



Privacy-preserving revocable access control for LLM-driven electrical distributed systems

Peng Xiao¹ · Shunkun Yang² · Hailin Wang¹ · Zhenhong Zhang¹ · Chunsheng Zou³

Received: 11 February 2025 / Accepted: 24 March 2025 / Published online: 15 April 2025
© The Author(s) 2025

Abstract

Large Language Models (LLMs) have become transformative tools in natural language processing, significantly advancing the fields of communication, information analysis, and knowledge sharing. However, the vast amounts of sensitive data they handle pose significant challenges for data security and privacy. Traditional cryptographic methods face limitations in efficiently managing access control in LLM-driven electrical distributed systems. This paper introduces a novel Privacy-preserving Revocable Access Control for LLM-driven electrical distributed systems, addressing key concerns such as access policy concealment, user revocation efficiency, and computational overhead. Leveraging an inner-product-based access control mechanism, the proposed scheme achieves complete access policy concealment while supporting flexible access control with wildcard capabilities. Additionally, it facilitates efficient user revocation without requiring costly updates to ciphertexts in such an LLM-driven electrical distributed system. The concise algorithmic structure ensures high efficiency, further enhanced through online/offline encryption and outsourced decryption mechanisms.

Keywords Fine-grained access control · Attribute-based encryption · Large language model (LLM) · Policy hiding · Direct revocation

1 Introduction

Large Language Models (LLMs) [1] are distinguished by their exceptional capability to perform natural language processing tasks. These models demonstrate impressive skills in comprehending and generating human-like text with high precision and fluency, facilitating numerous applications across various domains. LLM-driven electrical distributed systems can seamlessly translate between languages, summarize lengthy documents, and answer complex questions by synthesizing information from diverse sources. Their adaptability to different domains and writing styles makes them versatile tools for content generation, automated writing assistance, and personalized recommendations. In industries

such as customer service, LLM-driven electrical distributed systems streamline operations by automating responses to common queries, increasing efficiency and productivity. Additionally, LLM-driven electrical distributed systems play a significant role in research and education, providing access to vast amounts of information and generating insights from large datasets. With their ability to understand context and infer meanings, LLM-driven electrical distributed systems have the potential to revolutionize communication, information processing [3], and knowledge dissemination.

Data security [4] in LLMs presents critical challenges due to the vast amounts of sensitive information they handle. Firstly, LLMs are typically trained on extensive datasets, potentially containing personal or proprietary data. Protecting this data during training, storage, and usage is essential to prevent unauthorized access or breaches. Additionally, the text generated by LLMs may inadvertently reveal sensitive information present in the training data, posing privacy risks [5]. Robust encryption, access controls, and anonymization techniques are crucial to mitigate these risks and safeguard data integrity. Access control is equally vital in LLMs to regulate who can interact with the model and access its capabilities. Implementing granular access controls ensures that

✉ Shunkun Yang
yangshunkun@uestc.edu.cn

¹ Information Center, China Southern Yunnan Power Grid Co. Ltd, Kunming 650217, China

² Kash Institute of Electronics and Information Industry, Kash 844000, China

³ Chinaunicom, Chongqing 400010, China

only authorized users or systems can query the model, reducing the risk of misuse or exploitation. Implementing robust access control mechanisms allows organizations to safeguard the confidentiality, integrity, and availability of their data while harnessing the transformative capabilities of LLMs across diverse applications [2, 6].

Effective access control is essential in LLMs to ensure that only authorized users or systems are permitted to access and utilize the model's resources. Traditional cryptographic methods like public-key encryption [7] and identity-based encryption [8] have limitations in efficiently managing access to LLMs due to their rigid nature. However, attribute-based encryption (ABE) [18] showcases a more adaptable and scalable method for implementing access control in LLMs, enabling fine-grained control over data access. ABE allows access policies to be defined based on attributes rather than specific identities or keys. This means that access to LLMs can be granted based on a set of predetermined attributes that users or systems possess, such as role, department, or clearance level. By utilizing ABE, access to data is restricted to users or systems that possess the corresponding attributes, providing a fine-grained and highly controlled method for secure data access. One major advantage of ABE is its ability to enforce complex access policies tailored to the specific needs of organizations. This granularity enables precise specification of who can access what information within LLMs, providing enhanced security and confidentiality. Additionally, ABE reduces the administrative overhead associated with managing access control lists for individual users or entities, making it well-suited for securing LLMs in dynamic and decentralized environments. Overall, ABE provides a powerful solution for achieving fine-grained access control in LLMs, ensuring data security while maintaining flexibility and scalability.

However, existing ABE schemes fail to perfectly align with the application requirements of LLM electrical distributed systems due to the following problems. Firstly, the disclosure of access policies alongside ciphertexts in ABE exposes users' attribute privacy, consequently risking sensitive information leakage. Secondly, in large-scale model applications, frequent changes or revocations of user permissions are common. However, traditional ABE necessitates re-encryption of ciphertexts with new access policies, leading to significant computational overheads. Lastly, the pairing and exponentiation operations involved in ABE's encryption and decryption algorithms impose heavy computational burdens, posing challenges to real-time responsiveness and local user access in large-scale model services.

In response to the identified challenges, this paper presents the first privacy-preserving and fine-grained access control scheme specifically designed for LLM-powered electrical distributed systems. The key contributions of this work are as follows:

- Inner-Product-Based Access Control: We propose a novel attribute-based encryption (ABE) framework that employs inner-product-based access control to ensure complete concealment of access policies. Unlike traditional approaches that risk partial policy exposure, our scheme maintains full confidentiality of attribute-based access conditions, preventing unauthorized users from deducing policy details. Additionally, our design incorporates wildcard capabilities, enabling a more adaptable and dynamic access structure. This flexibility facilitates efficient access control management in hierarchical and evolving environments while upholding robust security assurances.
- Efficient Direct User Revocation: Our scheme introduces a direct revocation mechanism that eliminates the need for frequent user key updates, thereby reducing administrative and computational overhead. In contrast to conventional ABE models, which typically require modifications to both user keys and ciphertexts during revocation, our method streamlines the process by allowing minimal ciphertext updates without affecting user keys. This optimization significantly lowers communication and computation costs, making the scheme highly suitable for dynamic and large-scale LLM-driven electrical applications where user access rights change frequently.
- Optimized Algorithmic Efficiency: To enhance performance, our scheme adopts an efficient algorithmic structure designed to minimize computational complexity. The encryption process benefits from an online/offline mechanism, which precomputes encryption components in advance, thereby reducing real-time encryption latency. Furthermore, the incorporation of outsourced decryption allows resource-constrained devices to delegate heavy computational tasks to external servers. By ensuring that only partial decryption is performed on the server side, user privacy is preserved while significantly enhancing decryption efficiency. These optimizations collectively improve system responsiveness, making the scheme practical for real-world applications that demand low latency and high computational efficiency.

2 Related works

Privacy-preserving LLM In recent years, there has been many research on data security and privacy issues in LLM. Raeini [14] explores the utilization of mathematical structures such as polynomial and vector spaces, alongside privacy-preserving delegation techniques for polynomial and matrix-vector functions. These methods aim to transform a Large Language Model (LLM) into a computational model that ensures privacy preservation. Li et al. [15] present

the privacy-preserving prompt tuning (RAPT) framework, designed to offer privacy assurances for LLM services. RAPT adopts a local privacy setting, enabling users to privatize their data through local differential privacy mechanisms. Moreover, they introduce an innovative task called privatized token reconstruction, which is trained alongside the downstream task, enhancing the LLMs' ability to learn task-specific representations more efficiently. To mitigate privacy risks like data leakage and unauthorized data collection in LLMs, Tong et al. [16] propose InferDPT, the first practical framework for privacy-preserving inference in black-box LLMs. InferDPT incorporates differential privacy into text generation, offering a robust solution to address privacy concerns during LLM inference. Ullah et al. [17] introduced PrivChatGPT, a conceptual model designed to protect privacy in LLMs. PrivChatGPT features two key components: one for safeguarding user privacy during data curation and preprocessing, and another for maintaining private context during the training of large-scale datasets. The authors employ methods such as differential privacy and reinforcement learning (RL) to train LLMs with a strong emphasis on user privacy.

Fine-granularity in Access control Fine-grained access control is often achieved through attribute-based encryption (ABE), the concept first introduced by Bethencourt et al. [18], which evolved from fuzzy identity-based encryption. ABE offers more flexible, privacy-preserving, and one-to-many access control mechanisms. Waters et al. [19] proposed an "expressive" ABE scheme, utilizing linear secret sharing schemes (LSSS) [20] to define access structures. This approach supports a range of access structures, including AND gates and threshold functions. To improve efficiency and practicality, Hohenberger et al. [21] and Miao et al. [22] presented ABE schemes featuring online/offline encryption and outsourced decryption, reducing the computational and storage overhead on both the encryption and decryption sides. For enhanced user privacy, Xiong et al. [23] developed a partially hidden ABE scheme, which separates attributes into names and values, revealing only the attribute values during access control while keeping the names concealed. This scheme also includes features like outsourced decryption and user revocation. To further strengthen privacy, Phuong et al. [13] proposed a fully policy-hidden ABE scheme, which employs a vector inner product for access control. However, its low computational and storage efficiency limits its practical use. In a similar effort, Sun et al. [10] introduced an attribute-based access control system with full policy hiding. Their scheme optimizes the generation of access and attribute vectors to improve computational and communication efficiency, and leverages online/offline encryption to enhance overall encryption performance.

Direct and indirect user revocation The user revocation mechanism can be classified into two types: direct revocation (DR) and indirect revocation (IDR). In DR, data owners determine which data users' access privileges are revoked. In IDR, this revocation is executed by a third-party entity. Cui et al. [12] proposed an indirect revocation ABE scheme where user keys are periodically updated with the involvement of a third-party entity, and ciphertexts also require periodic updates. Although adopting outsourced computation significantly reduces decryption overhead, this scheme still incurs substantial computational and storage costs for key updates. Building upon this indirect revocation mechanism, Bao et al. [11] introduced a policy-hidden ABE scheme, while it alleviates key update overhead on the user side, update operations persist. Liu et al. [25] utilize a direct revocation method by embedding the revocation list within the ciphertext. To manage the issue of an ever-growing revocation list over time, they introduce a secret key time validation technique. This approach enables user keys to expire on a specified date, ensuring that the revocation list contains only the keys revoked before their expiration, thus improving efficiency. Building on this, Zhao et al. [26] proposed a directly revocable ABE scheme with outsourced decryption, significantly reducing the decryption burden.

3 Preliminaries and definitions

3.1 Security assumption

Let \mathbb{G}_0 and \mathbb{G}_1 be two multiplicative cyclic groups of prime order p , and let g be a generator of \mathbb{G}_1 . Consider three random values a, b, c drawn from \mathbb{Z}_p . Given the tuple $(g, g^m, g^n, g^f, \mathcal{Z}) \in \mathbb{G}_0^4 \times \mathbb{G}_1$, the decisional Bilinear Diffie-Hellman (DBDH) problem involves determining whether $\mathcal{Z} = e(g, g)^{mnf}$ or \mathcal{Z} is a random element in \mathbb{G}_1 .

Definition 1 The DBDH assumption holds if no probabilistic polynomial time (PPT) adversary can solve the DBDH problem with non-negligible advantage.

3.2 System model and threat model

A system model for the electrical LLM access control system is abstracted as depicted in Fig. 1. The system comprises four types of entities: trusted authority (TA), LLM server, LLM application proxy, and LLM visitor. The TA is responsible for managing the entire system and issuing keys to LLM visitors. It is a completely trusted entity, immune to compromise and malicious behavior. The LLM server stores data and provides response or encryption data services, serving as the core of

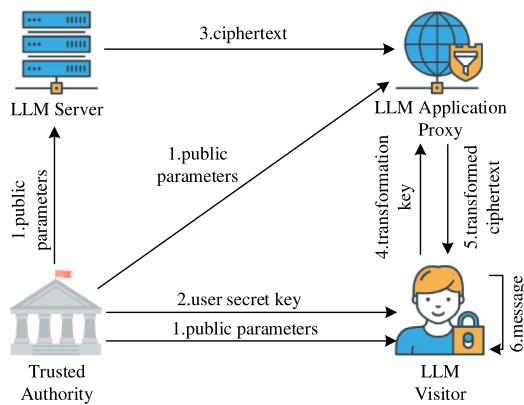


Fig. 1 System model

the large-scale model service. It is also considered a completely trusted entity. The LLM visitor enjoys data services from the LLM server and decrypts ciphertexts to recover plaintext data. It is an untrusted entity, meaning it may illegally access data and reveal privacy information related to ciphertexts. The LLM application proxy is deployed between the LLM server and LLM visitors, providing operations such as data caching and auxiliary computations for LLM visitors. It is an honest-but-curious entity, meaning it can faithfully execute programs but may be interested in some sensitive information.

Our security model is built upon the standard security notions of Attribute-Based Encryption (ABE), specifically: (1) Indistinguishability Under Chosen-Plaintext Attack (IND-CPA): Ensuring that an adversary with access to multiple ciphertexts cannot distinguish between encrypted messages. (2) Collusion Resistance: Preventing unauthorized users from combining their attributes to decrypt data they should not access. (3) Revocation Security: Ensuring that revoked users can no longer access previously accessible data, even if they retain old decryption keys. To enhance security in the LLM access control context, we introduce a dynamic revocation mechanism with efficient key updates, ensuring that access rights are immediately restricted when a user loses authorization. Additionally, our model incorporates fine-grained access policies that enforce role-based, time-sensitive, and hierarchical access control, crucial for LLM training, inference, and API usage restrictions.

3.3 Access control with wildcard

The system's attribute universe is defined as $A = \{A_1, A_2, \dots, A_L\}$, where each attribute A_k for $k \in \{1, \dots, L\}$ can take on one of two values: a negative value “-” or a positive value “+”. Each user in the system possesses a corresponding set of attributes $U = \{U_1, \dots, U_L\}$, where each attribute U_k is either “+” or “-”, for $k \in \{1, \dots, L\}$. Furthermore, an access policy is denoted by $\mathbb{W} = \{W'_1, \dots, W'_L\}$, where each W'_k

is drawn from the set {“+”, “-”, “*”}. The wildcard symbol “*” signifies that the specific value of the corresponding attribute, whether “+” or “-”, is irrelevant.

Table 1 offers an illustrative example. Consider the access structure $A = \{A_1 = "CS", A_2 = "SE", A_3 = "Faculty", A_4 = "Student"\}$, where “CS” and “SE” represent the computer science and software engineering departments, respectively. In this scenario, User1 is a faculty member in the CS department, User2 is a student in the SE department, and User3 is a faculty member associated with both the CS and SE departments. An access policy \mathbb{W}_1 could be satisfied by faculty members from the SE department who are not affiliated with CS. Alternatively, an access policy \mathbb{W}_2 might be satisfied by all faculty and students in the CS department, excluding those who are also part of SE.

3.4 Attributes and access policy vectorization

The privacy-preserving attribute-based access control in our proposed scheme is achieved through the inner product between an access vector and an attribute vector. To vectorize the attribute set and the access policy, Phuong *et al.* proposed a transformation algorithm. However, Sun et al. [10] highlighted the inefficiency of this algorithm and proposed an enhanced vectorization approach to optimize efficiency and reduce vector length. This optimized algorithm demonstrates superior communication and computational capabilities and has been widely employed in pertinent research endeavors, as elucidated in Algorithm 1.

3.5 Syntax

The proposed scheme consists of the following eight algorithms: **Setup**, **KeyGen**, **Off.Enc**, **On.Enc**, **Authorization**, **Pro.Dec**, **Vis.Dec**, and **CipherUpdate**, defined as follows:

- **Setup** (κ) $\rightarrow (MPK, MSK)$: Generates public parameters MPK and master secret key MSK using security parameter κ .
- **KeyGen** (MSK, MPK, S, ID) $\rightarrow SK$: Produces user secret key SK based on MSK , MPK , attribute set S , and identifier ID .

Table 1 Attribute-based access control with wildcard

Attribute set	A_1	A_2	A_3	A_4
Description	CS	SE	Faculty	Student
User1	+	-	+	-
User2	-	+	-	+
User3	+	+	+	-
\mathbb{W}_1	-	+	+	-
\mathbb{W}_2	+	-	*	*

Algorithm 1 Vectorization of Attributes and Access Policy

Input: An access structure that supports at most ℓ wildcards (“*”), ℓ_+ positive attributes (“+”) and ℓ_- negative attributes (“-”). Let the attribute set be $U = \{U_1, \dots, U_L\}$, where each attribute $U_i \in \{+, -\}$ for $i \in \{1, \dots, L\}$.

Output: An access vector and an attribute vector.

- 1: Separate the positions of positive attributes and wildcards in the access policy into two sets: J (for positive attributes) and I (for wildcards).
- 2: **for** each $k_w \in I$ **do**
- 3: Solve the polynomial equation $\prod_{k_w \in I} (i - k_w) = \sum_{j=0}^n a_j i^j$ to obtain the coefficients a_j .
- 4: **end for**
- 5: **for** each $k_w \in I$ and $i \in J$ **do**
- 6: Compute $\prod_{i \in J} \prod_{k_w \in I} (i - k_w)$.
- 7: **end for**
- 8: Extract only the positive attributes into a new set J' .
- 9: **for** $i = 1$ to ℓ , where $i \in J'$ **do**
- 10: Compute $u_j = \sum_{i \in J'} i^j$.
- 11: **end for**
- 12: Construct the attribute vector $\vec{u}_{J'} = (u_0, u_1, \dots, u_\ell)$ and the access vector $\vec{v} = (a_0, a_1, \dots, a_n, 0_{n+1}, \dots, 0_\ell)$.

- **Off.Enc** (MPK) \rightarrow ICT : Outputs intermediate ciphertext ICT from MPK .
- **On.Enc** (MPK, ICT, W, Msg, RL) \rightarrow CT : Derives ciphertext CT from MPK, ICT , access policy W , message Msg , and revocation list RL .
- **Authorization** (SK, MPK) \rightarrow TK, DK : Generates transformation key TK and delegation key DK from SK and MPK .
- **Pro.Dec** (CT, TK) \rightarrow \perp / TC : Returns transformed ciphertext TC or \perp using CT and TK .
- **Vis.Dec** (TC, DK) \rightarrow Msg : Decrypts TC with DK and outputs the message Msg .
- **CipherUpdate** (CT) \rightarrow CT : Refreshes and returns updated ciphertext CT .

Definition 2 The proposed scheme is correct, if $\forall \kappa \in \mathbb{N}$, and S satisfying W , then there exists:

$$\Pr[\text{Vis.Dec}(TC, DK) = Msg] = 1$$

where **Setup**, **KeyGen**, **Off.Enc**, **On.Enc**, **Authorization**, **Pro.Dec** are honestly executed.

3.6 Security model

The selective indistinguishability of the proposed scheme under a chosen-plaintext attack is defined through an interactive game $\text{Game}_{\text{IND-CPA}}$ played between a challenger \mathcal{A} and a probabilistic polynomial time (PPT) adversary \mathcal{A} . The steps are outlined as follows:

- **Initialization:** The adversary \mathcal{A} selects a target access policy W^* and revocation list RL^* and forwards them to the challenger \mathcal{C} .
- **Setup:** The challenger \mathcal{C} runs the **Setup** procedure to generate public parameters MPK , which are then provided to \mathcal{A} .
- **Phase 1 & Phase 2:** The adversary \mathcal{A} makes queries for secret keys corresponding to an identity ID and attribute set S . The challenger \mathcal{C} executes **KeyGen** and sends the resulting secret key SK to \mathcal{A} .
- **Challenge:** The adversary \mathcal{A} submits two messages of equal length, Msg_0 and Msg_1 . The challenger \mathcal{C} selects $b \in \{0, 1\}$ at random and encrypts Msg_b using **Off.Enc** and **On.Enc** to obtain ciphertext CT_b , which is returned to \mathcal{A} .
- **Guess:** The adversary \mathcal{A} attempts to guess the value of b by outputting b' . If $b' = b$, \mathcal{A} wins $\text{Game}_{\text{IND-CPA}}$.

Definition 3 (IND-CPA): If adversary \mathcal{A} wins $\text{Game}_{\text{IND-CPA}}$ with a non-negligible advantage ε , then a PPT algorithm \mathcal{C} can be constructed to break the IND-CPA security of the proposed scheme with non-negligible probability ε' .

4 Concrete construction

The proposed solution is derived from Sun *et al.*'s ABE scheme with inner-product access control [9]. Renowned for its features of lightweight and privacy-preserving, this scheme has been widely referenced and discussed in subsequent research [11]. Building upon that, our proposed scheme introduces an efficient user-level direct revocation mechanism and further reduces computational and storage overhead through algorithm optimization. This scheme encompasses eight algorithms, delineated as follows:

- **Setup** (κ) \rightarrow (MPK, MSK): The trusted authority (TA) initializes the system by taking the security parameter κ and selecting two multiplicative cyclic groups \mathbb{G}_1 and \mathbb{G}_2 of prime order p , where g is a generator of \mathbb{G}_1 . A bilinear pairing $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ is established, and a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$ is defined. The TA randomly chooses $\tau_1, \dots, \tau_n, \alpha, \beta \in \mathbb{Z}_p$, and computes $g_i = g^{\tau_i}$ and $U = e(g, g)^\alpha$, for $i \in [1, n]$. Next, the TA selects a random vector $\vec{\alpha} = (\alpha_1, \dots, \alpha_R)^T \in \mathbb{Z}_p^R$, where R is the number of users, and computes $\mathbf{F} = (f_1, \dots, f_R)^T = (g^{\alpha_1}, \dots, g^{\alpha_R})^T$. The public parameters are then published as $MPK = \{\mathbb{G}_1, \mathbb{G}_2, p, g, e, \mathbf{F}, \{g_i\}_{i \in [1, n]}, U\}$, while the master secret key $MSK = (\tau_1, \dots, \tau_n, \alpha)$ is kept confidential.
- **KeyGen** (MSK, MPK, S, ID) \rightarrow SK : The trusted authority (TA) takes the master secret key MSK , the

- user's attribute set $S \subseteq U$, and their identifier ID as input. First, the attribute vector $\vec{u} = (u_1, \dots, u_n)$ is generated by running **Algorithm 1** with input S . The TA then selects random values $s, s' \in \mathbb{Z}_p$ and calculates $sk_1 = g^{\alpha+s \sum_{i=1}^n \tau_i u_i}$ and $sk_2 = g^s$. Additionally, for each $j = 2, \dots, R$, it computes $F_j = (f_1^{-ID^{j-1}} \cdot f_j)^t$. The resulting user secret key is $SK = \{\vec{u}, sk_1, sk_2, \{F_j\}_{j=2, \dots, R}\}$.
- **Off.Enc** (MPK) \rightarrow ICT : The LLM server, using the public parameters MPK , randomly selects $t, \theta, \eta_1, \dots, \eta_n \in \mathbb{Z}_p$. It then computes the values $c_0 = g^t$, $c_1 = g_1^t g^{-\eta_1}$, and continues similarly for all $c_i = g_i^t g^{-\eta_i}$, where $i \in [1, n]$. The resulting intermediate ciphertext is recorded as $ICT = (c_0, \{c_i\}_{i \in [1, n]}, t, \theta, \eta_1, \dots, \eta_n)$.
 - **On.Enc** (MPK, ICT, W, Msg, RL) \rightarrow CT : This algorithm takes the public parameters MPK , the access structure W , the revocation list $RL = (ID_1, \dots, ID_\mu)$ as input, where μ denotes the number of visitors to be revoked, $\mu < R$. This algorithm generates the access vector $\vec{v} = (v_1, \dots, v_n)$ by running **Algorithm 1** and inputting W . The polynomial for the revocation list $RL = (ID_1, \dots, ID_\mu)$ is constructed as $F_{RL}(x) = (x - ID_1)(x - ID_2) \cdots (x - ID_\mu)$, which expands to $F_{RL}(x) = x_1 + x_2 x + \cdots + x_\mu x^{\mu-1} + x_\mu x^{\mu-1}$. Furthermore, LLM server calculates $\gamma_i = \eta_i + v_i \theta$, $c = Msg \cdot U^t$, $c'' = (f_1^{x_1} \cdots f_R^{x_R})^t$. Finally, LLM server returns $CT = (c, c', c'', c_0, \{c_i, \gamma_i\}_{i \in [1, n]})$.
 - **Authorization** (SK, MPK) \rightarrow TK, DK : The LLM visitor takes the secret key SK and public parameters MPK as inputs, selects $z \in \mathbb{Z}_p$ at random, and computes the following: $sk'_1 = sk_1^z = g^{z(\alpha+s \sum_{i=1}^n \tau_i u_i)}$, $sk'_2 = sk_2^z = g^{zs}$, $F'_j = F_j^z = (f_1^{-ID^{j-1}} \cdot f_j)^{utz}$, and $\vec{u}' = z \cdot \vec{u} = (zu_1, \dots, zu_n)$. The output is the transformation key $TK = \{\vec{u}', sk'_1, sk'_2, \{F'_j\}_{j \in [2, \dots, R]}\}$ and the delegation key $DK = z$.
 - **Pro.Dec** (CT, TK) \rightarrow \perp / TC : The LLM application proxy processes the ciphertext CT and transformation key TK . It creates two vectors of the same length: $\mathbf{X} = (x_1, \dots, x_R)$ and $\mathbf{Y} = (1, ID, \dots, ID^{R-1})$. The polynomial $F_{RL}(x)$, evaluated at ID , is equivalent to the inner product of \mathbf{X} and \mathbf{Y} , written as $F_{RL}(ID) = x_1 + x_2 ID + \cdots + x_u ID^{u-1} + x_{u+1} ID^u = \langle \mathbf{X}, \mathbf{Y} \rangle$. If $u+1 < R$, the remaining coefficients x_{u+2}, \dots, x_R are set to zero.
- If the LLM visitor's attribute set S does not satisfy the access policy or if $ID \in RL$ (i.e., $\langle \mathbf{X}, \mathbf{Y} \rangle = F_{RL}(ID) = 0$), the algorithm outputs \perp . Otherwise, the LLM application proxy computes the transformed ciphertext TC as:

$$TC = \frac{e\left(g^{\sum_{i=1}^n \gamma_i z u_i} \prod_{i=1}^n c_i^{z u_i}, sk'_2\right)}{e(c_0, sk'_1)}.$$

The proxy then outputs the resulting transformed ciphertext TC .

- **Vis.Dec** (TC, DK) \rightarrow Msg : The LLM visitor, using the transformed ciphertext TC and delegation key DK , retrieves the message Msg by computing:

$$Msg = c \cdot (TC)^{1/z}.$$

- **CipherUpdate** (CT) \rightarrow CT' : When the revocation list is updated to $RL' = (ID'_1, \dots, ID'_\mu)$, the LLM server updates the ciphertext component c'' by recalculating $F_{RL'}(x) = (x - ID'_1)(x - ID'_2) \cdots (x - ID'_\mu)$, which expands as $x_1 + x_2 x + \cdots + x_\mu x^{\mu-1}$. The updated c'' is then computed as $c'' = (f_1^{x_1} \cdots f_R^{x_R})^t$.

5 Security analysis

Theorem 1 (Correctness): *The proposed scheme is correct.*

Proof After receiving the ciphertext CT from the LLM server, if **Authorization**, **Pro.Dec**, **Vis.Dec** are honestly executed, then the message Msg can be recovered as:

$$\begin{aligned} Msg &= c \cdot (TC)^{1/z} \\ &= c \cdot \left(\frac{e(g^{\sum_{i=1}^n \gamma_i z u_i} \prod_{i=1}^n c_i^{z u_i}, sk'_2)}{e(c_0, sk'_1)} \right)^{1/z} \\ &= c \cdot \left(\frac{e(g^{\sum_{i=1}^n (\eta_i + v_i \theta) u_i} \prod_{i=1}^n g^{\tau_i u_i t} g^{-\eta_i u_i}, g^{zs})}{e(g^t, g^{z(\alpha+s \sum_{i=1}^n \tau_i u_i)})} \right)^{1/z} \\ &= c \cdot \left(\frac{e(g^{\sum_{i=1}^n v_i \theta u_i} \prod_{i=1}^n g^{\tau_i u_i t}, g^{zs})}{e(g^t, g^{z(\alpha+s \sum_{i=1}^n \tau_i u_i)})} \right)^{1/z} \\ &= c \cdot \left(\frac{1}{e(g, g)^{\alpha t}} \right)^{1/z} \\ &= Msg \cdot e(g, g)^{\alpha t} \cdot \frac{1}{e(g, g)^{\alpha t}} \end{aligned}$$

Theorem 2 (IND-CPA security): *Under the DBDH assumption, no PPT adversary \mathcal{A} can break the security of the proposed scheme when presented with a challenge access structure W^* and a revocation list RL^* .*

Proof If an adversary \mathcal{A} can selectively compromise the indistinguishability of the proposed scheme under a chosen plaintext attack with non-negligible advantage ε , a challenger \mathcal{C} can be constructed to solve the DBDH problem with a non-negligible probability ε' . This can be demonstrated through the following game $\text{Game}_{\text{IND-CPA}}$:

- **Initialize.** Given a DBDH input tuple $(g, g^a, g^b, g^c, \mathcal{R})$, the adversary \mathcal{A} defines the challenge access structure W^* and revocation list RL^* . By applying **Algorithm 1**,

the access structure W^* is transformed into a vector $\vec{v} = (v_1, \dots, v_n)$.

- **Setup.** The challenger \mathcal{C} selects random values $\varsigma, \beta'_1, \dots, \beta'_n \in \mathbb{Z}_p$ and simulates the parameters:

$$g_1 = (g^a)^{-\varsigma v_1} g^{\beta'_1}, \dots, g_n = (g^a)^{-\varsigma v_n} g^{\beta'_n}, \quad U = e(g^a, g^b),$$

where $\alpha = ab$ and $\tau_i = -a\varsigma v_i + \beta'_i$ for $i \in [1, n]$.

Let $|RL^*| = l$ and assume $l \leq q - 2$. Define vectors $\mathbf{Y}_1, \dots, \mathbf{Y}_l$ for the revoked list $RL = \{ID_1, \dots, ID_l\}$, with $\mathbf{Y}_j = (1, ID_j, \dots, ID_j^{q-2})$ for $j \in [1, l]$. For each $j \in [1, l]$, define:

$$M_{\mathbf{Y}_j} = \begin{pmatrix} -ID_j & -ID_j^2 & \cdots & -ID_j^{q-2} \\ & I_{q-2} \end{pmatrix},$$

where I_{q-2} is an identity matrix of size $(q-2) \times (q-2)$. The challenger generates a vector $\vec{b}_j \in \mathbb{Z}_p^{q-1}$ such that $\vec{b}_j \cdot M_{\mathbf{Y}_j} = \vec{0}$, with $\vec{b}_j = (1, ID_j, \dots, ID_j^{q-2})$.

For $j \in [r+1, q-1]$, set $\vec{b}_j = \vec{0}$. Construct a $(q-1) \times (q-1)$ matrix $B = (\vec{b}_1 | \cdots | \vec{b}_r | \vec{0} | \cdots | \vec{0})$, where the first r columns are vectors $\{\vec{b}_j\}_{j \in [1, r]}$ and the remaining columns are zero vectors. The challenger defines $\vec{\mu} = (\mu_1, \dots, \mu_{q-1})^T$, with $\mu_i = a^{q+1-i}$, and computes $g^{\vec{\mu}} = (g^{a^q}, \dots, g^{a^2})^T$. A random vector $\vec{\delta} \in \mathbb{Z}_p^{q-1}$ is chosen to set $\vec{\alpha} = B \cdot \vec{\mu} + \vec{\delta}$. Finally, the challenger defines:

$$\mathbf{F} = g^{B \cdot \vec{\mu}} \cdot g^{\vec{\delta}} = g^{\vec{\alpha}} = (g^{\alpha_1}, \dots, g^{\alpha_R})^T = (f_1, \dots, f_R)^T.$$

- **Phase 1.** The adversary \mathcal{A} repeatedly issues secret key queries for (ID, S) to the challenger \mathcal{C} . The challenger verifies if the following conditions are met: (1) the identifier ID is present in the revocation list, i.e., $ID \in RL^*$; (2) the attribute set S does not satisfy the access policy W^* . If either condition holds, the challenger simulates the user secret key as follows:

If $S \notin W^*$, the challenger first derives the attribute vector $\vec{u} = (u_1, \dots, u_n)$ by running **Algorithm 1** with input S , ensuring $\langle \vec{u}, \vec{v} \rangle \neq 0$. The challenger then selects a random value $\epsilon'_1 \in \mathbb{Z}_p$ and computes:

$$sk_1 = \prod_{i=1}^n ((g^a)^{-\varsigma u_i} g^{\beta'_i})^{v_i \epsilon'_1} \cdot (g^b)^{\frac{\beta'_i v_i}{\varsigma \langle \vec{u}, \vec{v} \rangle}},$$

$$sk_2 = g^{\epsilon'_1} (g^b)^{\frac{1}{\varsigma \langle \vec{u}, \vec{v} \rangle}}$$

The challenger \mathcal{C} then generates and returns the user secret key $SK = (sk_1, sk_2, u_1, \dots, u_n)$. To clarify the derivation

of SK , let $s = \epsilon' 1 + \frac{b}{\varsigma \langle \vec{u}, \vec{v} \rangle}$. The key sk_1 can be rewritten as:

$$\begin{aligned} sk_1 &= \prod_{i=1}^n ((g^a)^{-\varsigma v_i} g^{\beta'_i})^{u_i \epsilon'_1} \cdot (g^b)^{\frac{\beta'_i u_i}{\varsigma \langle \vec{u}, \vec{v} \rangle}} \\ &= \prod_{i=1}^n g^{-au_i v_i \varsigma \epsilon'_1} g^{-ab \varsigma u_i v_i \frac{1}{\varsigma \langle \vec{u}, \vec{v} \rangle}} g^{ab \varsigma u_i v_i \frac{1}{\varsigma \langle \vec{u}, \vec{v} \rangle}} \\ &\quad g^{\beta'_i v_i \epsilon'_1} g^{\frac{b \beta'_i u_i}{\varsigma \langle \vec{u}, \vec{v} \rangle}} \\ &= \prod_{i=1}^n (g^{-a \varsigma v_i})^{u_i (\epsilon'_1 + \frac{b}{\varsigma \langle \vec{u}, \vec{v} \rangle})} (g^{\beta'_i})^{u_i (\epsilon'_1 + \frac{b}{\varsigma \langle \vec{u}, \vec{v} \rangle})} \\ &= g^{ab \varsigma u_i v_i \frac{1}{\varsigma \langle \vec{u}, \vec{v} \rangle}} \\ &= g^{ab} \prod_{i=1}^n (g_i)^{u_i s} \end{aligned}$$

$$\text{where } g^{ab \varsigma u_i v_i \frac{1}{\varsigma \langle \vec{u}, \vec{v} \rangle}} = g^{ab} = g^\alpha.$$

If $ID \in RL^*$, where $ID_j \in RL^*$ and $j \in [1, \mu]$ is queried by the adversary, the challenger calculates the first coordinate of $\vec{\alpha}$:

$$\alpha_1 = \alpha' + \sum_{j=1}^{\mu} u_j = \alpha' + \sum_{j=1}^{\mu} a^{q+1-j}$$

The challenger retrieves $\vec{\mu} = (\mu_1, \dots, \mu_{q-1})^T$ from $g^{\vec{\mu}} = (g^{\mu_1}, \dots, g^{\mu_{q-1}})^T = (g^{a^q}, \dots, g^{a^2})^T$. On this basis, \mathcal{C} simulates:

$$\begin{aligned} F'_j &= g^{t M_{\mathbf{Y}_j}^T \vec{\alpha}} \cdot g^{-\sum_{i \in [1, n]} w_k a^{q+1-k} M_{\mathbf{Y}_j}^T B \vec{\mu}} \cdot g^{-\sum_{i \in [1, n]} w_k a^{q+1-k} M_{\mathbf{Y}_j}^T B \alpha'} \\ &= g^{t M_{\mathbf{Y}_j}^T \vec{\alpha}} \cdot g^{-\sum_{i \in [1, n]} w_k a^{q+1-k} M_{\mathbf{Y}_j}^T B \vec{\alpha}} \\ &= g^{(t - \sum_{i \in [1, n]} w_k a^{q+1-k}) M_{\mathbf{Y}_j}^T \vec{\alpha}} \\ &= g^{t' M_{\mathbf{Y}_j}^T \vec{\alpha}} \end{aligned}$$

For $j \in 1, \dots, R$, the challenger then computes:

$$F_j = g^{t' M_{\mathbf{Y}_j}^T \vec{\alpha}} = g^{t' (-ID_k^{i-1} \alpha_1 + \alpha_k)} = (f_1^{-ID_k^{i-1}} \cdot f_j)^{t'}$$

Finally, the challenger \mathcal{C} returns the secret key $SK = \vec{u}, sk_1, sk_2, F_j$ to \mathcal{A} .

- **Challenge.** The adversary \mathcal{A} submits two messages of equal length, Msg_0 and Msg_1 , to the challenger \mathcal{C} . The challenger responds with the challenge ciphertext: Let $t = c, \theta = ac\varsigma$, then:

where $c_{1,i}^* = (g^c)^{\beta'_i} = (g^c)^{a\varsigma v_i - a\varsigma v_i + \beta'_i} = (g^c)^{a\varsigma v_i}$ ($(g^c)^{-a\varsigma v_i + \beta'_i} = g^{v_i\theta}(g_i)^t$). Additionally, the challenger randomly selects $\eta'_1, \dots, \eta'_n \in \mathbb{Z}_p$ and calculates $c1, i^*$. $g^{-\eta'_i} = g^{v_i\theta - \eta'_i}(g_i)^t = g_i^t g^{-\eta_i} = c1, i$, where $\eta_i = -v_i\theta + \eta'_i$. Furthermore, $\eta'_i = v_i\theta + \eta_i = \gamma_i$.

For the ciphertext component C'' , let $RL^* = (ID_1, \dots, ID_u)$ and $F_{RL^*}(x) = (x - ID_1)(x - ID_2) \cdots (x - ID_u) = x_1 + x_2x + \cdots + x_u x^{u-1} + x_{u+1} x^u$. For $u + 1 < R$, set $x_{u+2}, \dots, x_R = 0$. Define $\mathbf{X} = (x_1, \dots, x_R)^T$, with $\langle \mathbf{X}, \mathbf{Y}_j \rangle = 0$ for $j \in [1, \mu]$, and $\mathbf{X}^T \cdot B \cdot \vec{u} = 0$. The challenger then simulates:

$$c'' = \left(\prod_{j=1}^R f_j^{x_j} \right)^s = (g^s)^{\langle \mathbf{X}, \vec{\alpha}' \rangle}$$

Finally, the challenger returns the ciphertext $CT_b = (c = \text{Msg}_b \cdot \mathcal{R}, c_0, c_{i,i \in [1,n]})$ to the adversary \mathcal{A} .

6 Performance evaluations

In this section, we provide an in-depth comparison of our proposed scheme with several prominent existing approaches in terms of both functionality and performance. Specifically, we compare our scheme with Waters' expressive CP-ABE scheme [19], Sun et al.'s CP-ABE with keyword search (CP-ABKS) [10], and Cui et al.'s server-assisted and revocable ABE (SR-ABE) [12].

Table 2 provides a detailed comparison of the functionalities of our proposed scheme with other prominent works, including CP-ABE [19], CP-ABKS [10], and SR-ABE [12]. Symbols such as “✓” and “✗” represent supported and unsupported features, respectively, while “○” indicates non-applicability. Both CP-ABE [19] and CP-ABKS [10] lack support for user revocation, whereas SR-ABE [12] and our scheme provide non-direct and direct revocation mechanisms, respectively. CP-ABE [19] and SR-ABE [12] are built using Linear Secret Sharing Schemes (LSSS), which enable more expressive access control than the AND-Gate structure used in CP-ABKS [10] and our scheme. In terms of revoca-

tion, SR-ABE [12] requires updates to both keys and ciphertext, while our scheme only requires ciphertext updates. Both CP-ABKS [10] and our scheme employs offline/online encryption mechanisms to achieve rapid encryption, making them well-suited for high-concurrency scenarios. Regarding decryption, SR-ABE [12] and our scheme allow for delegable decryption, making them advantageous for devices with limited computational power. Additionally, our scheme supports access policy concealment and wildcard use, offering enhanced privacy and flexibility.

In this section, we present a comprehensive comparison of our proposed scheme with several established approaches, focusing on functionality and performance. Specifically, we compare our scheme with Waters' expressive CP-ABE scheme [19], Sun et al.'s CP-ABE with keyword search (CP-ABKS) [10], and Cui et al.'s server-assisted and revocable ABE (SR-ABE) [12]. The reasons why we choose these two schemes for comparisons are as follows: CP-ABE was excluded from the experimental evaluation because it lacks critical functionalities required for our targeted scenario, particularly revocation and searchable encryption capabilities. While CP-ABE serves as a foundational scheme, it does not support efficient user revocation, making it unsuitable for dynamic environments like LLM access control. Additionally, CP-ABE does not inherently provide keyword searchability, a key feature in CP-ABKS and SR-ABE, which are more aligned with our scheme's design. Thus, our experimental comparisons focused on CP-ABKS and SR-ABE, as they better represent state-of-the-art schemes with functionalities similar to ours.

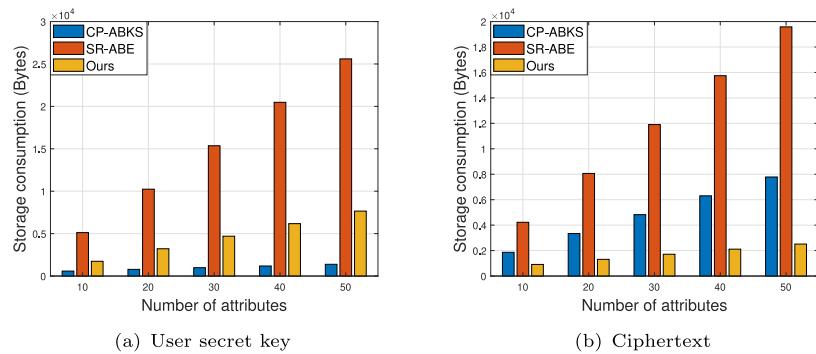
We conducted practical performance evaluations of the selected schemes on a laptop equipped with an AMD Ryzen 5 4600H 3.00 GHz processor, 16GB RAM, and a 64-bit Windows 10 operating system. The primary aim was to compare the efficiency of our proposed scheme against CP-ABKS [10] and SR-ABE [12]. The implementations utilized the PBC library [24], with algebraic structures based on a supersingular curve $E/\mathbb{F}_p : y^2 = x^3 + x$ over the finite field \mathbb{F}_p , with an embedding degree of 2, where p represents the prime order of group \mathbb{G}_1 . Consequently, we have $|\mathbb{G}_1| = |\mathbb{G}_2| = 128$ bytes and $|\mathbb{Z}_p^*| = 20$ bytes.

We simulated an LLM server encrypting a message using our proposed scheme, which was then distributed to LLM

Table 2 Functional comparisons

Schemes	Revocability	Access structure	Key update	Ciphertext update	Immediate encryption	Delegable computation	Policy hidden	Wildcard
CP-ABE [19]	✗	LSSS	○	○	✗	✗	✗	✗
CP-ABKS [10]	✗	AND-Gate	○	○	✓	✗	✓	✓
SR-ABE [12]	indirect	LSSS	✓	✓	✗	✓	✗	✗
Ours	direct	AND-Gate	✗	✓	✓	✓	✓	✓

Fig. 2 Comparisons of storage costs



visitors. To evaluate practicality, we adjusted the number of attributes from 1 to 50 in increments of 10, and recorded the storage overhead for keys and ciphertexts, as depicted in Fig. 2. Figure 2-(a) shows that while our scheme has slightly higher key storage requirements compared to CP-ABKS [10], it outperforms SR-ABE [12]. This difference arises due to the user revocation component in our scheme, which scales with the number of attributes, unlike CP-ABKS [10], which lacks user revocation functionality. Additionally, SR-ABE [12] features a more complex key structure than our scheme. Figure 2-(b) illustrates that the proposed scheme requires significantly less ciphertext storage compared to both CP-ABKS [10] and SR-ABE [12]. When the attribute count reaches 50, the size of a single ciphertext in our scheme is only 2512 bytes, thanks to its efficient ciphertext structure.

Figure 3 provides a comparison of the computational overhead across key functional phases-key generation, encryption, and decryption-for the three schemes. The analysis considers the number of attributes, ranging from 1 to 50, with data points recorded at intervals of 5. To ensure fairness, the number of revoked users was fixed at 20. Figure 3-(a) shows that key generation time increases with the number of attributes for all schemes, though their growth rates vary significantly. Our proposed scheme demonstrates lower key generation overhead compared to CP-ABKS [10] and SR-ABE [12], with a time of 719.88 ms for 50 attributes. This

improvement is due to the attribute aggregation mechanism used in our scheme. In Fig. 3-(b), the encryption phase shows that SR-ABE [12] experiences a sharp increase in computational overhead as attributes grow, attributed to its complex ciphertext structure. Conversely, both CP-ABKS [10] and our scheme maintain stable encryption times, outperforming SR-ABE significantly by leveraging online/offline encryption methods. However, our scheme incurs a slight increase in encryption time compared to CP-ABKS, due to the additional step of ciphertext updating, resulting in 291.12 ms for 50 attributes. Figure 3-(c) compares decryption times, illustrating that CP-ABKS [10] exhibits a rapid increase in decryption overhead with more attributes, whereas SR-ABE [12] and our scheme maintain low and consistent times. For 50 attributes, our scheme's decryption remains under 10 ms, benefiting from the outsourced decryption approach, which delegates most computational tasks to the LLM application proxy.

We performed experiments to examine the correlation between user revocation time and the number of revoked users. In the experiment, we set the complete binary tree depth in SR-ABE [12] to 11, allowing for up to $2^{10} = 1024$ users. Likewise, the revocation list size in our proposed scheme was set to 1024. Since SR-ABE requires updates to both keys and ciphertexts during user revocation, it incurs significant computational costs. As illustrated in Fig. 4a, although SR-ABE's revocation overhead decreases with

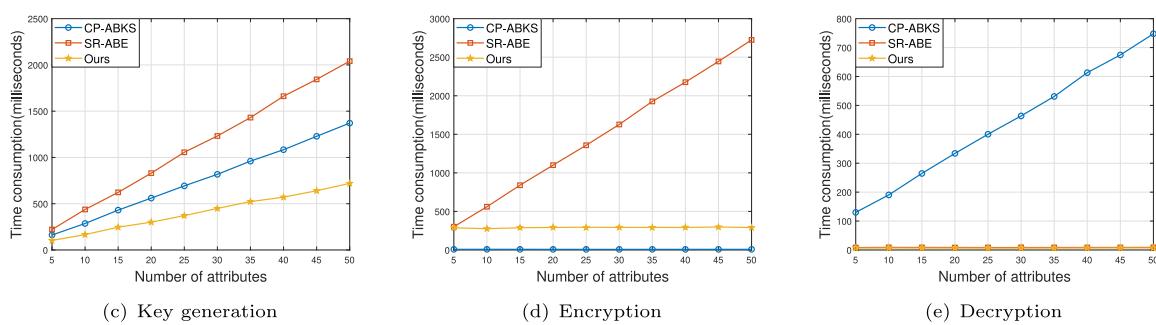
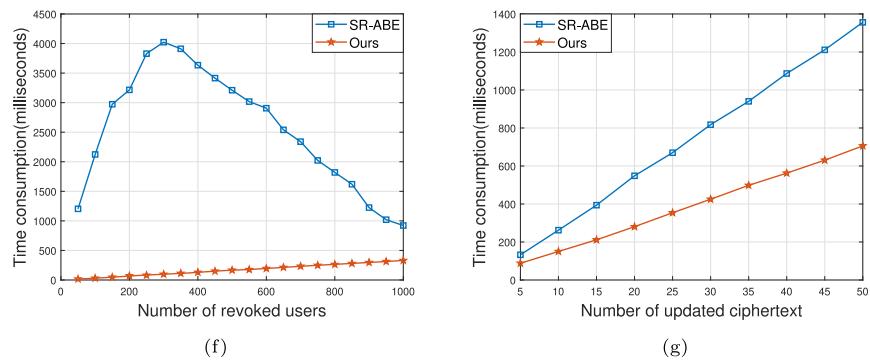


Fig. 3 Comparisons of computational costs

Fig. 4 (a) Comparisons on revocation time costs and (b) Comparisons on ciphertext update time costs



more users being revoked, it remains much higher compared to our scheme. In our approach, revocation only involves updating the ciphertexts based on the revocation list, making it more efficient.

We also compared the computational cost of ciphertext updates in both SR-ABE's non-direct revocation and our scheme's direct revocation mechanism. Figure 4b presents the results, with the number of ciphertexts updated ranging from 5 to 50. Both SR-ABE [12] and our proposed scheme demonstrated nearly linear growth in update overhead, but our scheme consistently exhibited lower overhead. Specifically, updating 50 ciphertexts in our scheme required only 705.84 ms.

7 Conclusion

LLMs have significantly advanced natural language processing, revolutionizing communication, information management, and knowledge dissemination. However, handling vast amounts of sensitive data necessitates robust data security, particularly regarding access control. Traditional cryptographic methods often struggle to manage access effectively in LLM-driven electrical distributed systems, highlighting the need for ABE solutions. This paper presents an ABE scheme for LLM contexts, addressing key challenges including access policy concealment, efficient user revocation, and minimized computational overhead. By leveraging inner-product-based access control and incorporating flexible access via wildcard features, the scheme ensures data security while maintaining operational efficiency. These contributions establish a pathway for implementing privacy-preserving, fine-grained access control in LLMs, supporting secure deployment across diverse applications.

Author Contributions Peng Xiao contributed to the manuscript by drafting Sections 1 and 2 and developing the computational source codes presented in Section 7. Zhenhong Zhang was responsible for writing Sections 3 and 6 and illustrating the system model in Section 3. Hailin Wang authored Section 5 and conducted the time consumption experiments. Chunsheng Zou contributed by drafting Section 4. Shunkun Yang was in charge of writing the Abstract and Conclusion, conducting

the real-world security analysis, refining the security definitions, and reviewing the entire manuscript for consistency and accuracy.

Funding N.A.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

Ethics Approval This study did not involve any ethical issues.

Consent to Publish All authors have read and agreed to the published version of this manuscript.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Chang Y, Wang X, Wang J et al (2023) A survey on evaluation of large language models. ACM Trans Intell Syst Technol pp 1–43. <https://doi.org/10.1145/3641289>
- Xu G, Li G, Guo S et al (2023) Secure decentralized image classification with multiparty homomorphic encryption. IEEE Trans Circ Syst Video Technology
- Min B, Ross H, Sulem E et al (2023) Recent advances in natural language processing via large pre-trained language models: A survey. ACM Comput Surv 56(2):1–40
- Pan X, Zhang M, Ji S, et al (2020) Privacy risks of general-purpose language models. 2020 IEEE Symposium on Security and Privacy (SP). IEEE, pp 1314–1331

5. Kasneci E, Sebler K, Kuchemann S et al (2023) ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences* 103:102274
6. Chen Y, Arunasalam A, Celik Z B (2023) Can large language models provide security & privacy advice? measuring the ability of llms to refute misconceptions. In: Proceedings of the 39th annual computer security applications conference. pp 366–378
7. Li X, Yang G, Xiang T et al (2024) Make Revocation Cheaper: Hardware-Based Revocable Attribute-Based Encryption. In: 2024 IEEE Symposium on security and privacy (SP). IEEE Computer Society pp 10–20
8. Deng H, Qin Z, Wu Q et al (2020) Identity-based encryption transformation for flexible sharing of encrypted data in public cloud. *IEEE Trans Inf Forensics Sec* 15:3168–3180
9. Sun J, Xiong H, Deng RH et al (2019) Lightweight attribute-based keyword search with policy protection for cloud-assisted IoT. In: 2019 IEEE Conference on dependable and secure computing 3rd DSC: Hangzhou, China, pp 1–8
10. Sun J, Xiong H, Liu X, Zhang Y et al (2020) Lightweight and privacy-aware fine-grained access control for IoT-oriented smart health. *IEEE Int Things J* 7(7):6566–6575
11. Bao Y, Qiu W, Cheng X et al (2023) Fine-grained data sharing with enhanced privacy protection and dynamic users group service for the IoV. *IEEE Trans Intell Transport Syst* 24(11):13035–13049
12. Cui H, Deng RH, Li Y et al (2016) Server-aided revocable attribute-based encryption. Computer Security-ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26–30, 2016, Proceedings, Part II 21. Springer International Publishing, LNSC 9879:570–587
13. Phuong TVX, Yang G, Susilo W (2015) Hidden ciphertext policy attribute-based encryption under standard assumptions. *IEEE Trans Inf Forensics Sec* 11(1):35–45
14. Raeini M (2023) Privacy-preserving large language models (PPLMs), Available at SSRN 4512071
15. Li Y, Tan Z, Liu Y (2023) Privacy-preserving prompt tuning for large language model services. [arXiv:2305.06212](https://arxiv.org/abs/2305.06212)
16. Tong M, Chen K, Qi Y et al (2023) Privinfer: Privacy-preserving inference for black-box large language model. [arXiv:2310.12214](https://arxiv.org/abs/2310.12214)
17. Ullah I, Hassan N, Gill SS et al (2023) Privacy preserving large language models: Chatgpt case study based vision and framework. [arXiv:2310.12523](https://arxiv.org/abs/2310.12523)
18. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: 2007 IEEE symposium on security and privacy (SP'07). IEEE, pp 321–334
19. Waters B (2011) Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. International workshop on public key cryptography. Berlin, Heidelberg: Springer, Berlin Heidelberg, LNSC 6571:53–71
20. Benhamouda F, Degwekar A, Ishai Y et al (2021) On the local leakage resilience of linear secret sharing schemes. *J Cryptol* 34:1–65
21. Hohenberger S, Waters B (2014) Online/offline attribute-based encryption. Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26–28, 2014. Proceedings 17. Springer Berlin Heidelberg, LNSC, vol. 8383, pp 293–310
22. Miao Y, Li F, Li X et al (2023) Verifiable Outsourced Attribute-Based Encryption Scheme for Cloud-Assisted Mobile E-health System. *IEEE Trans Dependable Secure Comput* early access, pp 1–18. <https://doi.org/10.1109/TDSC.2023.3292129>
23. Xiong H, Zhao Y, Peng L et al (2019) Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. *Future Generation Comput Syst* 97:453–461
24. Lynn B (2007) Pbc library-pairing-based cryptography. <http://crypto.stanford.edu/pbc/>
25. Liu JK, Yuen TH, Zhang P et al (2018) Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list. *Applied Cryptography and Network Security: 16th International Conference, ACNS 2018, Leuven, Belgium, July 2–4, 2018, Proceedings* 16. Springer International Publishing, LNSC 10892:516–534
26. Zhao Y, Wang Y, Cheng X et al (2021) Rfpap: A revocable fine-grained access control mechanism for autonomous vehicle platoon. *IEEE Trans Intell Transport Syst* 23(7):9668–9679

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Peng Xiao obtained a Bachelor's degree in Software Engineering from Dianchi College, Yunnan University. Currently is serving as the Deputy Manager of the Network Security Management Center of the Information Center of Southern Power Grid Yunnan Power Grid Co., Ltd., a Level 3 leading professional technical expert, with a main research focus on information security assessment technology, including network attack and defense technology, network security management, and enterprise security system construction.



Shunkun Yang is the executive president of Kash Institute of Electronics and Information Industry of China. He received the master's degree from University of Electronic Science and Technology of China in 2013. His research interests include network security, communication anti-interference and artificial intelligence.



Hailin Wang obtained a Master's degree in Software Engineering from Yunnan University in 2017, and currently works at the Information Center of Southern Power Grid Yunnan Power Grid Co., Ltd. His main research areas are network security and machine learning.



Zhenhong Zhang received the M.S. degree in computer technology from Beijing University of Posts and Telecommunications in 2015 and is currently an engineer in the information center of China Southern Power Grid Yunnan Power Grid Co., Ltd. Main research directions include network security, power information system operation and maintenance.



Chunsheng Zou is a Senior Technical Manager at China United Network Communications Group Co., Ltd. (China Unicom). He received his Bachelor's degree from Shanghai Jiao Tong University in 1996. His research focuses on big data, low-code development, network security, and artificial intelligence.