink commands or data are accessible only to authorized users or devices is critical, making fine-grained access control a core security requirement [20–22]. Attribute-Based Encryption (ABE) can be classified into Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE), distinguished by CP-ABE binding the access policy to the ciphertext, whereas KP-ABE binds it to the key [23]. Comparatively, CP-ABE is evidently more suitable for the “cloud–terminal” centralized IoT environment characterized by a vast user scale, massive shared data, and dynamically changing attributes [24].

However, CP-ABE relies on bilinear pairing computations, and its attribute matching efficiency decreases as the number of users and the complexity of the access policy increase [23,25,26]. Classical CP-ABE frameworks do not adequately address the efficiency challenges of large-scale matching, such as the BSW framework [23], CGW framework [27], ABGW framework [28], and FAME framework [29]. In 2022, Riepel et al. proposed the FABEO framework [30], which utilizes cluster operations to concentrate substantial computations into the smallest group, achieving the fastest performance among classical CP-ABE frameworks. Subsequently, the FABESA scheme [31], based on the FABEO framework, introduced an adaptive chosen-plaintext attack (CPA) secure scheme with anonymity. In the “cloud–terminal” centralized IoT environment, attribute name hiding can better mitigate the privacy and security risks introduced by “centralization”. However, FABESA

is not specifically designed for IoT environments and thus cannot be well-adapted to the “cloud–terminal” centralized IoT setting.

Simultaneously, considering the resource-constrained nature of IoT environments, cloud computing outsourcing has emerged as a solution to alleviate the computational overhead of bilinear pairings in CP-ABE [32–35]. However, the scheme proposed in [32] has a weaker security model, offering only static security. The scheme in [33] achieves stronger CCA security but imposes a significant computational burden on the server and lacks in-depth investigation into dynamic attribute management in IoT contexts. The FOC-PH-CP-ABE scheme [34] significantly enhances privacy protection through policy hiding and a fully outsourced mechanism, but this security improvement comes at the cost of partially reduced encryption/decryption efficiency, resulting in overall performance inferior to schemes like [30] that prioritize computational speed.

These schemes do not sufficiently address dynamic attribute management, particularly attribute revocation. Yet, in “cloud–terminal” centralized IoT environments, large-scale dynamic management of terminal attributes is inevitable. Some schemes have proposed attribute revocation methods, such as the MTA scheme [36] and the TRH scheme [37], but their revocation efficiency is limited, thus are not suitable for achieving lightweight revocation in IoT environments.

However, these existing methods are not well-suited for the “cloud–terminal” centralized IoT environment:

1. **Resource Constraints and Environmental Adaptability Issues:** Schemes such as FABESA [31] achieve strong theoretical security but are not optimized for the resource-constrained nature of IoT environments. Moreover, they lack efficient attribute revocation mechanisms essential for dynamic IoT deployments.
2. **Insufficient Revocation Efficiency:** Although revocation schemes like MTA [36] and TRH [37] exist, they introduce complexities of $O(|A|)$ or $O(\log |U|)$ that scale with system size. For large-scale IoT deployments with frequent attribute updates, such overhead becomes prohibitively high and contradicts the lightweight requirements of IoT terminals.
3. **Imbalanced Computation Distribution:** Some IoT-oriented schemes, such as FOC-PH-CP-ABE [34], achieve computational outsourcing at the cost of significantly increased encryption time or server burden. This creates new bottlenecks in the “cloud–terminal” processing pipeline and shifts rather than resolves the computational challenges.

In summary, the current landscape lacks a CP-ABE scheme that simultaneously provides all of the following characteristics: (1) an $O(1)$ lightweight revocation mechanism suitable for dynamic IoT management; (2) truly efficient outsourced decryption with constant-time client-side operations; (3) attribute anonymity for enhanced privacy in centralized architectures; and (4) practical security trade-offs that align with IoT operational realities, such as prioritizing forward security and accepting limited backward secrecy based on data timeliness.

To address these limitations, this paper proposes a novel attribute-based encryption scheme specifically designed for centralized IoT environments. Our main contributions are as follows:

1. The proposed LOR-A2ABE scheme achieves lightweight attribute revocation by partially updating keys and ciphertexts, providing an efficient attribute revocation management solution for large-scale IoT environments.
2. A prominent advantage of our scheme lies in its highly efficient and practical decryption performance on the user side. We employ outsourcing techniques to offload the majority of computations to the cloud, requiring IoT terminal devices to perform only

constant, low-complexity operations for decryption, thereby significantly alleviating the computational burden on end-users.

3. In terms of security, our scheme incorporates a deliberate trade-off in its security design. We forgo backward security and operate under a limited collusion resistance assumption. Concurrently, we formally prove that the scheme achieves forward security, IND-CPA security, and anonymity security under the DLIN assumption.

A comprehensive performance evaluation combining theoretical analysis and experimental simulations demonstrates that: by accepting controllable overhead in key issuance and encryption, our scheme achieves lightweight attribute revocation while ensuring its decryption performance is significantly superior to that of most mainstream schemes. This highlights its exceptional practical value, making it particularly suitable for scenarios in “cloud–terminal” IoT where a massive number of resource-constrained terminal devices frequently perform decryption operations.

The structure of this paper is outlined as follows. In Section 2, we introduce essential cryptographic background information and formally establish the security model. Section 3 elaborates on the construction of our LOR-A2ABE scheme. Section 4 furnishes a formal security proof of the proposed scheme. A performance evaluation is conducted in Section 5, encompassing theoretical analysis and experimental comparisons with prior works. Lastly, Section 6 summarizes the paper and suggests avenues for future investigation.

2. Preliminaries

2.1. Bilinear Pairings

Suppose λ denotes the security parameter and p is a large prime whose bit-length equals λ , which ensures the system withstands attacks of computational cost $O(2^\lambda)$, thereby delivering λ -bit security. Consider G_1, G_2 as additive cyclic groups of prime order p , and G_T as a multiplicative cyclic group also of order p . Assume $g_1 \in G_1$ and $g_2 \in G_2$ are generators of their respective groups. A bilinear map $e : G_1 \times G_2 \rightarrow G_T$ is called a Type 3 (asymmetric) pairing if it is non-degenerate, no efficient homomorphism exists between G_1 and G_2 , and it satisfies the following properties:

1. Bilinearity: For any $a, b \in \mathbb{Z}_p$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
2. Non-degeneracy: $e(g_1, g_2)$ is a generator of G_T . For all $a, b \in \mathbb{Z}_p$, $e(g_1^a, g_2^b) = 1_{G_T}$ if and only if $a = 0$ or $b = 0$.
3. Computability: There exists an efficient algorithm such that $e : G_1 \times G_2 \rightarrow G_T$ is computable in deterministic polynomial time with respect to the security parameter λ .

2.2. Monotone Span Programs

A monotone span program (MSP) is defined by the quadruple $(\mathbb{F}, M, \rho, \mathbf{v})$, where:

1. $\mathbb{F} = \mathbb{Z}_p$ is a finite base field of prime order p .
2. $M \in \mathbb{Z}_p^{m \times n}$ is an $m \times n$ matrix over $\mathbb{F}$.
3. $\rho : \{1, 2, \dots, m\} \rightarrow U$ is a labeling function that maps each row of M to an attribute in the universal attribute set U . Multiple rows may be mapped to the same attribute.
4. $\mathbf{v} = (1, 0, 0, \dots, 0) \in \mathbb{Z}_p^{1 \times n}$ is the target vector.

For any attribute set $S \subseteq U$, define the row index set $I_S = \{i \in \{1, \dots, m\} \mid \rho(i) \in S\}$. The set S is considered authorized (i.e., satisfying the access policy) if and only if the target vector $\mathbf{v}$ lies in the row space spanned by $\{M_i\}_{i \in I_S}$.

Formally, there must exist scalars $\{w_i \in \mathbb{F}\}_{i \in I_S}$ such that:

$$\sum_{i \in I_S} w_i \cdot M_i = \mathbf{v}$$

2.3. Partially Hidden Structure (Anonymity)

To mitigate privacy inference threats in centralized IoT management systems using a “cloud–terminal” model, it is beneficial to conceal sensitive attribute values while revealing only non-sensitive attribute names. This approach not only enhances decryption efficiency but also prevents leakage of sensitive information. The concept of “partially hidden structure” was introduced by Lai et al. [38] to address this challenge.

Let $S = \{u_i\}_{i \in [m]}$ be an attribute set, where each $u_i = (n_i, v_i)$ includes an attribute name n_i and its corresponding value v_i . An access policy A can be expressed as $A = (M, \pi, \{\pi(i)\}_{i \in [l]})$, where M is an $l \times n$ matrix and the mapping function π maps the i -th row of M to an attribute $\pi(i) = (n_{\pi(i)}, v_{\pi(i)})$. The attribute set S satisfies the access policy A if and only if there exists a collection of constants $\{\gamma_i\}_{i \in I}$ satisfying

$$\sum_{i \in I} \gamma_i M_i = (1, 0, \dots, 0),$$

and $\pi(i) = u_i$ holds for all $i \in I$.

It is important to delineate the privacy guarantee of this “partially hidden” approach. The scheme protects the confidentiality of specific attribute values (e.g., whether “Age” = 25 or 40) but does not hide the attribute names themselves (e.g., “Age”, “Department”, “Medical_Condition”). In scenarios where attribute names carry sensitive metadata (e.g., an access policy involving “Has_Heart_Disease” directly reveals the medical context of the data), this constitutes a semantic leakage. Our security model, following the standard notion of attribute anonymity [38], ensures that given a ciphertext, an adversary cannot determine which among a set of possible value assignments was used, assuming the attribute names are known or non-critical.

Therefore, the applicability of our anonymity feature depends on the deployment context. In many IoT scenarios (e.g., access control based on “Sensor_Type”, “Location”, or “Temperature_Range”), attribute names are non-sensitive, making value-hiding both sufficient and efficient. For environments where even attribute names are confidential, fully policy-hiding techniques would be required, which typically incur significantly higher computational and communication overhead, conflicting with the lightweight objectives of this work.

2.4. Decisional Linear (DLIN) Assumption

Let $\text{par} = (p, G_1, G_2, G_T, g_1, g_2, e : G_1 \times G_2 \rightarrow G_T)$ be the description of a standard asymmetric (Type 3) bilinear-map setting with security parameter 1^λ .

Pick random $x_1, x_2, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_p$ and a random $R \xleftarrow{\$} \mathbb{Z}_p$. The public information is given by

$$D = (g_1^{x_1}, g_1^{x_2}, g_2^{y_1}, g_2^{y_2}, g_1^{x_1 y_1}, g_1^{x_2 y_1}, g_2^{x_1 y_1}, g_2^{x_2 y_1}).$$

An adversary $\mathcal{A}$ is challenged to distinguish between the two tuples:

$$T_0 = (g_1^{y_2 + y_1}, g_2^{y_2 + y_1}), \quad T_1 = (g_1^R, g_2^R).$$

The advantage of $\mathcal{A}$ in solving the DLIN problem within polynomial time is defined as

$$\text{Adv}_{\mathcal{A}}^{\text{DLIN}}(\lambda) = |\Pr[\mathcal{A}(\text{par}, D, T_0) = 1] - \Pr[\mathcal{A}(\text{par}, D, T_1) = 1]|.$$

We say that the DLIN assumption holds under the group parameters par if the advantage $\text{Adv}_{\mathcal{A}}^{\text{DLIN}}(\lambda)$ is negligible for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$.

3. LOR-A2ABE

3.1. System Model

The architecture of the LOR-A2ABE scheme is illustrated in Figure 1. It consists of the following entities: the Central Authority (CA), the IoT Device User (DU), the Cloud Server (CS), and the data acquisition system (Sensor). The scheme mainly involves five processes as described below.

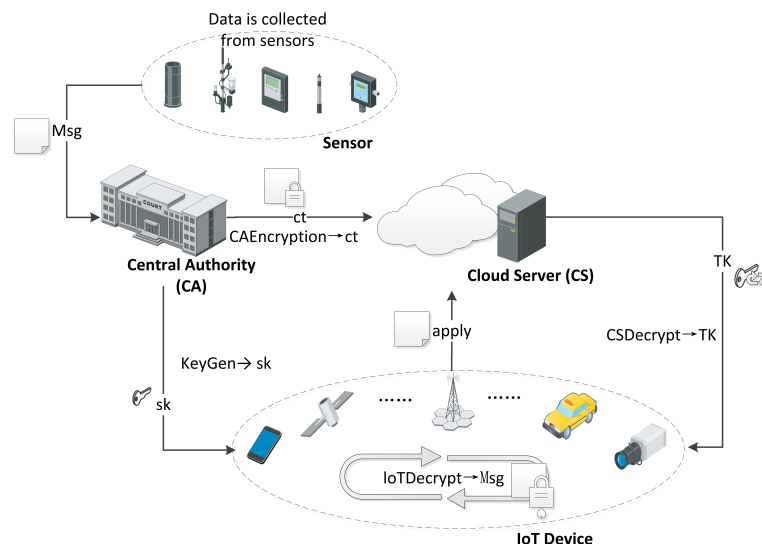


Figure 1. System framework diagram.

1. System Initialization: The CA based on the security parameter λ , and runs the system initialization algorithm to produce the master key pair.
2. Key Generation: The DU submits a registration request to the CA, and the CA generates attribute keys for the DU according to its attribute set.
3. Data Encryption: The CA encrypts the data Msg submitted by the Sensor under a access control structure, and uploads the resulting to the CS for storage, enabling the DU to apply for download on demand.
4. Data Decryption: The DU applies to the CS for ciphertext download and submits part of its own attribute keys. Based on the request, the CS performs partial decryption on the ciphertext, generates a transformation key, and returns it to the DU. The DU then uses the transformation key to complete the final decryption and obtains the data Msg .
5. Revocation: The CA periodically generates key update information and updates the ciphertext stored in the CS. The key update information is sent to non-revoked DUs, who then update their own keys locally upon receipt.

In LOR-A2ABE, encryption and other complex operations are handled by the resource-sufficient CA, not by constrained terminals (DU). Terminal devices only perform lightweight decryption and key updates with constant complexity, ensuring the scheme's "lightweight" characteristic genuinely benefits the resource-constrained endpoint.

3.2. Detailed Construction

This section provides a detailed description of the proposed LOR-A2ABE scheme.

1. System Initialization. The CA performs system initialization according to the security parameter λ and generates the system master key pair via $\text{Setup}(1^\lambda) \rightarrow (msk, mpk)$. The CA randomly selects $\alpha, b_1, b_2 \xleftarrow{\$} \mathbb{Z}_p$. Then, the master secret key and the master public key are defined as

$$msk = (\alpha, \beta, b_1, b_2), \quad (1)$$

$$mpk = (\text{par}, g_1^{b_1}, g_1^{b_2}, g_2^\beta, H, e(g_1, g_2)^\alpha). \quad (2)$$

2. **Key Generation.** Upon receiving a registration request from a DU, the CA generates attribute keys according to the DU's attribute set $S \subseteq \mathbb{U}$ by running $\text{KeyGen}(mpk, msk, S) \rightarrow sk$.

Randomly choose $r \xleftarrow{\$} \mathbb{Z}_p$, a version number $Ver \xleftarrow{\$} \mathbb{Z}_p$, and a timestamp *nonce*. Compute:

$$sk_1 = g_1^r, \quad (3)$$

$$sk_2 = g_1^\alpha \cdot g_1^{\beta r}, \quad (4)$$

$$sk_{3,u} = g_1^{H(0\|u_i)r/b_1} \cdot g_1^{r/b_1} = g_1^{(H(0\|u_i)+1)r/b_1}, \quad (5)$$

$$sk_{4,u} = g_1^{H(1\|u_i)r/b_2} \cdot g_1^{r/b_2}, \quad (6)$$

$$sk_5 = e(g_1^\alpha, g_2)^{H(\text{nonce}\|Ver)}. \quad (7)$$

For each attribute $u \in S$, the full secret key is

$$sk = (sk_1, sk_2, \{sk_{3,u}, sk_{4,u}\}_{u \in S}, sk_5). \quad (8)$$

3. **Encryption.** The CA encrypts the received message *Msg* under an access structure (M, π) and uploads the resulting ciphertext to the CS via $\text{CAEncryption}(\text{Enc}(\text{Msg}), mpk, (M, \pi)) \rightarrow ct$.

Choose random $s_1, s_2 \xleftarrow{\$} \mathbb{Z}_p$ and set $s = s_1 + s_2$. Additionally, choose a random vector $\mathbf{v} = (s, v_2, \dots, v_n)^\top \in \mathbb{Z}_p^n$ where $v_2, \dots, v_n \xleftarrow{\$} \mathbb{Z}_p$. For each row i of the matrix M , compute the share $\lambda_i = M_i \cdot \mathbf{v}$. Then compute:

$$ct_0 = \text{Msg} \cdot e(g_1^\alpha, g_2)^s, \quad (9)$$

$$ct_{1,i} = g_2^{M_i(s\|v)^\top + s + s_1 H(0\|\pi(i)) + s_2 H(1\|\pi(i))}, \quad (10)$$

$$ct_2 = g_2^{s + H(\text{nonce}\|Ver)}, \quad (11)$$

$$ct_3 = g_2^{b_1 s_1}, \quad (12)$$

$$ct_4 = g_2^{b_2 s_2}, \quad (13)$$

$$ct_5 = g_2^{\beta(s + H(\text{nonce}\|Ver))} \cdot g_2^s. \quad (14)$$

Here, $M_i(s\|v)^\top$ equals the share λ_i computed for row i of the monotone span program M . The terms s , $s_1 H(0\|\pi(i))$, and $s_2 H(1\|\pi(i))$ serve as attribute-specific blinding factors essential for security and anonymity. The complete ciphertext is

$$ct = (ct_0, \{ct_{1,i}\}_{i \in [n_1]}, ct_2, ct_3, ct_4, ct_5). \quad (15)$$

4. **Outsourced Decryption (CS-Side).** After receiving a decryption request from a DU, the CS performs partial decryption on the ciphertext and obtains a transformation key by executing $\text{CSDecrypt}(S, ct, sk) \rightarrow TK$.

If the terminal's attribute set S is an authorized set under the access structure (M, π) , there exists a set of constants $\{\gamma_i\}_{i \in I}$ such that $\sum_{i \in I} \gamma_i M_i = (1, 0, \dots, 0)$. The CS then computes the transformation key as:

$$TK = \frac{e(sk_2, ct_2) \cdot e\left(\prod_{i \in I} (ct_{1,\pi(i)})^{\gamma_i}, sk_1\right)}{e(sk_1, ct_5) \cdot e\left(\prod_{i \in I} (sk_{3,i})^{\gamma_i}, ct_3\right) \cdot e\left(\prod_{i \in I} (sk_{4,i})^{\gamma_i}, ct_4\right)}. \quad (16)$$

5. Final Decryption (DU-Side). After obtaining the transformation key returned by the CS, the DU can complete the final decryption and recover the original message via $\text{loTDecrypt}(TK, sk_5) \rightarrow \text{Msg}$.

The DU first recovers the session key:

$$key = TK/sk_5. \quad (17)$$

The original message is then obtained as:

$$\text{Msg} = ct_0/key. \quad (18)$$

6. Lightweight Attribute Revocation. The CA periodically generates key update information and refreshes the stored ciphertext by running $\text{Revoked}(sk, ct) \rightarrow (sk', ct')$. The CA randomly selects a new version number $Ver' \xleftarrow{\$} \mathbb{Z}_p$ and updates the timestamp to $nonce'$. It then computes:

$$update = H(nonce' || Ver'), \quad (19)$$

$$key_update = e(g_1, g_2)^{\alpha \cdot update}. \quad (20)$$

The key_update is sent to all non-revoked DUs. Upon receiving it, each DU locally updates its secret-key component:

$$sk'_5 = sk_5 \cdot key_update. \quad (21)$$

In LOR-A2ABE, the revocation mechanism is designed with built-in constraints to counter practical collusion attacks. First, the key_update value is intrinsically linked to the specific $(nonce', Ver')$ pair generated in each update round. Devices that have been revoked are excluded from subsequent update distributions and thus cannot obtain the necessary $nonce'$ and Ver' values. Without these parameters, even if the key_update value itself is leaked, it is impossible to construct the correct versioned hash chain required for successful decryption. Second, any intercepted key_update value has a limited validity window, typically restricted to the period before the next update. This design significantly limits the practical feasibility of collusion attacks that rely on sharing such critical update information.

Concurrently, the CA updates the ciphertext. It chooses a fresh random $s' \xleftarrow{\$} \mathbb{Z}_p$ and computes:

$$ct_{2_new} = g_2^{s' + update}, \quad (22)$$

$$ct_{5_new} = g_2^{\beta(s' + update)}. \quad (23)$$

The updated ciphertext components are:

$$ct'_0 = ct_0 \cdot e(g_1, g_2)^{\alpha s'}, \quad (24)$$

$$ct'_2 = ct_2 \cdot ct_{2_new}, \quad (25)$$

$$ct'_5 = ct_5 \cdot ct_{5_new}, \quad (26)$$

$$ct' = (ct'_0, \{ct'_{1,i}\}_{i \in [n_1]}, ct'_2, ct_3, ct_4, ct'_5). \quad (27)$$

4. Security Proofs

4.1. Design Trade-Offs for Security

This scheme is specifically designed for the resource-constrained “cloud–terminal” centralized IoT Management Environments and incorporates explicit design trade-offs in security. We prioritize forward secrecy, which is critical to the overall system security, while achieving a lightweight user revocation mechanism. Given that data in such environments usually possesses strong timeliness and the set of devices remains relatively stable, backward secrecy is considered non-essential in this scenario. Furthermore, to address potential collusion risks that may arise in practical deployment, the scheme assumes a limited collusion resistance: by strictly confining key update operations to local execution on legitimate terminal devices, this prevents adversaries from acquiring or updating attribute keys belonging to other devices.

This design choice allows the scheme to effectively guard against common attacks and a certain range of internal collusion threats, while significantly reducing both computational and communication overhead.

4.1.1. Trade-Offs of Limited Collusion Resistance

The practical effectiveness of LOR-A2ABE in IoT deployments is enhanced by considering the operational constraints of real-world environments. In standard cryptographic theory, it is typically assumed that adversaries possess unlimited collusion capabilities. However, in the “cloud–terminal” centralized IoT environment, terminal devices are usually resource-constrained sensors or microcontrollers with severely limited computational power, storage capacity, and communication bandwidth.

Considering this practical context, we identify the following operational characteristics that enhance the scheme’s practical effectiveness in IoT deployments: In the “cloud–terminal” centralized IoT architecture, it is operationally difficult for an adversary to organize terminal devices into an effective, real-time collusion network for key information exchange:

1. Terminal devices typically perform predefined single functions and lack the computational and communication capabilities required to dynamically coordinate and execute complex collusion protocols.
2. The *key_update* information is strongly bound to system version numbers and timestamps, possessing a strong timeliness attribute. Revoked devices struggle to obtain the version hash chain corresponding to the ciphertext. Missing any version hash value will prevent normal decryption.
3. In practical deployments, terminal devices are often physically or logically isolated, making it difficult to establish direct and covert channels for key information exchange.
4. In the target IoT scenarios, protected downlink data (e.g., control commands, configuration updates) usually have strong timeliness requirements. A successful collusion attack not only requires obtaining all historical *key_update* values but must also be completed within the data’s validity period and before the next system update, which further increases the practical difficulty of the attack.

Under these operational conditions, even if an adversary controls some terminal devices, real-time exchange of *key_update* information remains challenging. This practical analysis explains why LOR-A2ABE is well-suited for IoT deployments, while its formal security is guaranteed by the cryptographic proofs in Sections 4.4 and 4.5.

4.1.2. Trade-Offs of Backward Secrecy

The design of LOR-A2ABE intentionally prioritizes forward secrecy and system efficiency over backward secrecy—a deliberate decision grounded in the practical con-

straints of IoT operational environments. This trade-off is justified by the following key considerations:

Time-Sensitive Nature of Data Value. In IoT environments, downstream data—such as control commands, sensor readings, and configuration updates—typically exhibits strong timeliness. Once data exceeds its effective operational period, its security significance diminishes considerably. Consequently, ensuring the security of future data (forward secrecy) is paramount for system operation, while the necessity of protecting historical data (backward secrecy) decreases as data gradually loses operational relevance.

Imperative for System Efficiency. Achieving cryptographically strong backward secrecy would impose substantial system overhead. Conventional approaches either require re-encryption of all historical ciphertexts during revocation (resulting in $O(n)$ communication costs) or necessitate complex key evolution mechanisms that increase storage and computational burdens on resource-constrained devices. Both methods would compromise the $O(1)$ revocation efficiency essential for scalable IoT deployments with frequent attribute updates.

Controlled Risk Exposure. In target applications such as sensor data streaming, environmental monitoring, and real-time control commands, data value decays rapidly over time. The practical risk associated with revoked devices accessing historical data is acceptably low, making the performance gains from efficient revocation mechanisms a worthwhile trade-off against comprehensive backward secrecy.

Coherence with Limited Collusion Resistance. It is important to note that this limitation in backward secrecy does not contradict the limited collusion resistance mechanism described in Section 4.1.1. While limited collusion resistance enhances protection during data's operational period by increasing the difficulty of real-time collusion, the acceptance of limited backward secrecy reflects the diminished need for protection after data exceeds its useful lifespan. Together, these design choices implement a time-proportional security model that provides strong protection when data matters most while maintaining efficient system operation.

Based on these considerations, LOR-A2ABE intentionally omits cryptographic backward secrecy—a clear design choice that enables the lightweight, efficient revocation essential for practical IoT deployments. This is not a compromise but a targeted optimization: we deliberately exchange comprehensive backward security for the performance characteristics required by resource-constrained, time-sensitive IoT environments.

4.2. Security Model

4.2.1. IND-CPA Security

The IND-CPA security is defined via a security game $\text{Game}_{\text{CP-ABE}}^{\text{IND-CPA}}(\mathcal{A}, \lambda)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Phase 1: Initialization. The challenger $\mathcal{C}$ executes the $\text{Setup}(\lambda)$ algorithm, which outputs the master public key mpk and the master secret key msk . Subsequently, $\mathcal{C}$ provides mpk to the adversary $\mathcal{A}$ while keeping msk confidential.

Phase 2: Query Phase 1. The adversary $\mathcal{A}$ is permitted to adaptively submit a polynomially bounded number of queries of the following types to $\mathcal{C}$:

- **Random oracle query $H(\cdot)$:** $\mathcal{C}$ maintains a query list and responds to each distinct query with a randomly generated value.
- **Key-generation oracle $O_{\text{KeyGen}}(S)$:** $\mathcal{A}$ submits an attribute set S . $\mathcal{C}$ runs KeyGen to generate a secret key sk_S , records sk_S in a user list UL , and returns sk_S to $\mathcal{A}$.
- **Revocation oracle $O_{\text{revok}}(uid)$:** $\mathcal{A}$ submits a user uid to be revoked, provided that $uid \in UL$ and $uid \notin RL$ (the revocation list). $\mathcal{C}$ generates a new version number $Ver' \xleftarrow{\$} \mathbb{Z}_p$ and a timestamp $nonce' \xleftarrow{\$} \mathbb{Z}_p$, runs the revoked algorithm to update the

ciphertext to ct_{update} and to produce a key-update message key_update , sends the updated ciphertext and the key-update message to $\mathcal{A}$, and records uid together with key_update in the revocation list.

- Key-update oracle $O_{\text{KeyUpdate}}(uid, key_update)$: $\mathcal{A}$ submits a user uid and a key-update message key_update , provided that $uid \in \text{UL}$ and $uid \notin \text{RL}$. $\mathcal{C}$ updates the user's secret key using key_update and returns the updated key sk_{update} to $\mathcal{A}$.

Remark 1. This step mimics the real-world scenario where each terminal or sensor can only update its own key. Because terminals are not actual users and their resources are extremely constrained, we consider a “limited-collusion” setting, i.e., the adversary $\mathcal{A}$ is not allowed to use update to refresh other users' keys.

- Outsourced-decryption oracle $O_{\text{Transform}}(ct, uid)$: $\mathcal{A}$ submits a ciphertext ct and a user uid . $\mathcal{C}$ runs CSDecrypt to compute the transformation key TK for ct and returns TK to $\mathcal{A}$.

Phase 3: Challenge. $\mathcal{A}$ chooses two messages m_0 and m_1 of identical length, together with a target access structure (M, π) , subject to the condition that none of the attribute sets requested in Phase 2 can satisfy (M, π) . $\mathcal{C}$ then picks a random bit $b \in \{0, 1\}$, invokes CAEncryption with the current version and timestamp to encrypt m_b under (M, π) , and delivers the resulting challenge ciphertext ct^* to $\mathcal{A}$.

Phase 4: Query Phase 2. $\mathcal{A}$ may again issue a polynomial number of queries as in Phase 2, subject to the following restrictions:

- In any $O_{\text{KeyGen}}(S)$ query, S must not satisfy (M^*, π^*) .
- In any $O_{\text{revok}}(uid)$ query, if the user uid possesses an attribute set that satisfies (M^*, π^*) and has already obtained the corresponding key in Phase 2, then uid cannot be revoked in this phase.
- In any $O_{\text{KeyUpdate}}(uid, update)$ query, if uid has been revoked, the update information $update$ cannot be used to refresh its key.
- The $O_{\text{Transform}}(\cdot)$ oracle cannot be queried on the challenge ciphertext ct^* .

Phase 5: Guess. $\mathcal{A}$ outputs a guess b' .

The advantage of $\mathcal{A}$ in this game is defined as

$$\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|.$$

The scheme is said to be IND-CPA secure if this advantage is negligible.

4.2.2. Anonymity Security

The anonymity security is defined via a security game $\text{Game}_{\text{A}^2\text{ABE}}^{\text{anonymity}}(\mathcal{A}, \lambda)$ between $\mathcal{C}$ and $\mathcal{A}$.

Phase 1 and 2: Identical to those in the IND-CPA security model.

Phase 3: Challenge. $\mathcal{A}$ provides two access policies (M, π_0) and (M, π_1) that have identical attribute–name matrices but distinct value assignments, along with two messages m_0 and m_1 of equal length. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, invokes CAEncryption to encrypt m_b under policy (M, π_b) , and returns the ciphertext ct to $\mathcal{A}$. The policy mapping satisfies $\pi_b(i) = (n_i, v_{b,i})$.

Phase 4: Query Phase 2. Phase 3 proceeds analogously to Phase 2, with the additional restriction that no attribute set S submitted for queries may satisfy the target policy (M, π_b) , and $\mathcal{A}$ is prohibited from requesting any attribute set that fulfills the challenge structure (M, π_b) .

Phase 5: Guess. $\mathcal{A}$ outputs a guess b' .

The advantage of $\mathcal{A}$ in the anonymity game is defined as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{anonymity}}(\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|.$$

The scheme is said to be attribute-anonymous if this advantage is negligible.

4.3. Practical Considerations for IoT Deployment

This section analyzes the practical operational implications of LOR-A2ABE's revocation mechanism, complementing the formal security proofs. The following technical discussion explains why the scheme remains effective in real IoT deployments under resource constraints.

The core of the LOR-A2ABE revocation mechanism is the version-timestamp chain. For a device created and issued a key at epoch 1, its secret key component evolves at epoch t as follows:

$$sk_5^{(t)} = e(g_1, g_2)^{\alpha \cdot \sum_{k=1}^t H(\text{nonce}^{(k)} \| \text{Ver}^{(k)})}. \quad (28)$$

Similarly, a ciphertext created at epoch 1 evolves at epoch t into the following component:

$$ct_2^{(t)} = g_2^{s^{(t)} + \sum_{k=1}^t H(\text{nonce}^{(k)} \| \text{Ver}^{(k)})}. \quad (29)$$

Decryption requires the version-timestamp terms in $ct_2^{(t)}$ and $sk_5^{(t)}$ to cancel each other out. Therefore, for a device revoked at epoch r to decrypt a ciphertext from a later epoch $t > r$, it must possess the complete and consistent chain of incremental values:

$$\mathcal{CHAIN}_{r+1 \rightarrow t} = \left\{ e(g_1, g_2)^{\alpha \cdot H(\text{nonce}^{(k)} \| \text{Ver}^{(k)})} \right\}_{k=r+1}^t \quad (30)$$

and its updated secret key must satisfy:

$$sk_5^{(t)} = sk_5^{(r)} \cdot \prod_{k=r+1}^t e(g_1, g_2)^{\alpha \cdot H(\text{nonce}^{(k)} \| \text{Ver}^{(k)})}. \quad (31)$$

If any term in this product is missing during the update to $sk_5^{(t)}$, the version-timestamp components in $sk_5^{(t)}$ and $ct_2^{(t)}$ will not align, thereby preventing successful decryption.

Consequently, achieving limited collusion resistance in LOR-A2ABE imposes the following requirements on an adversary:

- **Complete Version-Timestamp Chain Requirement:** A revoked device must obtain every *key_update* value from its revocation epoch onward. Missing even a single update breaks the chain integrity.
- **Persistent Collusion Requirement:** Colluding devices must share *key_update* values continuously and in real-time, rather than through one-time exchanges.
- **Increasing Attack Complexity:** The longer a device remains revoked, the more *key_update* values it must collect, making sustained collusion progressively more difficult to maintain.

However, practical constraints in real-world “cloud–terminal” centralized IoT environments render such collusion highly challenging:

- **Update Frequency:** In real-time IoT systems, key updates typically occur every few seconds to minutes.
- **Continuous Monitoring Requirement:** To maintain decryption capability, a revoked device would need to receive *key_update* values continuously and reliably throughout the revocation period.

- **Difficulty in Maintaining Covert Channels:** Establishing and sustaining a hidden communication channel that operates at the same frequency as system updates is particularly challenging for resource-constrained IoT devices with limited computational and communication capabilities.
- **Detection Risk:** Such persistent, periodic communication patterns are more susceptible to detection by network monitoring and intrusion detection systems.

The practical effectiveness of LOR-A2ABE in IoT environments is demonstrated by its requirement for strict continuity of the version chain, which raises the practical barrier for mounting collusion attacks beyond the reach of typical resource-constrained terminals. This approach achieves the dual objectives of ensuring highly efficient ($\mathcal{O}(1)$) revocation—a critical requirement for large-scale IoT deployments—while providing effective protection against realistic threats in actual operational scenarios. Ultimately, this design aligns with the security realities of IoT ecosystems, where operational efficiency and protection against sporadic, opportunistic attacks take precedence over defending against unbounded theoretical adversaries.

4.4. IND-CPA Security Proof

Theorem 1. *Our proposed CP-ABE scheme achieves adaptive IND-CPA security within the random oracle model, relying on the Decisional Linear (DLIN) assumption.*

Proof of Theorem 1. Suppose an adversary $\mathcal{A}$ can win the security game $\text{Game}_{\text{CP-ABE}}^{\text{IND-CPA}}(\mathcal{A}, \lambda)$ with a non-negligible advantage ε . In this case, we can build a simulator $\mathcal{B}$ that utilizes $\mathcal{A}$ to solve the DLIN problem with advantage ε .

Given a DLIN instance defined as follows:

$$A = g_1, B = g_2, C = g_1^{b_1}, D = g_1^{b_2}, E = g_2^{b_1}, F = g_2^{b_2}, \quad (32)$$

$$G = g_1^{b_1 s_1}, H = g_1^{b_2 s_2}, I = g_2^{b_1 s_1}, J = g_2^{b_2 s_2}, Y, Z, \quad (33)$$

where $s = s_1 + s_2$, $R \xleftarrow{\$} \mathbb{Z}_p$, and the challenger $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$. If $b = 0$, set $(Y, Z) = (g_1^s, g_2^s)$; otherwise $(Y, Z) = (g_1^R, g_2^R)$. The simulator $\mathcal{B}$ guesses the value of b as follows.

Phase 1: Initialization. $\mathcal{B}$ picks $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and sets:

$$pk = (H, p, G_1, G_2, G_T, e, B^\beta, A, B, E, F, e(A, B)^\alpha), \quad (34)$$

$$msk = (\alpha, \beta). \quad (35)$$

$\mathcal{B}$ sends pk to the adversary $\mathcal{A}$ and keeps msk secret.

Phase 2: Query Phase 1. $\mathcal{A}$ is allowed to make a polynomial number of queries to the following oracles:

- **Random oracle $H(\cdot)$:** $\mathcal{B}$ maintains the lists $P = \langle H(x_i), h_i, k_i \rangle$, $L = \langle H(0||x_i) + 1, h_{0,i}, t_i \rangle$, $Q = \langle H(1||x_i) + 1, h_{1,i}, \sigma_i \rangle$, $W = \langle H(0||x_i) + 1, hh_{0,i}, w_i \rangle$, $N = \langle H(1||x_i) + 1, hh_{1,i}, \delta_i \rangle$, where $h_i = A^{k_i} = g_1^{k_i}$, $h_{0,i} = C^{t_i} = g_1^{b_1 t_i}$, $h_{1,i} = D^{\sigma_i} = g_1^{b_2 \sigma_i}$, $hh_{0,i} = E^{w_i} = g_2^{b_1 w_i}$, $hh_{1,i} = F^{\delta_i} = g_2^{b_2 \delta_i}$. On a query for $H(x_i)$, $H(0||x_i) + 1$ or $H(1||x_i) + 1$, $\mathcal{B}$ checks whether the value already exists in the corresponding list. If yes, it returns the recorded h_i , $h_{0,i}$, $h_{1,i}$, $hh_{0,i}$ or $hh_{1,i}$; otherwise it picks a random $\xi_i \xleftarrow{\$} \mathbb{Z}_p$ and returns A^{ξ_i} , C^{ξ_i} , D^{ξ_i} , E^{ξ_i} or F^{ξ_i} , respectively, and stores the new entry in the list.

- Key-generation oracle $O_{\text{KeyGen}}(\cdot)$: $\mathcal{A}$ submits an attribute set $S = \{u_i\}$. $\mathcal{B}$ queries $H(\cdot)$ for each u_i to obtain $h_i, h_{0,i}$ and $h_{1,i}$. Choosing random $r, ver, nonce \xleftarrow{\$} \mathbb{Z}_p$, $\mathcal{B}$ constructs the secret key as:

$$sk_1 = A^r, \quad sk_2 = A^{\alpha+\beta r}, \quad sk_{3,u} = A^{t_i r}, \quad sk_{4,u} = A^{\sigma_i r}, \quad (36)$$

$$sk_5 = e(A^\alpha, B)^{H(nonce||ver)}. \quad (37)$$

Indeed,

$$sk_{3,u} = g_1^{H(0||u_i)r/b_1} g_1^{r/b_1} = g_1^{(H(0||u_i)+1)r/b_1} = g_1^{(b_1 t_i)r/b_1} = A^{t_i r}, \quad (38)$$

$$sk_{4,u} = g_1^{H(1||u_i)r/b_1} g_1^{r/b_1} = g_1^{(H(1||u_i)+1)r/b_1} = g_1^{(b_1 \sigma_i)r/b_1} = A^{\sigma_i r}. \quad (39)$$

$\mathcal{B}$ returns $sk_S = (sk_1, sk_2, \{sk_{3,u}, sk_{4,u}\}_{u \in S}, sk_5)$ to $\mathcal{A}$ and records the user information in the user list UL.

- Revocation oracle $O_{\text{revok}}(\cdot)$: $\mathcal{A}$ submits a user uid to be revoked, provided $uid \in \text{UL}$ and $uid \notin \text{RL}$. $\mathcal{B}$ picks new $ver', nonce', s' \xleftarrow{\$} \mathbb{Z}_p$ and sets

$$update = H(nonce' || ver'), \quad (40)$$

$$key_update = e(A^\alpha, B)^{update}. \quad (41)$$

It then simulates the updated ciphertext:

$$ct'_0 = ct_0 \cdot e(A, B)^{as'}, \quad (42)$$

$$ct'_2 = ct_2 \cdot B^{s'+update}, \quad (43)$$

$$ct'_5 = ct_5 \cdot B^{\beta(s'+update)}, \quad (44)$$

and returns $ct' = (ct'_0, \{ct'_{1,i}\}_{i \in [n_1]}, ct'_2, ct_3, ct_4, ct'_5)$ together with key_update to $\mathcal{A}$. $\mathcal{B}$ records $\text{Record}[uid] = \{ver', nonce', update, key_update\}$.

- Key-update oracle $O_{\text{KeyUpdate}}(\cdot)$: $\mathcal{A}$ submits uid and key_update , provided $uid \in \text{UL}$ and $uid \notin \text{RL}$. $\mathcal{B}$ updates the user's secret-key component as $sk'_5 = sk_5 \cdot key_update$ and returns the updated key.
- Outsourced-decryption oracle $O_{\text{Transform}}(\cdot)$: $\mathcal{A}$ provides a ciphertext ct and a user uid whose attribute set meets the access policy associated with ct . Assuming there exist coefficients $\gamma_i i \in I$ satisfying $\sum i \in I \gamma_i M_i = (1, 0, \dots, 0)$, $\mathcal{B}$ proceeds to compute the transformation key:

$$TK = \frac{e(sk_2, ct_2) e(\prod_{i \in I} (ct_{1,\pi(i)})^{\gamma_i}, sk_1)}{e(sk_1, ct_5) e(\prod_{i \in I} (sk_{3,i})^{\gamma_i}, ct_3) e(\prod_{i \in I} (sk_{4,i})^{\gamma_i}, ct_4)}, \quad (45)$$

and returns TK to $\mathcal{A}$.

Phase 3: Challenge. $\mathcal{A}$ selects two messages m_0 and m_1 of the same length, along with a target access policy (M', π') . This choice is subject to the constraint that all attribute sets previously queried in Phase 2 are unable to satisfy (M', π') . $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$ and encrypts m_b as follows.

For each attribute $\pi(i)^*$, $\mathcal{B}$ queries $H(\cdot)$ to obtain $h_{0,i}, h_{1,i}, hh_{0,i}, hh_{1,i}$, picks random $s_1, s_2 \xleftarrow{\$} \mathbb{Z}_p$ and a vector $v \xleftarrow{\$} \mathbb{Z}_p^{n-1}$. In the exponent we set $H(0||\pi(i)) + 1 = b_1 w_i$, $H(1||\pi(i)) + 1 = b_2 \delta_i$. Then:

$$ct_0^* = m_b \cdot e(Y, B)^\alpha, \quad (46)$$

$$ct_{1,i}^* = g_2^{M_i^*(s||v)^\top} \cdot g_2^{s_1(H(0||\pi(i))+1)} \cdot g_2^{s_2(H(1||\pi(i))+1)} = Z^{M_{i,1}^*} \cdot B^{\sum_{j=2}^n M_{i,j}^* v_j} \cdot I^{w_i} \cdot J^{\delta_i}, \quad (47)$$

$$ct_2^* = g_2^s \cdot g_2^{H(\text{nonce}||\text{ver})} = Z \cdot B^{H(\text{nonce}||\text{ver})}, \quad (48)$$

$$ct_3^* = I, \quad (49)$$

$$ct_4^* = J, \quad (50)$$

$$ct_5^* = g_2^{s(\beta+1)} \cdot g_2^{\beta H(\text{nonce}||\text{ver})} = Z^{\beta+1} \cdot B^{\beta H(\text{nonce}||\text{ver})}, \quad (51)$$

$$ct^* = (ct_0^*, \{ct_{1,i}^*\}_{i \in [n_1]}, ct_2^*, ct_3^*, ct_4^*, ct_5^*). \quad (52)$$

$\mathcal{B}$ sends ct^* to $\mathcal{A}$.

Phase 4: Query Phase 2. $\mathcal{A}$ may again issue a polynomial number of queries, subject to the same restrictions as in Phase 2 and the additional constraints:

- No $O_{\text{KeyGen}}(S)$ query may use an attribute set S satisfying (M^*, π^*) .
- No $O_{\text{revok}}(uid)$ query is allowed for a user uid whose attribute set satisfies (M^*, π^*) and who already obtained the corresponding key in Phase 2.
- No $O_{\text{KeyUpdate}}(uid, update)$ query may use an update for a revoked user.
- The oracle $O_{\text{Transform}}(\cdot)$ cannot be queried on the challenge ciphertext ct^* .

Phase 5: Guess. $\mathcal{A}$ outputs a guess b' . If $b' = b$, $\mathcal{B}$ outputs $\mu' = 0$ (meaning that (Y, Z) is a valid DLIN tuple); otherwise $\mathcal{B}$ outputs $\mu' = 1$ (meaning that (Y, Z) is random).

When $\mu = 0$, $\mathcal{A}$ has advantage ε , so $\Pr[b' = b \mid \mu = 0] = \frac{1}{2} + \varepsilon$; when $\mu = 1$, $\mathcal{A}$ has no advantage, so $\Pr[b' = b \mid \mu = 1] = \frac{1}{2}$. Consequently,

$$\Pr[\mu' = \mu] = \frac{1}{2} \Pr[\mu' = \mu \mid \mu = 0] + \frac{1}{2} \Pr[\mu' = \mu \mid \mu = 1] = \frac{1}{2} \left(\frac{1}{2} + \varepsilon \right) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} + \frac{\varepsilon}{2}. \quad (53)$$

Thus, the advantage of $\mathcal{B}$ in solving the DLIN problem is

$$\text{Adv}_{\mathcal{B}}^{\text{DLIN}}(\lambda) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right| = \frac{\varepsilon}{2}. \quad (54)$$

This completes the proof. $\square$

4.5. Anonymity Security Proof

Theorem 2. Our proposed CP-ABE scheme achieves adaptive anonymity within the random oracle model, relying on the Decisional Linear (DLIN) assumption.

Proof. Suppose an adversary $\mathcal{A}$ can win the security game $\text{Game}_{\text{CP-ABE}}^{\text{IND-CPA}}(\mathcal{A}, \lambda)$ with a non-negligible advantage ε . In this case, we can build a simulator $\mathcal{B}$ that utilizes $\mathcal{A}$ to solve the DLIN problem with advantage ε .

The DLIN instance is given as in the IND-CPA proof. $\mathcal{B}$ proceeds as follows.

Phase 1 and 2: Identical to those in the IND-CPA proof.

Phase 3: Challenge. $\mathcal{A}$ submits two challenge access policies (M^*, π_0^*) and (M^*, π_1^*) that share the same attribute–name structure but differ in attribute values, together with two equal-length messages m_0, m_1 . Here $\pi_b^*(i) = (n_{\pi(i)}, v_{\pi(i)}^b)$ for $b \in \{0, 1\}$.

$\mathcal{B}$ randomly picks $b \in \{0, 1\}$ and encrypts m_b under (M^*, π_b^*) . For each attribute $\pi_b^*(i)$, $\mathcal{B}$ queries $H(\cdot)$ to obtain $h_{0,i}, h_{1,i}, hh_{0,i}, hh_{1,i}$. Because $v_{\pi(i)}^0 \neq v_{\pi(i)}^1$, the values w_i^0 (from $hh_{0,i} = g_2^{b_1 w_i^0}$) and w_i^1 , as well as δ_i^0 (from $hh_{1,i} = g_2^{b_2 \delta_i^0}$) and δ_i^1 , are independently random.

Setting $H(0||\pi(i)) + 1 = b_1 w_i^b$ and $H(1||\pi(i)) + 1 = b_2 \delta_i^b$ in the exponent, $\mathcal{B}$ computes the challenge ciphertext exactly as in the IND-CPA challenge phase:

$$ct_0^* = m_b \cdot e(Y, B)^\alpha, \quad (55)$$

$$ct_{1,i}^* = Z^{M_{i,1}^*} \cdot B^{\sum_{j=2}^n M_{i,j}^* v_j} \cdot I^{w_i^b} \cdot J^{\delta_i^b}, \quad (56)$$

$$ct_2^* = Z \cdot B^{H(\text{nonce} \| \text{ver})}, \quad (57)$$

$$ct_3^* = I, \quad (58)$$

$$ct_4^* = J, \quad (59)$$

$$ct_5^* = Z^{\beta+1} \cdot B^{\beta H(\text{nonce} \| \text{ver})}, \quad (60)$$

$$ct_b^* = (ct_0^*, \{ct_{1,i}^*\}_{i \in [n_1]}, ct_2^*, ct_3^*, ct_4^*, ct_5^*). \quad (61)$$

$\mathcal{B}$ sends ct_b^* to $\mathcal{A}$.

Analysis:

- If $b = 0$: then $(Y, Z) = (g_1^s, g_2^s)$, $I = g_2^{b_1 s_1}$, $J = g_2^{b_2 s_2}$. Substituting into $ct_{1,i}^*$ gives

$$ct_{1,i}^* = g_2^{s M_{i,1}^*} \cdot g_2^{\sum_{j=2}^n M_{i,j}^* v_j} \cdot (g_2^{s_1 b_1})^{w_i^b} \cdot (g_2^{s_2 b_2})^{\delta_i^b} = B^{M_i^* (s \| v)^\top + s_1 b_1 w_i^b + s_2 b_2 \delta_i^b}. \quad (62)$$

Because $H(0 \| \pi(i)) + 1 = b_1 w_i^b$ and $H(1 \| \pi(i)) + 1 = b_2 \delta_i^b$, we have

$$ct_{1,i}^* = g_2^{M_i^* (s \| v)^\top + s_1 (H(0 \| \pi(i)) + 1) + s_2 (H(1 \| \pi(i)) + 1)} = B^{M_i^* (s \| v)^\top + s + s_1 H(0 \| \pi(i)) + s_2 H(1 \| \pi(i))}, \quad (63)$$

which is a valid encryption of m_b under (M^*, π_b^*) with attribute value $v_{\pi(i)}^b$.

- If $b = 1$: then $(Y, Z) = (g_1^R, g_2^R)$, and

$$ct_{1,i}^* = B^{R M_{i,1}^*} \cdot B^{\sum_{j=2}^n M_{i,j}^* v_j} \cdot B^{s_1 b_1 w_i^b} \cdot B^{s_2 b_2 \delta_i^b}. \quad (64)$$

The term $g_2^{R M_{i,1}^*}$ is completely random, making $ct_{1,i}^*$ a random element of G_2 that carries no information about the attribute value $v_{\pi(i)}^b$.

Phase 4: Query Phase 2. $\mathcal{A}$ may again issue queries, with the restriction that every queried attribute set S must not satisfy (M^*, π_b^*) , and $\mathcal{A}$ is forbidden to query any attribute set matching the challenge policy (M^*, π_b^*) .

Phase 5: Guess. $\mathcal{A}$ outputs a guess b' . If $b' = b$, $\mathcal{B}$ outputs $\mu' = 0$; otherwise $\mathcal{B}$ outputs $\mu' = 1$.

When $\mu = 0$, $\mathcal{A}$ has advantage ϵ' , so $\Pr[b' = b \mid \mu = 0] = \frac{1}{2} + \epsilon'$; when $\mu = 1$, $\Pr[b' = b \mid \mu = 1] = \frac{1}{2}$. Hence

$$\Pr[\mu' = \mu] = \frac{1}{2} \left(\frac{1}{2} + \epsilon' \right) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} + \frac{\epsilon'}{2}, \quad (65)$$

and

$$\text{Adv}_{\mathcal{B}}^{\text{anonymous}}(\lambda) = \left| \Pr[\mu' = \mu] - \frac{1}{2} \right| = \frac{\epsilon'}{2}. \quad (66)$$

Thus, the scheme is attribute-anonymous under the DLIN assumption. $\square$

4.6. Corollaries

1. **Revocation Security:** Our scheme guarantees forward secrecy: a user revoked at time (epoch) t_r cannot decrypt any ciphertext generated or updated after t_r , even if the user's pre-revocation secret key is compromised.

This property is intrinsically ensured by our revocation mechanism and is covered by the IND-CPA security proof. Recall the decryption process from Section 3.2: successful

recovery of the session key requires the cancellation of the version-timestamp term between sk_5 and ct_2 .

For a user revoked at t_r , its sk_5 component is frozen with the hash value $h_{old} = H(\text{nonce}_{old} \| \text{Ver}_{old})$ from the last update it received. Any ciphertext ct created or refreshed after t_r will be associated with a new, independent hash value $h_{new} = H(\text{nonce}_{new} \| \text{Ver}_{new})$ in its ct_2 component. Since the revoked user does not receive the $\text{key_update} = e(g_1^\alpha, g_2)^{h_{new}}$, its sk_5 remains $e(g_1^\alpha, g_2)^{h_{old}}$. The mismatch ($h_{old} \neq h_{new}$) prevents the necessary cancellation in the decryption equation, making decryption fail.

Formally, in the IND-CPA security game $\text{Game}_{\text{CP-ABE}}^{\text{IND-CPA}}$ (Section 4.2.1), an adversary $\mathcal{A}$ modeling a forward secrecy attack would need to, after revoking a user, obtain a ciphertext encrypted under a new version/timestamp. The game's restriction that "the $O_{\text{KeyUpdate}}$ oracle cannot be queried for a revoked user" (Phase 4) directly prohibits the adversary from acquiring the necessary updated key component. Therefore, any advantage $\mathcal{A}_{\text{FS}}$ might have in breaking forward secrecy translates into an advantage in the standard IND-CPA game against the updated system parameters, which is proven to be negligible under the DLIN assumption in Theorem 1. Hence, our scheme provides forward secrecy.

2. Forward Secrecy: If a user uid is revoked at time t , it cannot decrypt any ciphertext generated after t . This holds because:
 - After revocation, sk_5 does not contain the new version number and timestamp, so it cannot cancel the corresponding terms in fresh ciphertexts.
 - In the IND-CPA game, the adversary is prohibited from updating the keys of revoked users whose attribute sets satisfy the challenge policy.

Hence revoked users cannot adapt to the new version, achieving forward secrecy. The forward-security property is already embedded in the IND-CPA security proof, and our scheme provides forward secrecy under the DLIN assumption.

5. Performance Evaluation

5.1. Theoretical Analysis

As shown in Table 1, we present a theoretical comparison of the computational complexity among various schemes in terms of key generation, encryption overhead, and total decryption overhead (including both user-side and via cloud server). Additionally, Table 1 compares the functional features of these schemes.

Table 1. Comparison of computational costs among different attribute-based encryption schemes.

Schemes	Key Generation	Encryption	Decryption
BSW [23]	$[(m+1)M + (m+2)E + mH]G_1 + [mE]G_2$	$[lE + lH]G_1 + [(l+1)E]G_2$	$[(2x+1)H]G_T + (2x+1)P$
CGW [27]	$[(3m+6)E]G_2$	$[6lnM + (6l+9n)E]G_1$	$[3xM + 3xE + 6H]G_T + 6P$
ABGW [28]	$[(2m+1)E]G_2$	$[2lM + 5lE]G_1$	$[2xM + (x+2)H]G_T + (x+2)P$
FAME [29]	$[(6m+9)M + (9m+9)E + (6m+6)H]G_1 + [3E]G_2$	$[(6ln+3l)M + 6lE + (6l+6n)H]G_1 + [3E]G_2$	$[(6x+3)M + 6H]G_T + 6P$
FABEO [30]	$[1M + (m+2)E + (m+1)H]G_1 + [1E]G_2$	$[lM + 2lE + (l+1)H]G_1 + [2E]G_2$	$[2xM + 3H]G_T + 3P$
FABESA [31]	$[1M + (2m+2)E + 2mH]G_1 + [1E]G_2$	$[2lM + 3lE + 2lH]G_1 + [3E]G_2$	$[3xM + 4H]G_T + 4P$
LOR-A2ABE	$[(4m+2)E + (2m+1)H]G_1 + [1E]G_T + 1P$	$[(l+4)E + lH]G_2$	$[3(x-1)M + 3xE]G_1 + [xM + xE]G_2 + [2M]G_T + 5P$

Notations: m : number of attributes; l, n : rows and columns of MSP matrix; x : attributes used in decryption. Abbreviations: M : Mul, E : Exp, H : Hash, P : Pairing.

Compared with other schemes, the LOR-A2ABE scheme introduces hash and pairing operations during the key generation phase, resulting in higher computational overhead than lightweight schemes such as BSW and FABEO. Nevertheless, these operations enable lightweight user revocation functionality.

In the encryption phase, a notable characteristic of the LOR-A2ABE scheme is that all encryption computations are performed in the G_2 group. On the BN254 curve used in our implementation, a single exponentiation in G_2 is approximately 3–5 times slower than in G_1 according to well-established benchmarking studies [39]. This quantitative performance gap directly explains the slower encryption time of LOR-A2ABE despite its comparable operation count to other schemes. Nevertheless, in the LOR-A2ABE G_2 is used for ciphertext computation and G_1 for key computation. This design represents an optimization strategy: it reduces the waiting time for users during key issuance while maintaining lightweight final decryption overhead at the client side—where users only perform division operations in the G_T group. Since encryption is handled by the resource-sufficient Certificate Authority (CA) and ciphertexts are stored in the cloud server after encryption, the computational cost of G_2 operations does not impact client-side performance. This constitutes a reasonable trade-off in IoT environments.

In the total decryption phase, unlike schemes such as BSW whose costs grow linearly with the number of attributes x , the LOR-A2ABE scheme involves a fixed number of six pairing operations, which is only slightly higher than those of FABEO and FABESA. This represents a significant advantage of the LOR-A2ABE scheme. Moreover, it is worth noting that while supporting user revocation, LOR-A2ABE also outsources a substantial portion of the decryption computations to the cloud server. The user side only needs to perform constant-level lightweight operations to complete decryption, further reducing the computational burden on the client.

In terms of functionality, the LOR-A2ABE scheme achieves anonymity, which enhances user privacy protection. It also supports outsourced decryption and lightweight user revocation, outperforming other schemes in terms of functional completeness.

Theoretical analysis indicates that although the LOR-A2ABE scheme exhibits higher computational overhead in the key generation and encryption phases, it offers notable advantages in decryption efficiency and functional integrity. It is particularly suitable for resource-constrained, decryption-intensive, and dynamically managed centralized IoT environments, demonstrating strong theoretical justification and practical applicability.

5.2. Experimental Analysis

As shown in Figure 2, we implemented the proposed LOR-A2ABE scheme and performed a comprehensive performance comparison with several existing schemes. All experiments were conducted in a Python 3.10 environment using the Charm-Crypto 0.5 library. The hardware platform consisted of an ASUS TUF Gaming A15 laptop equipped with an AMD Ryzen 7 7940H processor and 16 GB RAM. All compared schemes employ the industry-standard asymmetric pairing BN254 curve, ensuring consistent security assumptions for fair performance evaluation. Furthermore, we simulated a worst-case experimental scenario: all attributes in the access policy were concatenated using AND gates (e.g., “(1 AND 2) AND 3”). The number of attributes was incrementally increased from 10 to 500 in steps of 20. Each data point in Figure 2 represents the average execution time over ten independent runs. A full statistical analysis including variance metrics is a subject for future extended benchmarking.

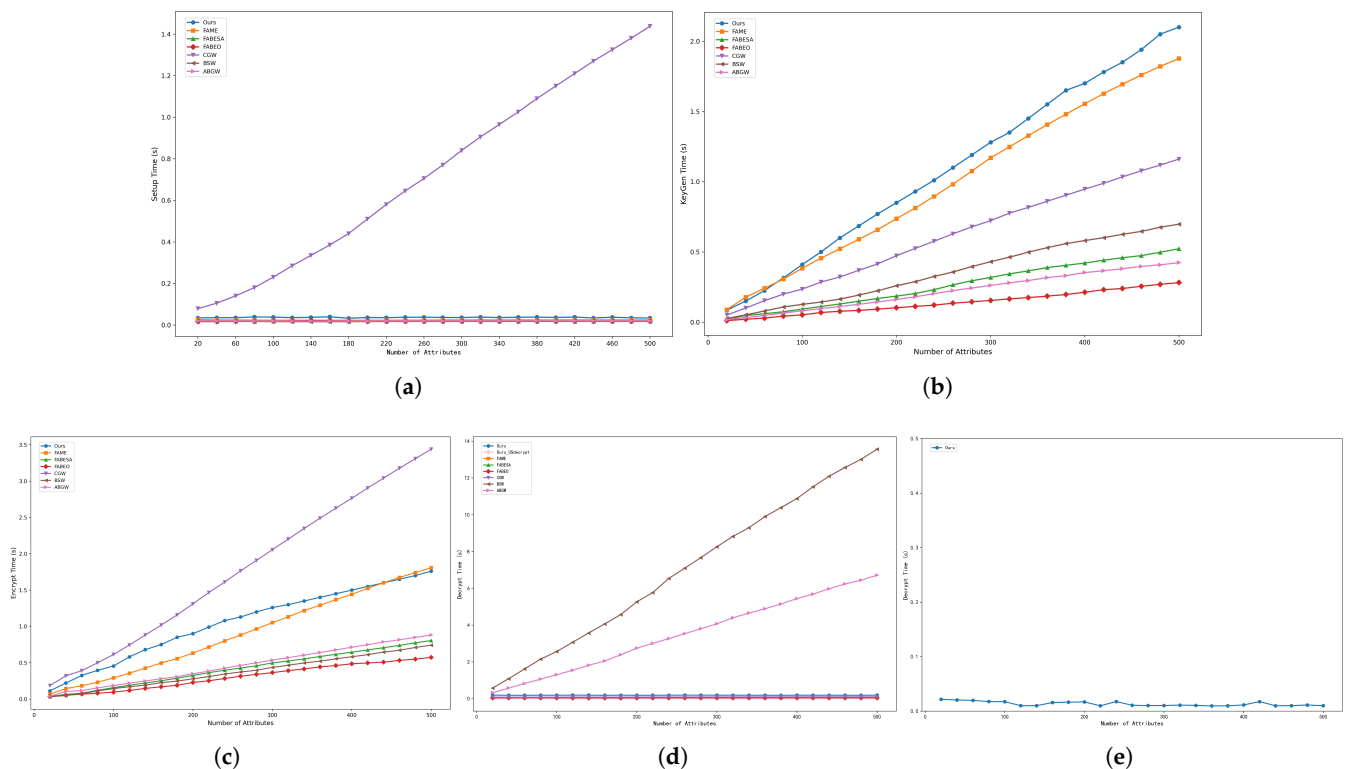


Figure 2. Performance comparison of different schemes: (a) Setup time, (b) Key generation time, (c) Encryption time, (d) Total decryption time, and (e) Revocation time.

The detailed experimental analysis is presented as follows.

- **Setup Time (Figure 2a):** The LOR-A2ABE scheme demonstrates significantly higher efficiency in setup time compared to the CGW schemes (which grows from 79 ms to 1.44 s). While it exhibits a slightly longer initialization time (≈ 35 ms) than schemes such as FABESA (≈ 19 ms) and FABEO (≈ 18 ms), its overall setup duration remains low and stable, showcasing the scheme's reliable and efficient initialization characteristics.
- **Key Generation Time (Figure 2b):** The experimental results for key generation align with the theoretical analysis. The introduction of a hash operation and an additional pairing to enable the revocation feature results in a higher key issuance time for LOR-A2ABE (86 ms to 2.1 s) compared to other schemes (e.g., FABEO: 11 ms to 281 ms). However, this overhead remains comparable to that of the FAME scheme (89 ms to 1.88 s) and within an acceptable range. It is important to note that LOR-A2ABE targets “cloud-terminal” Centralized IoT Management Environments, where data acquisition systems are relatively static, and key issuance is not a frequent operation. Consequently, the increased key generation overhead has minimal practical impact in real-world deployment.
- **Encryption Time (Figure 2c):** Consistent with the theoretical expectations, the encryption time of LOR-A2ABE (113 ms to 1.76 s) is marginally higher than that of other schemes (e.g., FABESA: 34 ms to 805 ms) due to the concentration of all computations within the G_2 group. Nevertheless, its time growth curve exhibits a relatively flat trend, in contrast to the near-linear growth observed in schemes like CGW (184 ms to 3.44 s) and FAME (68 ms to 1.81 s). This indicates that the LOR-A2ABE scheme can maintain stable encryption efficiency even when handling a large number of attributes.
- **Total Decryption Time (Figure 2d):** The LOR-A2ABE scheme demonstrates a distinct advantage in total decryption time (constant ≈ 185 ms). Its performance is comparable to that of efficient schemes like FABESA (≈ 60 ms) and FABEO (≈ 45 ms), and crucially,

it remains largely invariant with respect to the number of attributes—unlike BSW (572 ms to 13.56 s) and ABGW (316 ms to 6.71 s). Notably, when employing outsourced decryption techniques, the user-side decryption efficiency of LOR-A2ABE drops to only ≈ 0.2 ms, which is the fastest among all evaluated schemes. This makes it particularly suitable for environments where decryption operations are frequent and user-end resources are severely constrained.

- **Revocation Time (Figure 2e):** The LOR-A2ABE scheme exhibits constant and low revocation overhead (9.7–22 ms) as the number of attributes increases. This scalability characteristic demonstrates the efficiency of its partial update mechanism, which avoids the linear-cost re-encryption required by traditional revocation approaches.

The experimental analysis confirms that the results for the LOR-A2ABE scheme are in precise agreement with the theoretical predictions. Specifically designed for “cloud-terminal” Centralized IoT Management Environments, LOR-A2ABE successfully achieves lightweight revocation and efficient outsourced decryption, underscoring its significant practical deployment value.

5.3. Theoretical Analysis of Revocation Mechanisms

In Sections 5.1 and 5.2, we systematically analyzed the performance of relevant CP-ABE schemes during the initialization, key generation, encryption, and decryption phases through both theoretical and experimental comparisons with LOR-A2ABE. However, in terms of revocation functionality, most existing CP-ABE schemes that support revocation are implemented based on the SS512 curve [36,37], whereas LOR-A2ABE adopts the BN254 curve, which offers stronger security guarantees. Since these two curves correspond to different security assumptions and parameter configurations, a direct experimental performance comparison would be invalid due to these differing security foundations. We therefore focus on theoretical complexity analysis, which provides meaningful insights across security levels. The following analysis examines algorithmic complexity characteristics within each scheme’s respective security context, rather than making direct performance comparisons.

As shown in Table 2, the theoretical complexity analysis reveals that our proposed LOR-A2ABE scheme achieves the most favorable complexity profile for revocation operations, making it particularly suitable for centralized IoT systems that require frequent and efficient revocation operations.

Table 2. Theoretical complexity comparison of revocation mechanisms.

Scheme	Security Level	Key Update	Ciphertext Update	Outsourced Decryption
LOR-A2ABE	128-bit (BN254)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✓
MTA [36]	96-bit (SS512)	$\mathcal{O}(A)$	$\mathcal{O}(1)$	×
TRH [37]	96-bit (SS512)	$\mathcal{O}(\log U)$	$\mathcal{O}(r)$	×

✓: supported; ×: not supported. $|A|$: number of attributes, $|U|$: total users, r : size of revocation cover set. Note: Complexity comparisons are presented within their respective security contexts.

In terms of computational complexity for key update, the LOR-A2ABE scheme achieves constant complexity $\mathcal{O}(1)$, requiring only one group multiplication per user. Notably, this efficiency is maintained at the higher 128-bit security level (BN254). In comparison, the MTA scheme requires users to update all non-revoked attributes, resulting in linear complexity $\mathcal{O}(|A|)$. Meanwhile, the TRH scheme incurs logarithmic overhead $\mathcal{O}(\log |U|)$ due to its binary tree structure. These comparisons highlight LOR-A2ABE’s algorithmic efficiency advantage within its security context.

In terms of computational complexity for ciphertext update, LOR-A2ABE enables constant-complexity updates $\mathcal{O}(1)$ through partial updates. This contrasts with MTA's approach which, although also $\mathcal{O}(1)$, requires full ciphertext re-encryption. Meanwhile, TRH's complexity $\mathcal{O}(r)$ depends on the revocation cover set size, which may increase with system scale. The constant overhead of LOR-A2ABE remains advantageous within its higher-security framework.

In terms of theoretical complexity, LOR-A2ABE demonstrates clear advantages for IoT systems. Unlike MTA's linear key update cost or TRH's cover-set-dependent ciphertext update, LOR-A2ABE provides constant-time operations while operating at a higher security level. These characteristics align well with IoT requirements including resource constraints and frequent revocation events.

In summary, the theoretical complexity analysis indicates that LOR-A2ABE achieves constant-time key and ciphertext updates ($\mathcal{O}(1)$) within its 128-bit security framework, making it suitable for resource-constrained IoT architectures where both security and efficiency are critical.

5.4. Performance Inference and Analysis for IoT Terminals

Although the performance evaluation of this scheme was conducted on a general-purpose computing platform (laptop), one of its core design objectives is to substantially reduce the computational burden on IoT terminal devices. To more rigorously assess the applicability of this scheme in real resource-constrained environments, this section combines benchmark data provided by Perazzo et al. [40] from their empirical study on IoT ABE performance to conduct a theoretical inference and comparative analysis of the decryption performance of the LOR-A2ABE scheme on terminal devices (client side). The following analysis provides theoretical performance inferences based on architectural comparisons.

The work of Perazzo et al. [40] presents the first comprehensive empirical measurements (time, energy consumption, battery life) of various ABE schemes, including the classic CP-ABE scheme BSW, on two typical constrained IoT hardware platforms: ESP32 and RE-Mote. Their key conclusion indicates that ABE schemes can be deployed on devices with extremely limited memory and computational resources while achieving acceptable battery life, provided the number of attributes does not exceed 10 and hardware-accelerated cryptography is utilized. Specifically, the BSW scheme requires approximately 1.2 s and consumes about 100 μWh to decrypt a ciphertext with 10 attributes (worst-case policy) on the ESP32 platform.

As described in the system model in Section 3.2, in our “cloud–terminal” centralized IoT architecture, key generation and encryption are performed by the Central Authority (CA), while the majority of decryption computations are offloaded to the resource-abundant Cloud Server (CS). Resource-constrained client devices are only responsible for executing lightweight final decryption steps. Based on the computational complexity analysis in Section 5.1, the client-side decryption in the BSW scheme requires $2x + 1$ bilinear pairing operations (where x is the number of matched attributes), with its overhead growing linearly with policy complexity. In contrast, the LOR-A2ABE scheme shifts all bilinear pairing operations to the CS. The client side merely performs two fixed modular division operations in the finite field G_T , resulting in a constant computational complexity of $\mathcal{O}(1)$.

Given these architectural differences, we can infer that on the same IoT hardware platform (e.g., ESP32), the decryption time and energy consumption on the LOR-A2ABE client side will no longer increase with the number of attributes but will remain at an extremely low constant level. Compared to the empirically measured second-level decryption latency and hundreds of microwatt-hour energy consumption of the BSW scheme with 10 attributes, the client side of our scheme involves only two basic modular operations. Its

expected overhead is bound to be at least one order of magnitude lower than the “second-hundreds of μWh ” level of BSW. Reference [40] has already confirmed that BSW is “feasible” on constrained terminals, while the computational load on the LOR-A2ABE client side is significantly lower than that of BSW. Therefore, it is reasonable to infer that under the same deployment conditions (number of attributes ≤ 10), the energy efficiency performance of LOR-A2ABE on the terminal side will be far superior to that of traditional schemes, with a more negligible practical impact on device battery life.

Consequently, although this work does not involve on-site IoT deployment, by integrating the structural advantages of our scheme (outsourced decryption, constant terminal overhead) with reliable third-party IoT performance benchmarks, we can theoretically and robustly argue that the LOR-A2ABE scheme fully meets the requirements for efficient operation on resource-constrained IoT terminals such as ESP32. Its client-side performance is significantly superior to that of traditional CP-ABE schemes that have been empirically validated as feasible.

6. Conclusions and Future Work

This paper addresses the access control requirements in “cloud–terminal” centralized IoT Management Environments by proposing a lightweight anonymous attribute-based encryption scheme with support for lightweight revocation and outsourced decryption. The scheme embeds hash functions into both the key and ciphertext structures, enabling efficient attribute revocation and outsourced decryption without introducing significant computational overhead. Security analysis demonstrates that the LOR-A2ABE scheme achieves IND-CPA security under the Decisional Linear Diffie–Hellman assumption. Both theoretical and experimental results show that the proposed scheme exhibits excellent efficiency in decryption and revocation operations. It is particularly suitable for IoT scenarios characterized by frequent data updates, resource-constrained terminals, and dynamically evolving access policies, thereby providing a feasible technical pathway for achieving secure and efficient fine-grained access control in practical deployments.

In future work, we plan to further investigate lattice-based post-quantum secure lightweight anonymous attribute-based encryption schemes to mitigate potential threats posed by quantum computing. This direction aims to offer both theoretical foundations and practical guidance for the long-term secure deployment of IoT systems. Furthermore, while the performance inference presented in Section 5.4 strongly supports the efficiency of LOR-A2ABE on resource-constrained terminals based on architectural comparison and established benchmarks, conducting direct on-device measurements on platforms such as ESP32 or ARM Cortex-M remains an important empirical step for future validation. Exploring system architectures that integrate with other emerging IoT-specific security constructs, such as hybrid access control models for smart environments [41], and with lightweight in-sensor computing frameworks, presents a valuable direction for building more comprehensive and efficient secure IoT systems.

Author Contributions: Conceptualization, D.G.; methodology, H.X.; formal analysis, D.G. and H.X.; investigation, D.G.; resources, S.Q.; writing—original draft preparation, D.G.; writing—review and editing, D.G. and S.Q.; supervision, S.Q.; project administration, S.Q.; funding acquisition, S.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (Grants 62241301, 12361044, 61762001), the Basic Research Project of Science and Technology Plan of Guizhou Province (Grant ZK[2023] General 022), and Guizhou Provincial Science and Technology Projects (Grant [2024] Qingnian 287).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

- Elhattab, K.; Abouelmehdi, K.; Elmoutaouakkil, A. Internet of things (iot) for smart city, agriculture and healthcare. *J. Theor. Appl. Inf. Technol.* **2022**, *100*, 1104–1112.
- Nassereddine, M.; Khang, A. Applications of internet of things (iot) in smart cities. In *Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy*; CRC Press: Boca Raton, FL, USA, 2024; pp. 109–136.
- Babayigit, B.; Abubaker, M. Industrial internet of things: A review of improvements over traditional scada systems for industrial automation. *IEEE Syst. J.* **2023**, *18*, 120–133.
- Dangana, M. Suitability of NB-IoT for indoor industrial environment: A survey and insights. *Sensors* **2021**, *21*, 5284.
- Shala, B. Novel trust consensus protocol and blockchain-based trust evaluation system for M2M application services. *Internet Things* **2019**, *7*, 100058.
- Luft, A.; Luft, N.; Arntz, K. A basic description logic for service-oriented architecture in factory planning and operational control in the age of industry 4.0. *Appl. Sci.* **2023**, *13*, 7610.
- Wang, W. Smart contract token-based privacy-preserving access control system for industrial internet of things. *Digit. Commun. Netw.* **2023**, *9*, 337–346.
- Al-Hawawreh, M. Securing the industrial internet of things against ransomware attacks: A comprehensive analysis of the emerging threat landscape and detection mechanisms. *J. Netw. Comput. Appl.* **2024**, *223*, 103809.
- Costa, B. Orchestration in fog computing: A comprehensive survey. *ACM Comput. Surv.* **2022**, *55*, 1–34.
- Aljubayrin, S. Dynamic offloading strategy for computational energy efficiency of wireless power transfer based MEC networks in industry 5.0. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101841.
- Abid, A.; Jemili, F.; Korbaa, O. Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques. *Clust. Comput.* **2024**, *27*, 2217–2238.
- Ganga Shirisha, M.S. Secured iot data management using aes encryption and blockchain technology. *J. Inf. Syst. Eng. Manag.* **2025**, *10*, 728–735.
- Ibrahim, Z.A.; Ilyas, M. Hidden hierarchy based on cipher-text attribute encryption for iot data privacy in cloud. *Comput. Mater. Contin.* **2023**, *76*, 939–956.
- Mohammed, M.A.; Abdul Wahab, H.B. Enhancing iot data security with lightweight blockchain and okamoto uchiyama homomorphic encryption. *CMES-Comput. Model. Eng. Sci.* **2024**, *138*, 1731–1748.
- Mishra, N.; Islam, S.H.; Zeadally, S. A survey on security and cryptographic perspective of industrial-internet-of-things. *Internet Things* **2024**, *25*, 101037.
- Aljrees, T. Enhancing iot security through a green and sustainable federated learning platform: Leveraging efficient encryption and the quondam signature algorithm. *Sensors* **2023**, *23*, 8090.
- Li, Y. Research on Distributed Operation Mode of Internet of Things. *J. Phys. Conf. Ser.* **2021**, *1827*, 012129.
- Bhalla, T.; Kaur, R. Convergence of Internet of Things (IoT) and Edge Computing: Enhancing Scalability and Efficiency Through Cloud Integration: A Review. *Int. J. Eng. Sci. Humanit.* **2024**, *14*, 233–239.
- Mehta, A.; Verma, R.K. QoS-aware edge server placement for collaborative predictive maintenance in industrial internet of things. *J. Supercomput.* **2024**, *80*, 19324–19350.
- Mishra, R.K.; Yadav, R.K.; Nath, P. Access control models and frameworks for the iot environment: Review, challenges, and future direction. *Wirel. Pers. Commun.* **2024**, *138*, 1671–1701.
- Duan, L. Secure and fine-grained data sharing in internet of things: Integration of interplanetary file system and cross-blockchain for access control. *IEEE Internet Things J.* **2025**, *12*, 37301–37308.
- Yao, M. Efficient and verifiable bilateral fine-grained access control for cloud-edge iot healthcare. *IEEE Internet Things J.* **2025**, *12*, 43181–43194.
- Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
- Hafizpour, H. New attribute-based encryption schemes with anonymous authentication and time limitation in fog computing. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7681.
- Chawla, S.; Gupta, N. A proxy-based and collusion resistant multi-authority revocable cpabe framework with efficient user and attribute-level revocation. *Int. J. Saf. Secur. Eng.* **2023**, *13*, 527–538.
- Zhao, H. Efficient key management scheme for health blockchain. *CAAI Trans. Intell. Technol.* **2018**, *3*, 114–118.

27. Chen, J.; Gay, R.; Wee, H. Improved dual system abe in prime-order groups via predicate encodings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2015.
28. Ambrona, M. Attribute-based encryption in the generic group model: Automated proofs and new constructions. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 647–664.
29. Agrawal, S.; Chase, M. FAME: Fast attribute-based message encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017*; Association for Computing Machinery: New York, NY, USA, 2017.
30. Riepel, D.; Wee, H. FABEO: Fast attribute-based encryption with optimal security. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 2491–2504.
31. Meng, L. FABESA: Fast (and anonymous) attribute-based encryption under standard assumption. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, Salt Lake City, UT, USA, 14–18 October 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 4688–4702.
32. Yang, H.; Feng, T.; Yan, Y. Implementing efficient attribute encryption in iov under cloud environments. *Comput. Netw.* **2022**, *218*, 109363.
33. Zhang, Z.; Zhou, S. A decentralized strongly secure attribute-based encryption and authentication scheme for distributed internet of mobile things. *Comput. Netw.* **2021**, *201*, 108553.
34. Luo, W. FOC-PH-CP-ABE: An efficient CP-ABE scheme with fully outsourced computation and policy-hidden in the industrial internet of things. *IEEE Sens. J.* **2024**, *24*, 28971–28981.
35. Tao, Y. ORR-CP-ABE: A secure and efficient outsourced attribute-based encryption scheme with decryption results reuse. *Future Gener. Comput. Syst.* **2024**, *161*, 559–571.
36. Deng, S. Flexible revocation in ciphertext-policy attribute-based encryption with verifiable ciphertext delegation. *Multimed. Tools Appl.* **2023**, *82*, 22251–22274.
37. Han, D.; Pan, N.; Li, K.-C. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 316–327.
38. Lai, J.; Deng, R.H.; Li Y. Expressive CP-ABE with partially hidden access structures. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, Republic of Korea, 2–4 May 2012*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 18–19.
39. Beuchat, J.-L.; González-Díaz, J.E.; Mitsunari, S.; Okamoto, E.; Rodríguez-Henríquez, F.; Teruya, T. High-speed software implementation of the optimal ate pairing over Barreto–Naehrig curves. In *Pairing-Based Cryptography—Pairing 2010*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6487, pp. 21–39.
40. Perazzo, P.; Righetti, F.; Manna, M.L.; Vallati, C. Performance evaluation of Attribute-Based Encryption on constrained IoT devices. *Comput. Commun.* **2021**, *170*, 151–163.
41. Ameer, S.; Benson, J.; Sandhu, R. Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 4032–4051.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.