Blockchain-Assisted Privacy-Preserving Retrieval for Sensitive Outsourced Data

Bian Zhu* and Ling Niu

School of Computer Science and Technology, Zhoukou Normal University, Zhoukou, Henan 466000, China
E-mail: 18438035975@163.com
**Corresponding Author*

Received 01 April 2026; Accepted 18 May 2026

Abstract

With the continuous outsourcing of sensitive data to cloud platforms and third-party environments, how to achieve efficient and trustworthy data retrieval while preserving data confidentiality and query privacy has become a critical issue in outsourced data security management. Existing studies mainly focus on secure storage and privacy-preserving retrieval but suffer from limitations in the efficiency of multi-attribute conjunctive queries, the suppression of intermediate information leakage during auxiliary condition verification, and the trustworthy traceability of the retrieval process. To address these issues, this paper proposes a blockchain-assisted privacy-preserving retrieval framework for sensitive outsourced data. The framework adopts a collaborative paradigm of off-chain storage and retrieval together with on-chain commitment and auditing. By combining a frequency-aware primary search term selection strategy with a concealed auxiliary verification mechanism, it enables secure filtering of task-relevant data under multi-attribute conditions. Meanwhile, by recording index commitments, query digests, and result digests on the blockchain, the proposed framework enhances the verifiability and traceability of index states and access

Journal of Web Engineering, Vol. 25_5, 1015–1044.
doi: 10.13052/jwe1540-9589.25510

processes. Security analysis and experimental results demonstrate that the proposed scheme can achieve more efficient construction of auxiliary decision structures, more stable query performance, and better storage overhead while preserving data confidentiality and query privacy.

Keywords: Sensitive data outsourcing, privacy-preserving retrieval, multi-attribute conjunctive query, frequency awareness, blockchain.

1 Introduction

With the continuous development of cloud computing and data outsourcing models, an increasing amount of sensitive data is being stored and managed on third-party platforms [1]. Although the outsourcing paradigm offers significant advantages in terms of storage scalability, resource reuse, and cross-domain sharing, once sensitive data leaves the local trusted environment, issues such as data confidentiality protection, query privacy control, and trustworthy management of the access process become increasingly prominent [2]. In application scenarios such as healthcare, finance, government affairs, and enterprise management, systems are required not only to ensure secure storage of outsourced data but also to support authorized users in efficiently obtaining task-relevant data under privacy constraints [3].

To address these challenges, existing studies have proposed various solutions from the perspectives of secure storage, access control, and searchable encryption [4, 5]. These methods have achieved substantial progress in protecting data confidentiality, hiding query contents, and supporting retrieval over encrypted data. However, they still suffer from three major limitations [6–8]. First, the execution efficiency of multi-attribute conjunctive queries heavily depends on the selection of the primary search condition. An inappropriate choice of the primary search term may significantly enlarge the candidate result set and increase the cost of subsequent verification [9]. Second, during conjunctive query processing, auxiliary condition verification may still reveal intermediate matching relationships between candidate results and additional conditions, thereby introducing extra information leakage risks [10]. Third, even if privacy-preserving retrieval can be achieved, the lack of trustworthy records for index states, query behaviors, and result-return processes still limits subsequent auditing, accountability tracing, and process verification [11, 12].

To overcome the above limitations, this paper develops a privacy-preserving retrieval framework for sensitive outsourced data by integrating

frequency-aware primary search term selection, concealed auxiliary verification, and lightweight on-chain recording. The proposed framework follows a collaborative paradigm of off-chain storage and retrieval together with on-chain commitment and auditing. On the off-chain side, the system performs secure filtering of task-relevant data through multi-attribute conjunctive queries and preferentially selects low-frequency query conditions as the primary search term based on frequency estimation results, thereby reducing the candidate result space. On the on-chain side, the system records index commitments, query digests, and result digests to strengthen the binding of index states, the traceability of query behaviors, and the tracking capability of result-return processes. In this way, the system achieves secure data filtering under multi-attribute conditions and lightweight traceable recording while preserving data confidentiality and query privacy.

The main contributions of this paper are as follows.

(1) A blockchain-assisted privacy-preserving retrieval framework for sensitive outsourced data is proposed, enabling the collaboration between efficient off-chain retrieval and lightweight on-chain auditing.
(2) A frequency-aware primary search term selection mechanism is designed to reduce the candidate result space by preferentially selecting low-frequency query conditions, thereby improving the efficiency of multi-attribute conjunctive queries.
(3) A concealed auxiliary verification mechanism is constructed, which transforms auxiliary condition checking into set-level consistency verification, thus reducing the risk of exposing intermediate matching relationships.
(4) Security analysis and experimental evaluation are conducted to validate the proposed framework. The results show that, while preserving privacy, the scheme achieves better performance in construction time, query efficiency, and storage cost.

2 Related Work

2.1 Research on Protection and Retrieval of Sensitive Outsourced Data

In the area of sensitive outsourced data protection, Song et al. [13] proposed a cloud-secure storage mechanism based on data dispersion and encryption. By fragmenting sensitive data and combining it with encrypted storage, their scheme reduces the risk of data leakage caused by third-party platforms

directly accessing complete plaintext data. However, this method mainly focuses on confidentiality protection during the data storage stage and provides limited support for privacy-preserving queries after data outsourcing. Zhang et al. [14] proposed a policy-hidden attribute-based keyword search and data sharing scheme for cloud-assisted Internet of Things scenarios. By integrating attribute constraints with keyword retrieval, their scheme supports controlled search while ensuring secure sharing of sensitive data, thus alleviating the problem that encrypted data is difficult to retrieve in specific application domains. Nevertheless, its support for multi-attribute result filtering in general sensitive outsourced data scenarios remains limited. Wang et al. [15] presented a trusted sharing and multi-keyword retrieval scheme for sensitive data in multi-user environments. By improving the inverted index structure, the scheme supports multi-keyword retrieval while suppressing keyword-result pattern leakage, thereby mitigating the conflict between data usability and data protection in traditional sharing models. However, such methods mainly focus on secure retrieval in the sharing process, and they still provide insufficient discussion on efficient query organization and intermediate matching protection under complex multi-attribute conditions.

In studies on privacy-preserving retrieval over encrypted data, Cash et al. [16] proposed the Oblivious Cross-Tags (OXT) scheme, which divides conjunctive queries into a primary search term and auxiliary search terms, so that the query complexity mainly depends on the result size corresponding to the primary search term. This design addresses the low efficiency of early conjunctive retrieval schemes. However, OXT still leaks keyword-result patterns during execution, which introduces the risk of intermediate information exposure. Lai et al. [9] subsequently proposed the Hidden Cross-Tags (HXT) scheme, which suppresses keyword-result pattern leakage by introducing Bloom filters and hidden vector encryption. This approach alleviates the problem of intermediate matching leakage in OXT, but it requires additional rounds of interaction and incurs high storage overhead. Ma et al. [17] further proposed the Practical Hidden Cross-Tags (PHXT) scheme, which replaces the relatively heavy processing in HXT with a more efficient hash-based subset membership checking mechanism, thereby reducing storage and communication costs and improving the practicality of the scheme. Nevertheless, PHXT still cannot eliminate the additional interaction required during the query process. On this basis, Wang et al. [10] proposed a non-interactive encrypted conjunctive search scheme designed to suppress both keyword-result pattern leakage and cross-query intersection pattern leakage. By combining symmetric subset predicate encryption with a filtering

structure, Doris further reduces the computational overhead of membership checking, thereby alleviating the trade-off between interaction and leakage suppression in HXT and PHXT. However, the query efficiency of Doris is still affected by the selection of the primary search term, and its storage cost remains closely related to the underlying filtering structure.

2.2 Research on Blockchain-Assisted Trustworthy Recording and Traceability Mechanisms

Chakraborty et al. [18] proposed the BASPED (Blockchain-Assisted Searchable Public key Encryption over Outsourced Data) scheme, which introduces blockchain and smart contracts into the searchable public-key encryption process. Their scheme addresses the problem that, in traditional searchable encryption settings, cloud servers may return incorrect results that are difficult to verify. However, this work mainly focuses on retrieval result verifiability and server-side anti-cheating mechanisms, while paying limited attention to candidate result compression and the protection of intermediate matching relationships in multi-attribute conjunctive queries. Qiu [19] investigated the auditing problem of encrypted databases under the integration of searchable encryption and blockchain technology. By leveraging blockchain mechanisms, their approach enhances the trustworthiness of auditing in encrypted database environments and alleviates the limitations of traditional audit records, which are vulnerable to single-point control and lack sufficient credibility support. However, this line of work is more concerned with the auditing process itself, and does not further explore query organization, auxiliary condition verification, or result filtering in multi-attribute privacy-preserving retrieval. Banaeian Far et al. [20] proposed a blockchain-assisted general framework for on-chain auditing in off-chain storage environments. By storing the main data off-chain while performing auditing and verification on-chain, the framework reduces the storage and communication burden caused by directly writing large volumes of data onto the blockchain. Nevertheless, this framework is mainly designed for general auditing scenarios, and it does not specifically address the fine-grained coordination between query issuance, result return, and on-chain recording in privacy-preserving retrieval processes [21, 22].

Overall, existing studies have separately improved the protection and retrieval capabilities of sensitive outsourced data, as well as the trustworthy recording and traceability of retrieval processes. However, close integration between these two aspects within the same application scenario is

still lacking. In particular, under multi-attribute conditions, how to simultaneously balance query efficiency, intermediate information protection, and lightweight trustworthy traceability remains an open issue that deserves further investigation.

3 Preliminaries

3.1 Searchable Symmetric Encryption

Searchable Symmetric Encryption (SSE) is used to support keyword search over encrypted data. Its basic objective is to enable the searchability of outsourced data while preserving data confidentiality [23]. In general, an SSE scheme can be described by the following two stages.

(1) Encrypted database generation stage: The data owner processes the plaintext dataset DBDBDB and its index to generate an encrypted database EDBEDBEDB and then outsources the EDBEDBEDB to the cloud server.
(2) Query stage: An authorized user generates a search token for the queried keyword, and the server performs matching over the EDBEDBEDB based on the token and returns the corresponding results.

Because SSE provides a good balance between retrieval efficiency and system overhead, it is well suited for large-scale sensitive outsourced data scenarios.

3.2 OXT-Based Conjunctive Query

In multi-keyword conjunctive query scenarios, OXT is a representative query framework [24]. For the query

$$\Phi(w) = w_1 \wedge w_2 \wedge \cdots \wedge w_n \tag{1}$$

OXT-based schemes usually select one keyword as the primary search term, while treating the remaining keywords as auxiliary search terms. The server first extracts a candidate result set according to the primary search term and then performs additional verification for the remaining keywords. As a result, the overall query cost mainly depends on the size of the candidate result set associated with the primary search term. Let the primary search term be denoted as the $s-term$, and the remaining keywords as $x-term$. If the result set corresponding to the $s-term$ is large, the cost of subsequent auxiliary verification will also increase accordingly. Therefore, the selection

of the primary search term directly affects the efficiency of conjunctive queries. Meanwhile, traditional OXT may still leak intermediate information, such as keyword-result patterns, during the auxiliary verification process.

3.3 Count-Min Sketch

Count-Min Sketch (CMS) is a lightweight frequency estimation data structure that can approximately maintain the occurrence frequencies of elements with limited storage overhead [10]. Let the CMS consist of α hash functions and an $\alpha \times \beta$ counting matrix. When inserting an element x, the positions are computed using the corresponding hash functions and the associated counters are updated accordingly. When querying x, the minimum value among these counters is returned as its estimated frequency.

CMS has the following characteristics.

(1) It incurs fixed and relatively low storage overhead.
(2) It provides high efficiency for both update and query operations.
(3) It is suitable for approximate frequency maintenance over large-scale keyword sets.

In this paper, CMS is used to construct the keyword frequency table. During the query stage, the system preferentially selects the query condition with a lower estimated frequency as the primary search term, thereby reducing the candidate result space and lowering the cost of subsequent verification.

3.4 XBPE Mechanism

XOR-Extended Binary Fuse Filter based Predicate Encoding (XBPE) is an encoding mechanism for set relationship verification [25]. It can be used to compress and encode a given element set into a verifiable structure and supports subsequent set-level verification based on query tokens. The XBPE verification mechanism is illustrated in Figure 1. Specifically, let Y denote the set to be encoded and X denote the set to be tested. XBPE first generates the system parameters and the master secret key through an initialization algorithm. Then, during the database construction stage, the set Y is encoded together with a predefined predicate message into a ciphertext structure. During the query stage, the authorized user generates the corresponding query token according to the set X, and the server performs verification over the encoded structure using this token. When $X \subseteq Y$, the server can recover the predefined message; otherwise, it outputs $\perp$. In this way, the original problem of element-by-element matching is transformed into a single

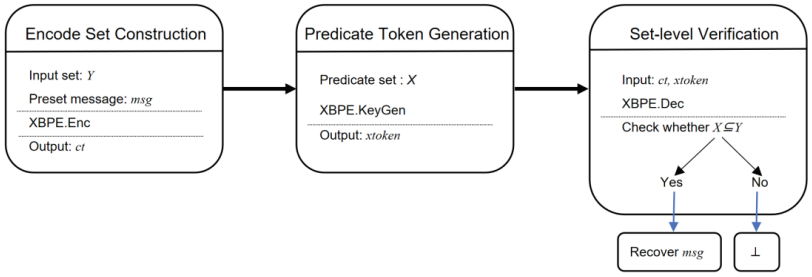

Figure 1 XBPE-based set-level predicate verification mechanism.

set-level consistency verification. Compared with methods that explicitly preserve matching relationships, XBPE is more suitable for integration with the auxiliary condition verification process in multi-attribute conjunctive queries. It can represent sets related to auxiliary verification in a compact form. On the other hand, it compresses the verification outcome into an overall decision of "satisfied" or "not satisfied," thereby supporting auxiliary condition verification in the subsequent query stage.

4 System Model and Security Requirements

4.1 System Model

The system considered in this paper consists of five entities: the Trusted Authority (TA), the Data Owner (DO), the Authorized User (AU), the Cloud Server (CS), and the Blockchain (BC). The overall system framework is illustrated in Figure 2.

Trusted Authority (TA): The TA is responsible for system initialization, including generating public parameters, registering system entities, and distributing the corresponding key materials. The TA participates in parameter configuration and identity management during the system setup stage but does not directly participate in the subsequent query processing procedure.

Data Owner (DO): The DO is responsible for collecting sensitive data and extracting relevant keywords or attribute information. After locally completing data encryption, frequency table construction, and encrypted index generation, the DO outsources the ciphertext data and encrypted index to the cloud server. Meanwhile, the DO generates an index digest and writes it to the blockchain, thereby establishing the binding between the off-chain index state and the on-chain commitment.

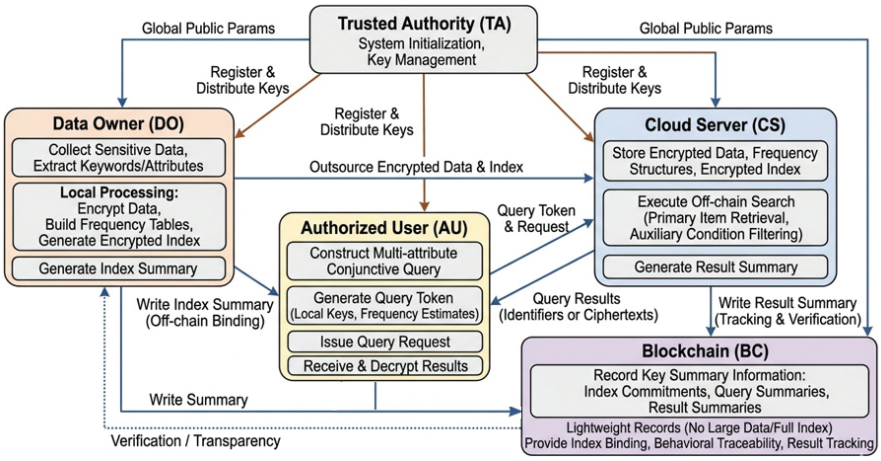

Figure 2 System architecture.

Authorized User (AU): The AU constructs multi-attribute conjunctive queries according to retrieval requirements, generates query tokens using local keys and frequency estimation results, and submits query requests to the cloud server. After the query is completed, the AU receives the identifiers of matched results or the corresponding ciphertext results and recovers the final query result within the authorized scope.

Cloud Server (CS): The CS is responsible for storing ciphertext data, frequency-related structures, encrypted index, and for performing off-chain retrieval operations according to the query tokens submitted by the AU. Specifically, the CS first extracts the candidate result set based on the primary search term and then filters the results satisfying multi-attribute constraints by combining the auxiliary condition verification mechanism. After that, the CS returns the final query result. Meanwhile, the CS generates a result digest and writes it to the blockchain for subsequent tracing and verification.

Blockchain (BC): The BC is used to record key digest information in the system, including the index commitment, query digest, and result digest. The blockchain does not store large-scale business data or the complete index content; instead, it only provides lightweight recording capabilities for index binding, behavior traceability, and result tracking.

Based on the above entities, the proposed system adopts a collaborative paradigm of off-chain storage and retrieval together with on-chain

commitment and auditing. The ciphertext data, frequency structures, and encrypted index are all stored on the cloud server, and the specific query processing is also completed off-chain. The blockchain is only responsible for recording key digest information, thereby enhancing the trustworthiness and traceability of the retrieval process.

4.2 Problem Definition

Let the sensitive dataset be denoted as

$$DB = \{(id_i, D_i)\}_{i=1}^{N} \tag{2}$$

where id_i is the unique identifier of sensitive data record i and D_i is the corresponding original data content. For each record, the DO extracts a set of keywords or attributes, denoted as

$$W_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,t_i}\} \tag{3}$$

Based on this, the complete keyword universe in the system is denoted as

$$W = \bigcup_{i=1}^{N} W_i \tag{4}$$

During the database construction stage, the DO encrypts the original dataset and the related index structures to generate the encrypted database $EDB = (C, \mathcal{I})$. Specifically, C denotes the set of ciphertext data and $\mathcal{I}$ denotes the encrypted index structure used to support privacy-preserving queries. During the query stage, the multi-attribute conjunctive query submitted by the AU can be expressed as

$$\Phi(w) = w_1 \wedge w_2 \wedge \cdots \wedge w_n \tag{5}$$

where $w_j \in W$ denotes the j-th keyword or attribute condition in the query and n is the number of query conditions. The goal of the system is to return the result set that satisfies all query conditions, together with the corresponding plaintext

$$Res(\Phi) = \{id_i \{w_1, w_2, \ldots, w_n\} \subseteq W_i\} \tag{6}$$

within the authorized scope, without directly exposing the query contents or intermediate matching relationships.

4.3 Security Requirements

To ensure the security of the system in sensitive outsourced data scenarios, the following security requirements are considered.

Data confidentiality: The data outsourced to the cloud server should not be exposed in plaintext form. Except that authorized users may recover query results within their legitimate access scope, the cloud server and other unauthorized entities should not obtain the original contents of the sensitive data.

Query privacy: The query conditions submitted by users should not be directly disclosed to the cloud server. The server is only allowed to perform matching operations based on query tokens, without recovering the complete query contents or directly inferring the user's retrieval intent.

Intermediate information protection: During multi-attribute conjunctive query processing, the server should not directly learn the intermediate matching relationships between candidate results and individual auxiliary conditions through the auxiliary verification process. The system should reduce the additional information leakage caused by item-by-item auxiliary checking as much as possible.

Index integrity and result traceability: The system should be able to effectively bind the off-chain index state and keep records of the query initiation and result return processes. With the help of on-chain digest information, the index state, query behavior, and result return process can be verified and traced afterwards.

Query efficiency: For multi-attribute conjunctive queries, the system should avoid the expansion of candidate results caused by an inappropriate primary search term as much as possible and reduce the overall retrieval cost through an efficient query organization method, so as to meet the practical requirements of sensitive outsourced data scenarios.

5 Proposed Scheme

5.1 Initialization Phase

The system initialization phase is carried out by the TA. Given the security parameter λ, the TA generates the public parameters required by the system together with the relevant key materials and then distributes them to the

DO and the AU, respectively. Specifically, the TA performs the following operations:

(1) Select a symmetric encryption algorithm $SE = (Enc, Dec)$, and initialize the pseudorandom function $F : \{0,1\}^{\lambda} \times \{0,1\}^{k} \rightarrow \{0,1\}^{\lambda}$, hash function $H : \{0,1\}^{*} \rightarrow \mathbb{Z}_q$;

(2) Initialize the XBPE parameters for auxiliary set verification, generate the master secret key msk and the corresponding public parameters, and run

$$XBPE.Setup(1^{\lambda}) \rightarrow (pp_x, msk) \tag{7}$$

where pp_x denotes the XBPE public parameters and msk denotes the master secret key. Here, $pp_x = (B, h_f, H)$ represents the public parameters required for XBPE encoding and determination, where B is the public description of the filtering structure, h_f is the fingerprint hash function, and $H = \{h_1, h_2, \ldots, h_\kappa\}$ denotes the set of hash functions used for set mapping.

(3) Randomly generate two types of system keys,$K_S \xleftarrow{\$} \{0,1\}^{\lambda}$ and $K_I \xleftarrow{\$} \{0,1\}^{\lambda}$, where K_S is used to generate primary search tags and K_I is used to generate auxiliary judgment-related labels.

After completing the above steps, the system enters the database creation phase.

5.2 Frequency Table Construction

To reduce the size of candidate results in multi-attribute conjunctive queries, this paper constructs a keyword frequency table during the database construction phase for subsequent main search term selection. Based on the complete keyword set *W* defined in Section 4.2, the DO uses CMS to construct a keyword frequency estimation structure.

Assuming the parameters (α, β) are determined during the initialization phase, DO constructs an $\alpha \times \beta$ counting matrix

$$C = (C[j][k])_{\alpha \times \beta} \tag{8}$$

and initializes it to $\emptyset$. Simultaneously, α hash functions $H_1, H_2, \ldots, H_\alpha$ are selected, where $H_j : W \rightarrow \{0, 1, \ldots, \beta - 1\}$, $1 \leq j \leq \alpha$. Subsequently, DO iterates through the entire keyword set W and updates the counting matrix based on the frequency of each keyword in the dataset. For any keyword $w \in W$, execute

$$C[j][H_j(w)] \leftarrow C[j][H_j(w)] + f(w), \quad 1 \leq j \leq \alpha \tag{9}$$

where $f(w)$ represents the true frequency of keyword w in the dataset. During the query phase, for a multi-attribute conjunctive query $\Phi(w) = w_1 \wedge w_2 \wedge \cdots \wedge w_n$, the system first estimates the frequency of each query keyword

$$\hat{f}(w_i) = \min_{1 \le j \le \alpha} C[j][H_j(w_i)], \quad 1 \le i \le n \tag{10}$$

Based on this, the keyword $\arg\min_{w_i \in \Phi(w)} \hat{f}(w_i)$ with the lowest estimated frequency is selected from the query condition set as the primary search term. The remaining keywords $\Phi(w) \backslash \{\arg\min_{w_i \in \Phi(w)} \hat{f}(w_i)\}$ constitute the auxiliary search term set. Through this frequency-aware selection mechanism, the system can prioritize determining the lower-frequency primary search conditions before querying.

5.3 Encrypted Database Construction

After system initialization and keyword frequency table construction, the DO encrypts sensitive data and its search structure, generating an encrypted database which is then outsourced to a cloud server. To simultaneously support main retrieval and auxiliary condition determination, this paper organizes the encrypted database into two parts: one part is used for extracting candidate results corresponding to the main search terms, and the other part is used for set-level consistency determination of auxiliary conditions. The final generated encrypted database can be represented as

$$EDB = (C, \mathcal{I}_m, \mathcal{I}_a) \tag{11}$$

where C represents the ciphertext data set, $\mathcal{I}_m$ represents the main search structure, and $\mathcal{I}_a$ represents the auxiliary determination structure.

5.3.1 Main search structure construction

The main search structure supports the server in quickly extracting a set of candidate results based on the main search term. For any keyword $w \in W$, DO first generates its corresponding search tag

$$stag_w = F(K_S, w) \tag{12}$$

where K_S is the main search tag key generated during the initialization phase. Subsequently, DO collects all record identifiers containing the keyword w and encrypts these identifiers using a symmetric encryption algorithm, obtaining the ciphertext identifier sequence related to w as follows

$$T(w) = \{e_{w,1}, e_{w,2}, \ldots, e_{w,|DB(w)|}\} \tag{13}$$

where $e_{w,c} = Enc(F(K_S, w), id_{w,c})$, $1 \leq c \leq |DB(w)|$. Based on this, DO uses the search tag $stag_w$ as an index and writes its corresponding ciphertext identifier sequence $T(w)$ into the main search structure $\mathcal{I}_m$. Therefore, $\mathcal{I}_m$ can be represented as

$$\mathcal{I}_m = \{(stag_w, T(w))w \in W\} \tag{14}$$

During the query phase, after an authorized user submits the search tag corresponding to the main search term, the server can extract a set of candidate results related to that keyword from $\mathcal{I}_m$. Since the size of the candidate results directly determines the processing overhead of subsequent auxiliary judgments, the main search structure provides the basic search entry point for the entire multi-attribute conjunctive query.

5.3.2 Construction of auxiliary decision structure

To avoid explicitly exposing the correspondence between candidate results and additional conditions during the auxiliary condition decision-making process, this paper further constructs an auxiliary decision structure $\mathcal{I}_a$. The set of cross-labels related to the records and auxiliary conditions is pre-encoded into XBPE ciphertext, thus transforming the subsequent item-by-item matching problem into a set-level consistency decision.

For any keyword $w \in W_i$, DO uses its position count c in the corresponding ciphertext identifier sequence as an index, combined with other keywords $w' \in W_i \backslash \{w\}$, to generate the cross-label

$$xtag_{i,w,w',c} = F(K_I, w \| w' \| c) \tag{15}$$

where K_I is the auxiliary decision-related label key. For record id_i, the set obtained by summarizing all its related cross-labels is

$$Y_i = \{xtag_{i,1}, xtag_{i,2}, \ldots, xtag_{i,m_i}\} \tag{16}$$

Subsequently, DO uses Y_i as the set to be encoded and utilizes the XBPE mechanism to generate the auxiliary decision ciphertext

$$ct_i \leftarrow XBPE.Enc(msk, True, Y_i) \tag{17}$$

where msk is the master key and $True$ is the preset decision message. Thus, each record id_i corresponds to an auxiliary decision ciphertext ct_i. Ultimately, the auxiliary decision structure $\mathcal{I}_a$ can be represented as

$$\mathcal{I}_a = \{(id_i, ct_i)\}_{i=1}^{N} \tag{18}$$

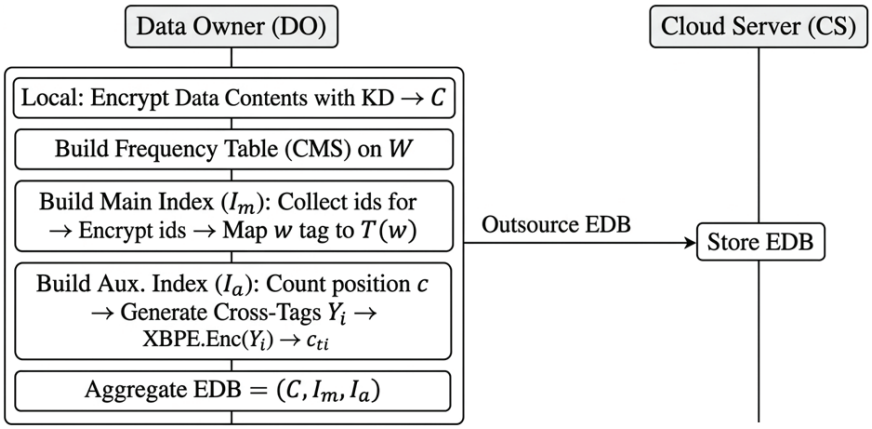

Figure 3 Encrypted database construction and database outsourcing.

During the query phase, the server does not need to check the matching relationship between each candidate result and each auxiliary condition. Instead, it directly determines whether the auxiliary set constructed in the query phase satisfies $X \supseteq Y_i$ through the XBPE decision process. If the determination is true, the message True is restored; otherwise, $\perp$ is output. Therefore, the auxiliary decision structure $\mathcal{I}_a$ actually provides compact coding support for the joint verification of multiple conditions on candidate results.

5.3.3 Data outsourcing

After the main retrieval structure and auxiliary decision structure are constructed, DO encrypts the original data content to obtain the ciphertext data set $C = \{C_i\}_{i=1}^{N}$, where $C_i = Enc(K_D, D_i)$ and K_D represent the data content encryption keys. Subsequently, DO uploads the ciphertext data set C, the main retrieval structure $\mathcal{I}_m$, and the auxiliary decision structure $\mathcal{I}_a$ to the cloud server to form the final encrypted database $EDB = (C, \mathcal{I}_m, \mathcal{I}_a)$, as shown in Figure 3. At this point, the system completes the construction of the encrypted database and enters the privacy-preserving query phase.

5.4 Privacy-Preserving Query

After completing the construction of the encrypted database and outsourcing it to the cloud server, AU can perform multi-attribute conjunctive queries on the encrypted database. For the query $\Phi(w) = w_1 \wedge w_2 \wedge \cdots \wedge w_n$, the entire

query process includes four steps: main search term selection, query token generation, candidate result filtering, and result return and recovery.

First, AU determines the main search term based on the frequency estimates of each query condition

$$w_s = \arg\min_{w_j \in \Phi} \hat{f}(w_j) \tag{19}$$

and sets the remaining query conditions as auxiliary search terms set

$$W_x = \Phi \backslash \{w_s\} \tag{20}$$

Subsequently, AU uses the main search tag key K_I to generate the main search token

$$\tau_s = F_{K_S}(w_s) \tag{21}$$

and sends τ_s to the cloud server CS. Upon receiving the token, CS locates the corresponding ciphertext identifier sequence $T(w_s)$ corresponding to w_s in the main search structure $\mathcal{I}_m$, obtaining the number of candidate results $m = |T(w_s)|$. Then, CS returns the candidate sequence length m to AU for subsequent auxiliary decision token generation. After knowing the candidate size, AU constructs the corresponding auxiliary decision set for each candidate position $c \in \{1, 2, \ldots, m\}$, combined with the auxiliary search term set W_x

$$X_c = \{F_{K_I}(w_s \| c \| w) : w \in W_x\} \tag{22}$$

Here, K_I is the auxiliary judgment related tag key. Subsequently, AU, based on the XBPE query interface, transforms the set X_c into the corresponding judgment token tok_c, and sends all auxiliary judgment tokens $Tok_x = \{tok_c\}_{c=1}^{m}$ to CS. Since these tokens consist of pseudo-random tags and XBPE query structures, CS cannot directly recover the specific content of the auxiliary retrieval items from them. After receiving Tok_x, CS performs auxiliary verification on the candidate results in T(ws) sequentially. Specifically, for the c-th candidate, let its corresponding ciphertext identifier be e_c, and its corresponding auxiliary judgment ciphertext be ct_c. CS calls the XBPE judgment algorithm to execute

$$XBPE.Query(pp, ct_c, tok_c) \rightarrow \{True, \perp\} \tag{23}$$

If the output result is $True$, it means that the candidate satisfies all auxiliary conditions, and CS adds the corresponding ciphertext identifier to the final result set

$$FRes = FRes \cup \{e_c\} \tag{24}$$

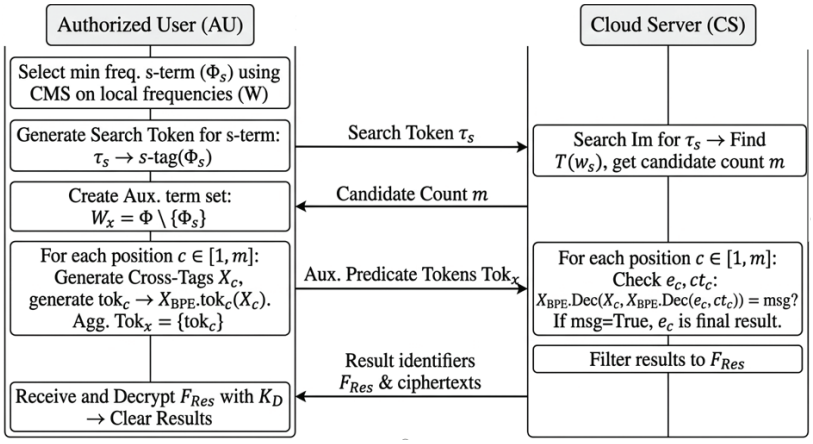

Figure 4 Privacy-preserving query.

If the output is $\perp$, the candidate is discarded. After the above processing, CS obtains the final result set $FRes$ that satisfies the multi-attribute conjunction condition. Finally, CS returns the corresponding result identifier and corresponding ciphertext result based on $FRes$. Within the authorized scope, the AU decrypts the returned ciphertext using the corresponding recovery key to obtain the final query result, as shown in Figure 4.

It should be noted that, in the above process, the CS performs off-chain matching operations only based on the main retrieval token and the auxiliary decision token and cannot directly obtain the complete query content. Simultaneously, the auxiliary condition verification is compressed into a set-level consistency determination; the server can only obtain the overall result and cannot directly observe the item-by-item correspondence between candidate records and each auxiliary condition. Therefore, this query process, while ensuring the efficiency of multi-attribute conjunctive retrieval, further enhances query privacy and the protection of intermediate information.

5.5 On-chain Recording Mechanism

To enhance the credibility of off-chain index states and retrieval processes, this paper implements lightweight recording of index commitments, query behaviors, and result return processes on the blockchain. The basic idea is to complete data storage, index organization, and privacy queries off-chain, while writing information representing key states in digest form to the blockchain, thus forming a verifiable and traceable record chain. Based

on this approach, this paper designs three types of on-chain records, denoted as index commitment transactions tx_{idx}, query digest transactions tx_q, and result digest transactions tx_r.

5.5.1 Index commitment record

After completing the construction of the encrypted database in Section 5.3, the DO generates a digest value for the off-chain index state

$$IndexRoot = H(\mathcal{I}_m \| \mathcal{I}_a \| ver \| ts) \tag{25}$$

where $\mathcal{I}_m$ represents the primary retrieval structure, $\mathcal{I}_a$ represents the auxiliary decision structure, ver represents the current index version number, and ts represents the database creation timestamp. Subsequently, DO constructs an index commitment transaction

$$tx_{idx} = (IndexRoot, ts, ver) \tag{26}$$

and uploads it to the blockchain. By recording tx_{idx}, the system can bind the off-chain index structure to the on-chain commitment without disclosing the specific index content. If the index is subsequently updated, DO recalculates the new $IndexRoot$ and generates a corresponding version of the index commitment record, thus forming a verifiable index evolution trajectory.

5.5.2 Query summary record

When an AU initiates a multi-attribute concatenation query, to avoid directly exposing the query content on the blockchain, the system only generates a summary of the relevant tokens and necessary metadata for this query and records it as a query summary transaction. Specifically, let the main retrieval token used in this query be τ_s, and the set of auxiliary decision tokens be $Tok_x = \{tok_c\}_{c=1}^{m}$, then AU can calculate the query summary

$$QH = H(\tau_s \| Tok_x \| ts_q) \tag{27}$$

where ts_q represents the query initiation timestamp. Subsequently, AU generates a query summary transaction

$$tx_q = (QH, ts_q, ver) \tag{28}$$

and uploads it to the blockchain. Here, QH is only used to identify a query behavior and its corresponding query status, without directly disclosing the query keywords, main retrieval item selection results, or auxiliary condition

content; ver is used to indicate the index version corresponding to this query, thereby associating the query behavior with the off-chain index status of a specific version. By recording ts_q, the system can leave a trace of the query initiation process, providing a basis for subsequent auditing and accountability.

5.5.3 Results summary record

After completing the candidate screening in Section 5.4, the cloud server CS obtains the final result set $FRes$. To achieve a reliable record of the result return process, CS does not directly write the result content to the blockchain, but instead generates a summary of the result set

$$RH = H(FRes \parallel QH \parallel ts_r) \tag{29}$$

where QH is the query summary corresponding to this query and ts_r represents the result return timestamp. Subsequently, CS constructs a result summary transaction

$$tx_r = (RH, QH, ts_r) \tag{30}$$

and uploads it to the blockchain. In this record, RH is used to represent the summary information of this returned result, QH is used to bind the result record with the corresponding query record, and ts_r is used to identify the result generation time. Thus, an associated record structure of "index commitment – query summary – result summary" can be formed on the chain. If it is necessary to verify a certain query process later, the association relationship between ts_q and ts_r can be used to confirm whether the result return behavior corresponds to a certain registered query operation, and the off-chain index status at that time can be traced in conjunction with the index version indicated by tx_{idx}.

6 Security Analysis

This paper analyzes the proposed scheme from three aspects: data confidentiality, query privacy and intermediate information protection, and index integrity and result traceability.

Data Confidentiality: During the database construction phase, the DO first performs symmetric encryption on the original sensitive data, obtaining a ciphertext data set C. This ciphertext data set C, along with the main retrieval structure $\mathcal{I}_m$ and the auxiliary decision structure $\mathcal{I}_a$, is then outsourced to the

cloud server. Therefore, the cloud server accesses the encrypted data content and corresponding index structure, not the original plaintext data.

For the main retrieval part, the server only sees the ciphertext identifier sequence T(w) organized by the main retrieval tags; for the auxiliary decision part, the server processes the XBPE-encoded auxiliary decision ciphertext, not the original keyword set corresponding to the record. Since the data content encryption key K_D is only used for result recovery within the legally authorized scope, cloud servers and other unauthorized entities that do not possess this key cannot directly recover sensitive data content from the ciphertext data set C.

Therefore, this scheme restricts the plaintext data exposure surface to the authorized recovery phase, ensuring that the data content in the outsourced storage environment is not leaked in plaintext form, thus meeting the data confidentiality requirements.

Privacy and Intermediate Information Protection: During the query phase, the AU does not directly send the query condition Φ to the cloud server. Instead, it first determines the main search term w_s based on the frequency estimation result and then generates a main search token $\tau_s = F_{K_S}(w_s)$ using the main search tag key K_S. The cloud server extracts candidate results in $\mathcal{I}_m$ based on τ_s, but cannot directly recover the actual content of the main search term from this token. For other auxiliary search terms, AU further constructs an auxiliary decision set using the auxiliary decision-related tag key K_I, and generates a decision token set Tok_x through the XBPE query interface. These tokens appear in the form of pseudo-random tags and set decision structures, and the server again cannot directly know the specific query conditions they correspond to.

Furthermore, in the traditional item-by-item auxiliary verification method, the server can often observe the individual matching relationship between candidate results and each additional condition, thus generating additional intermediate information leakage. In the scheme presented in this paper, the auxiliary condition-related information is pre-encoded into $\mathcal{I}_a$, and the server only performs set-level consistency determination during the query. For each candidate, the server ultimately only receives an overall output of "true" or "false" (True or $\perp$), without directly knowing which auxiliary conditions a candidate record satisfies or fails to satisfy.

Therefore, this proposed solution, on the one hand, avoids directly exposing the query content through a tokenization mechanism and, on the other hand, compresses observable information in the auxiliary verification process

using XBPE's set-level decision method, thereby simultaneously enhancing query privacy and the protection of intermediate information.

Index Integrity and Result Traceability: To enhance the credibility of off-chain index status and query processes, this paper introduces a collaborative mechanism of "off-chain storage and retrieval, on-chain commitment and auditing." Specifically, after the encrypted database is constructed, DO calculates the index digest $IndexRoot = H(\mathcal{I}_m \| \mathcal{I}_a \| ver \| ts)$ for the main retrieval structure $\mathcal{I}_m$ and the auxiliary decision structure $\mathcal{I}_a$, and writes the index commitment transaction $tx_{idx} = (IndexRoot, ts, ver)$ to the blockchain. Thus, the off-chain index status is bound to the on-chain commitment. If the index content is subsequently illegally tampered with, the recalculated digest value will be inconsistent with the on-chain record, thus enabling detection.

During the query execution phase, AU writes the query summary transaction $tx_q = (QH, ts_q, ver)$ to the blockchain; during the result return phase, CS writes the result summary transaction $tx_r = (RH, QH, ts_r)$ to the blockchain. This creates an on-chain associated record structure composed of tx_{idx}, tx_q, and ts_r. Where tx_q represents the existence of a query, ts_r represents the corresponding result return, and QH provides the link between the two. If subsequent auditing of a retrieval process is required, the query summary, result summary, and corresponding index version record can be combined to track and verify whether the query occurred, whether the result was returned, and the status of the indexes it depended on.

Due to the immutable and traceable characteristics of the blockchain, these records, once written, are difficult to delete or forge afterward. Therefore, this proposed solution can achieve lightweight and reliable traceability of index status, query behavior, and result return processes without directly putting large-scale indexes and business data on the blockchain, thus meeting the requirements of index integrity and result traceability.

Therefore, the blockchain-assisted privacy-preserving retrieval framework proposed in this paper can better meet the security requirements of data confidentiality, query privacy, intermediate information protection, and process traceability in sensitive outsourced data scenarios.

7 Experiments

7.1 Experimental Setup and Environment

All experiments were conducted in a unified environment. The experimental hardware platform consisted of an Intel(R) Core(TM) i5-12400F CPU

@ 2.50GHz and 16 GB RAM. The system was implemented in a Java environment. In terms of filtering, we used FAMF [26]. In addition, we used SL [27] to construct the frequency table. The experimental dataset was selected from Enron Email Dataset [28]. This paper treats each email as an independent record, extracts a set of keywords from the email subject and body, and constructs a complete set of keywords and an encrypted index after preprocessing such as case unification, word segmentation, and stop word removal. In order to examine the system performance under different conditions, the experiment further set up multi-attribute conjunctive queries with different data scales and different numbers of query conditions. This paper mainly evaluates the time overhead and storage overhead.

7.2 Experimental Analysis

In order to more accurately compare the costs of different auxiliary decision coding mechanisms in the library construction stage, this paper statistically analyzes the auxiliary decision structure construction time, that is, the time cost of encoding the set to be encoded into auxiliary decision ciphertext and generating auxiliary decision structure Ia. In the experiment, the proposed scheme uses XBPE to construct the auxiliary decision structure. Our scheme is compared with the XPE-based scheme LLMP proposed by Wang et al. [29], and the results are shown in Figure 5. As can be seen from Figure 5, the

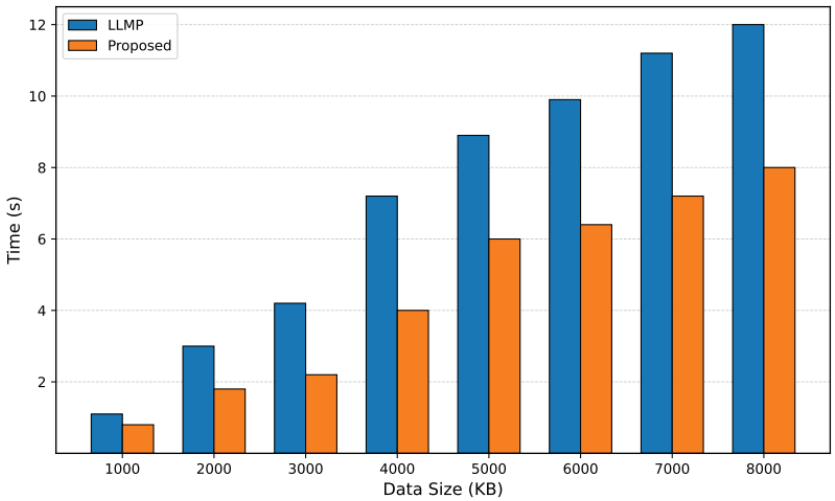

Figure 5 Comparison of auxiliary judgment structure construction time.

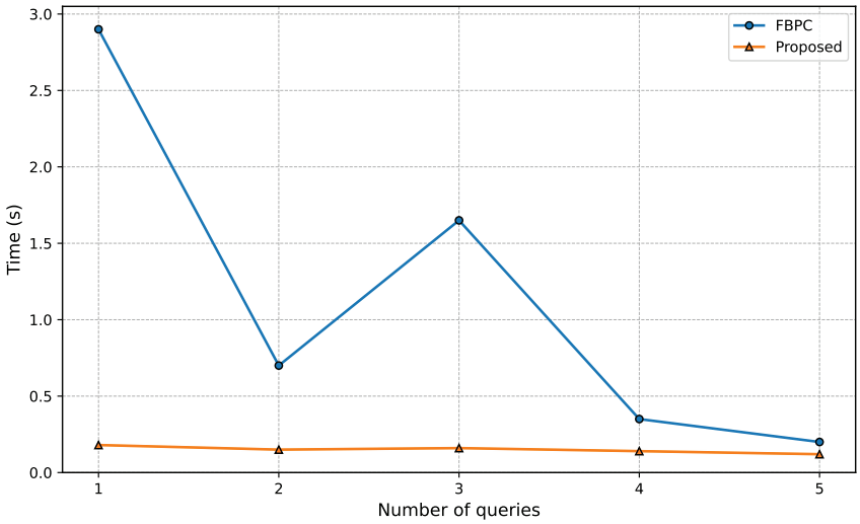

Figure 6 Search time comparison under different query instances.

scheme using XBPE performs better in terms of auxiliary decision structure construction time. That is to say, XBPE can improve the generation efficiency of auxiliary decision structure while maintaining the auxiliary decision function, thereby reducing the coding overhead in the library construction stage.

To further analyze the impact of the CMS mechanism on query performance, this paper designs a comparative experiment on search performance under different main search term selection strategies in the case of a small dataset (250 KB). In the experiment, the proposed scheme uses CMS to maintain a keyword frequency table on the client side and prioritizes the keyword with the lowest estimated frequency as the main search term, which is compared with the scheme FBPC proposed by Li et al. [30], which uses a random selection method for the main search term. The results are shown in Figure 6. The CMS-based method has more stable search time performance, and the overall fluctuation is significantly smaller than that of the random selection scheme.

To evaluate the impact of different auxiliary decision coding mechanisms on system space overhead, this paper conducts a storage cost comparison experiment. The impact of different auxiliary decision structures on the overall system space burden is analyzed as the data scale continuously increases. In the experiment, this paper uses an XBPE-based scheme as the research

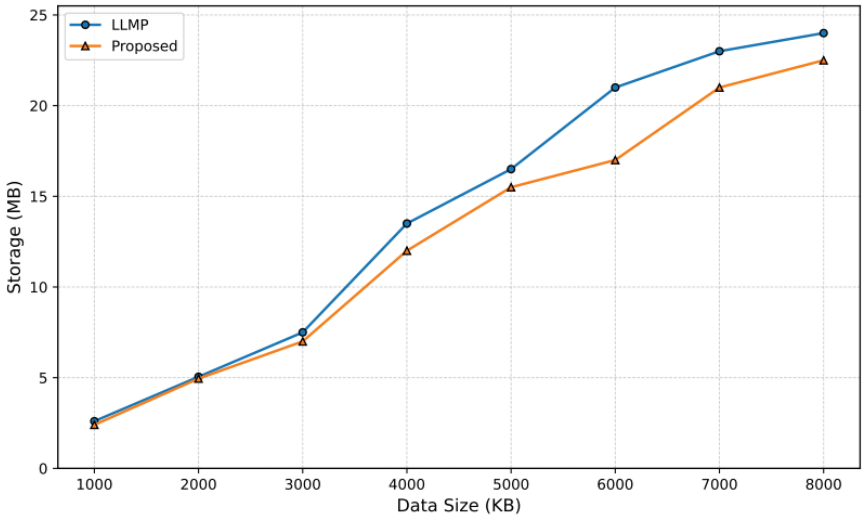

Figure 7 Comparison of storage costs for different data sizes.

object and compares it with LLMP. The results are shown in Figure 7. It can be seen that, when the data scale is small, the difference in storage cost between the two schemes is relatively limited. However, as the data volume continues to increase, the advantage of our scheme in terms of storage overhead gradually becomes apparent, significantly reducing the overall system storage cost.

8 Conclusion

This paper addresses the three requirements of "searchability, protection, and traceability" in sensitive outsourced data scenarios, designing a privacy-preserving retrieval framework for multi-attribute conjunctive queries. This method hierarchically organizes the main retrieval and auxiliary decision-making: on the one hand, it utilizes a frequency table to prioritize the selection of more suitable main retrieval terms, reducing the subsequent processing burden caused by invalid candidates; on the other hand, it uses XBPE to transform auxiliary condition verification into set-level decision-making, compressing intermediate exposure surfaces without changing the correctness of the query results. Simultaneously, this paper binds the index status, query initiation, and result return related summaries on-chain, enabling the system to perform efficient off-chain retrieval while possessing the foundation

for subsequent verification and auditing. Overall, this paper presents an implementation path that combines query organization optimization, privacy decision-making compression, and lightweight trusted records.

From the analysis and experimental results, the proposed scheme demonstrates good practicality in several key stages: in the database construction stage, the XBPE-based auxiliary decision-making structure has a lower construction cost; in the query stage, the CMS-based main retrieval term selection improves search stability; and in terms of storage, the spatial advantages of the scheme become more apparent as the data scale increases. This demonstrates that the proposed method not only meets the fundamental requirements of data confidentiality, query privacy, and protection of intermediate information, but also maintains a good balance between efficiency and system load.

However, there is still room for further improvement. Future work will focus on three aspects: first, researching index maintenance and consistency synchronization mechanisms for dynamic data updates; second, enhancing the system's applicability in complex authorization scenarios by incorporating finer-grained access control and result recovery processes; and third, conducting evaluations on larger-scale and more diverse real-world datasets, and further analyzing the impact of different parameter settings on query efficiency, storage costs, and system scalability. Through these extensions, the proposed framework is expected to better serve practical sensitive data management and analysis support tasks.

Funding

This research was supported by Research on Multi-dimensional Knowledge Graph Embedding and Lightweight Techniques for Knowledge Representation in Traditional Chinese Medicine (252103810297); 2025 Provincial Science and Technology R&D Program Joint Fund (252103810290): Integrated Application of Artificial Intelligence and Big Data for Smart Elderly Care-Research on Lightweight Behavior Recognition and Highly Reliable Early Warning Technologies Based on Multi-source Monitoring Data.

Declaration of Interest

The authors declare that it does not have any known interests or personal relationships that could potentially influence the reported work in this paper.

Data Availability

The data that support the findings of this study are available from Bian Zhu upon reasonable request.

Author Contributions

Bian Zhu: Writing – Original Draft, Writing – Review & Editing, Data Curation, Formal analysis; Ling Niu: Writing - Original Draft, Writing – Review & Editing, Resources.

Acknowledgements

During the manuscript writing stage, an artificial intelligence–assisted tool was used for language polishing to improve clarity and readability. The tool was not involved in the research design, experiments, or result analysis, and all scholarly content of this paper was completed by the authors, who take full responsibility for it.

References

[1] N. Soveizi, F. Turkmen, and D. Karastoyanova. Security and privacy concerns in cloud-based scientific and business workflows: A systematic review. *Future Generation Computer Systems*, 148: 184–200, 2023.

[2] J. Guo, C. Tian, X. Lu, L. Zhao, and Z. Duan. Multi-keyword ranked search with access control for multiple data owners in the cloud. *Journal of Information Security and Applications*, 82: 103742, 2024.

[3] M.E. Moudni and E. Ziyati. Advances and challenges in cloud data storage security: A systematic review. *International Journal of Safety & Security Engineering*, 15(4): 2025.

[4] H. Wu, B. Düdder, L. Wang, Z. Cao, J. Zhou, and X. Feng. Survey on secure keyword search over outsourced data: from cloud to blockchain-assisted architecture. *ACM Computing Surveys*, 56(3): 1–40, 2023.

[5] X. Zhang, D. Mu, and J. Zhao. Attribute-based keyword search encryption for power data protection. *High-Confidence Computing*, 3(2): 100115, 2023.

[6] M. Wang, L. Rui, S. Xu, Z. Gao, H. Liu, and S. Guo. A multi-keyword searchable encryption sensitive data trusted sharing scheme in multi-user scenario. *Computer Networks*, 237: 110045, 2023.

[7] P.S. Chakraborty, S. Tripathy, and S.K. Nayak. BASPED: Blockchain-assisted searchable public key encryption over outsourced data. *International Journal of Information Security*, 23(1): 2024.

[8] L. Yan, L. Ge, Z. Wang, G. Zhang, J. Xu, and Z. Hu. Access control scheme based on blockchain and attribute-based searchable encryption in cloud environment. *Journal of Cloud Computing*, 12(1): 61, 2023.

[9] S. Lai, S. Patranabis, A. Sakzad, J.K. Liu, D. Mukhopadhyay, R. Steinfeld, S. Sun, D. Liu, and C. Zuo. Result pattern hiding searchable encryption for conjunctive queries. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 745–762, 2018.

[10] Y. Wang, S.F. Sun, J. Wang, X. Chen, J.K. Liu, and D. Gu. Practical non-interactive encrypted conjunctive search with leakage suppression. *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 4658–4672, 2024.

[11] Y. Guo, C. Zhang, C. Wang, and X. Jia. Towards public verifiable and forward-privacy encrypted search by using blockchain. *IEEE Transactions on Dependable and Secure Computing*, 20(3): 2111–2126, 2022.

[12] L. Chen, W.K. Lee, C.C. Chang, K.K.R. Choo, and N. Zhang. Blockchain based searchable encryption for electronic health record sharing. *Future Generation Computer Systems*, 95: 420–429, 2019.

[13] H. Song, J. Li, and H. Li. A cloud secure storage mechanism based on data dispersion and encryption. *IEEE Access*, 9: 63745–63751, 2021.

[14] K. Zhang, Y. Li, and L. Lu. Privacy-preserving attribute-based keyword search with traceability and revocation for cloud-assisted IoT. *Security and Communication Networks*, 2021(1): 9929663, 2021.

[15] M. Wang, L. Rui, S. Xu, Z. Gao, H. Liu, and S. Guo. A multi-keyword searchable encryption sensitive data trusted sharing scheme in multi-user scenario. *Computer Networks*, 237: 110045, 2023.

[16] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.C. Roºu, and M. Steiner. Highly-scalable searchable symmetric encryption with support for Boolean queries. *Annual Cryptology Conference*, 353–373, 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.

[17] C. Ma, C. Jia, R. Du, G. Ha, and M. Li. Privacy-preserving searchable encryption based on anonymization and differential privacy. *2024 IEEE International Conference on Web Services (ICWS)*, 371–382, 2024.

[18] P.S. Chakraborty, S. Tripathy, and S.K. Nayak. BASPED: Blockchain-assisted searchable public key encryption over outsourced data. *International Journal of Information Security*, 23(1): 2024.

[19] J. Qiu. Ciphertext database audit technology under searchable encryption algorithm and blockchain technology. *Journal of Global Information Management (JGIM)*, 30(11): 1–17, 2022.

[20] S. Banaeian Far, M. Rajabzadeh Asaar, and A. Haghbin. A generic framework for blockchain-assisted on-chain auditing for off-chain storage. *International Journal of Information Security*, 23(3): 2407–2435, 2024.

[21] F. Li, J. Ma, Y. Miao, X. Liu, J. Ning, and R.H. Deng. A survey on searchable symmetric encryption. *ACM Computing Surveys*, 56(5): 1–42, 2023.

[22] Z. Gui, K.G. Paterson, and S. Patranabis. Rethinking searchable symmetric encryption. *2023 IEEE Symposium on Security and Privacy (SP)*, 1401–1418, 2023.

[23] J. Katz, M. Maffei, G. Malavolta, and D. Schröder. Subset predicate encryption and its applications. *International Conference on Cryptology and Network Security*, 115–134, 2017. Cham: Springer International Publishing.

[24] P. Aswar and S.T. Ali. Enhancements to ODXT SSE Scheme: Algorithm optimization and leakage suppression. *2024 International Conference on Distributed Systems, Computer Networks and Cybersecurity (ICDSCNC)*, 1–7, 2024.

[25] H.N.D. Nguyen, S. Cui, S. Lai, T.H. Yuen, and J.K. Liu. More practical non-interactive encrypted conjunctive search with leakage and storage suppression. *International Conference on Provable Security*, 329–349, 2025. Singapore: Springer Nature Singapore.

[26] T.M. Graf and D. Lemire. Fast Approximate Membership Filters in Java. https://github.com/FastFilter/fastfilter_java.

[27] Google. Stream Library. https://github.com/addthis/stream-lib.

[28] CALO Project. Enron Email Dataset. https://www.cs.cmu.edu/~enron/.

[29] Y. Wang, C. Gao, Y. Huang, L. Fu, and Y. Yu. Less leakage and more precise: Efficient wildcard keyword search over encrypted data. *High-Confidence Computing*, 5(3): 100297, 2025.

[30] S. Li, Y. Huang, Z. Fu, and B. Yu. Practical multi-source multi-client conjunctive searchable encryption with forward and backward privacy for cloud-IoT. *IEEE Internet of Things Journal*, 12(23): 49820–49842, 2025.

Biographies

Bian Zhu, lecturer, masters, graduated from Central China Normal University, China, in 2011. Worked at Zhoukou Normal University. Her research interests include privacy protection and trusted computing.

Ling Niu, associate professor, graduated from Chengdu University of Technology, China, in 2005. Worked at Zhoukou Normal University. Her research interests include graphics, image processing, and embedded systems.