electronics

MDPI

Article

An Efficient Ciphertext-Policy Decryptable Attribute-Based Keyword Search Scheme with Dynamic Attribute Support

Koon-Ming Chan [1], Swee-Huay Heng [2,*], Syh-Yuan Tan [1] and Shing-Chiang Tan [3]

[1] Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia; koonming1996@gmail.com (K.-M.C.); sytansy@gmail.com (S.-Y.T.)
[2] Centre for Intelligent Cloud Computing, CoE for Advanced Cloud, Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
[3] Centre for Advanced Analytics, CoE for Artificial Intelligence, Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia; sctan@mmu.edu.my
* Correspondence: shheng@mmu.edu.my

Abstract

Safeguarding data confidentiality and enforcing precise access regulation in cloud platforms continue to be major research concerns. Attribute-based encryption (ABE) offers a versatile framework for policy-driven control, whereas public key encryption with keyword search (PEKS) supports efficient querying of encrypted datasets. However, ABE lacks keyword search support, and PEKS offers limited control over access policies. To overcome these limitations, attribute-based keyword search (ABKS) schemes have been proposed, with recent advances such as ciphertext-policy decryptable ABKS (CP-DABKS) enabling secure channel-free keyword search. Nevertheless, the existing CP-DABKS schemes still face important challenges: the master public key grows linearly with the attribute universe, secure channels are often required to deliver trapdoors, and many designs remain vulnerable to keyword guessing attacks. This work introduces an efficient CP-DABKS scheme built upon a Type-3 pairing framework to directly overcome these limitations. The proposed design employs a commit-to-point mechanism that prevents linear key growth, eliminates the need for secure trapdoor transmission, and resists keyword guessing attacks. We implement and evaluate the proposed scheme using real-world data from the Enron Email dataset and demonstrate its practicality for secure and searchable cloud-based storage. We also discuss implementation considerations and outline directions for future enhancement of privacy-preserving searchable encryption systems.

Keywords: ciphertext-policy; attribute-based; keyword search; encryption; type-3 pairing

Academic Editor: Guosheng Xu

Received: 27 August 2025
Revised: 31 October 2025
Accepted: 3 November 2025
Published: 4 November 2025

Citation: Chan, K.-M.; Heng, S.-H.; Tan, S.-Y.; Tan, S.-C. An Efficient Ciphertext-Policy Decryptable Attribute-Based Keyword Search Scheme with Dynamic Attribute Support. *Electronics* **2025**, *14*, 4325. https://doi.org/10.3390/electronics14214325

1. Introduction

With the rapid growth of cloud computing, ensuring data confidentiality and enforcing precise access privileges have become critical requirements. Conventional approaches typically rely on data owners encrypting information and distributing decryption credentials to authorised users. The public key encryption with keyword search (PEKS) primitive allows legitimate users to perform queries on encrypted datasets through designated keywords without exposing the underlying plaintext. However, a significant drawback of PEKS is its inability to enforce fine-grained, attribute-based access control, limiting its utility in scenarios demanding precise data governance.

A principal security concern in PEKS schemes is their susceptibility to offline keyword guessing attacks (KGA) [1,2]. In such an attack, an adversary can successfully identify

a keyword by exploiting certain vulnerabilities. The feasibility of this attack stems from the relatively limited keyword space, which enables an adversary to perform exhaustive guessing. KGAs are generally classified into two categories based on the attacker's position: external and internal. An external KGA is conducted by malicious outsiders who utilise keyword ciphertexts and trapdoors intercepted from public channels. In contrast, a malicious insider executes an internal KGA, such as the cloud server, which leverages the keyword ciphertext, trapdoors provided by the sender and receiver, and the outcomes of the test function.

These approaches generally introduce complex key management procedures and impose considerable computational and administrative burdens on data owners. Although cloud computing provides cost-effective and scalable infrastructure, it also raises significant concerns regarding data confidentiality and access regulation when sensitive information is outsourced to untrusted servers [3]. To ensure data privacy and enforce controlled access, cryptographic primitives such as attribute-based encryption (ABE) have been widely adopted.

However, ABE itself does not support efficient keyword search capabilities, which are essential for large-scale cloud storage and retrieval systems. To bridge this gap, researchers have explored integrating searchable encryption into ABE-based frameworks, giving rise to attribute-based keyword search (ABKS) [4] and, more recently, ciphertext-policy decryptable ABKS (CP-DABKS) [5] schemes. While some existing schemes attempt to address this combination, many rely on Type-1 pairings [5,6], which are less efficient in real-world deployments due to larger key and ciphertext sizes, and higher computational overhead [7].

A fundamental limitation in many of these schemes is that the attribute universe must be fixed during system setup. Adding new attributes after deployment typically requires regenerating or redistributing the master public key, leading to significant challenges in scalability, key management, and secure system maintenance in practical cloud environments.

This study introduces an efficient CP-DABKS scheme built over the Type-3 pairing framework. The proposed design enables fine-grained access control and searchable encryption capability while operating securely even in the absence of a dedicated secure communication channel between the sender and the receiver. A notable enhancement is the integration of the commitment-to-point mechanism, which reduces reliance on a large global attribute universe and allows for efficient dynamic extension of attribute sets, facilitating flexible deployment in evolving access control environments.

1.1. Our Contributions

The main contributions of this paper are summarised as follows:

- We introduce a commit-to-point mechanism that supports dynamic and efficient attribute set expansion without reinitialising public parameters.
- We design a CP-DABKS scheme using Type-3 pairings to improve computation efficiency and reduce ciphertext size compared to Type-1 pairing-based designs.
- We implement the proposed CP-DABKS scheme and evaluate it on the Enron email dataset, demonstrating its practical feasibility and performance benefits in real-world scenarios.

1.2. Organisation of the Paper

The remainder of this paper is structured as follows. Section 2 provides a review of studies relevant to ABKS. Section 3 outlines the fundamental concepts employed in this work, including bilinear pairings, access structures, linear secret-sharing schemes, and the underlying computational hardness assumptions. Section 4 defines the CP-DABKS scheme and its associated security definition. Section 5 begins with the introduction of

a commit-to-point mechanism that enables scalable and supports a dynamic attribute universe, then details the design of the proposed CP-DABKS scheme implemented under the Type-3 pairing framework. Section 6 presents the formal security analysis under standard cryptographic assumptions. Section 7 provides the performance analysis of the Type-3 pairing-based CP-DABKS scheme. Future research directions are discussed in Section 8, and the paper is concluded in Section 9.

2. Related Work

Early attempts to support searchable encryption with access control include Hwang et al.'s [8] multi-user public key encryption scheme with conjunctive keyword search. Although it enables multi-user, multi-owner environments, the index size and search complexity grow with the number of authorised users, making it inefficient for large-scale cloud applications.

To simultaneously achieve fine-grained access control and efficient keyword search, various ciphertext-policy ABKS (CP-ABKS) constructions have been introduced. Wang et al. [9] introduced a CP-ABKS scheme using a linear secret sharing scheme (LSSS) to encrypt both keywords and messages. Sun et al. [4] incorporated user revocation via proxy re-encryption to reduce the online burden on data owners during membership updates.

For multi-owner systems supporting multi-keyword retrieval, Miao et al. [10] developed a construction that integrates ciphertext-policy ABE with conjunctive keyword-search capability to achieve fine-grained access control. They further expanded their work to hierarchical data sharing and verifiable keyword search [11].

He et al. [12] proposed a lightweight CP-ABKS scheme tailored for hybrid cloud environments, which supports conjunctive keyword search with reduced computational load. Yin et al. [13] extended the CP-ABE model of Bethencourt et al. [6] to design a ciphertext-policy ABKS scheme featuring expressive policy representation and improved access regulation.

Several works have also focused on improving efficiency for constrained devices or environments. Cao et al. [14] introduced a resource-efficient ABKS design tailored for wireless body area networks. Meanwhile, Cui et al. [15] employed an online/offline ABE mechanism with computation outsourcing to shift intensive decryption tasks to the cloud, thereby reducing the client's workload.

In the context of policy privacy and KGA resistance, Miao et al. [16] introduced a CP-ABKS construction that incorporates hidden-policy representation and traceability for shared multi-owner environments, asserting resilience against offline KGA. However, Sun et al. [17] demonstrated that their scheme remains vulnerable to four types of offline KGA attacks based on encrypted indexes and trapdoors. Pan et al. [18] introduced a PAEKS construction that strengthened resistance to KGA, representing a practical improvement in searchable encryption security.

Zhang et al. [19] proposed a secure ABKS scheme that supports hidden policies, traceability, and revocation, but it is limited to a single data owner setting, making it less applicable to general cloud sharing scenarios. Yan et al. [20] provided a recent survey of attribute-based searchable encryption, highlighting efficient revocation mechanisms as a key challenge for practical deployment.

Tang et al. [21] proposed VR-PEKS, a verifiable searchable encryption construction that explicitly targets resistance to keyword guessing attacks while providing verifiability of the search results, demonstrating another active direction for hardening searchable encryption against practical attacks.

In recent years, several studies have sought to enhance the practicality of attribute-based and pairing-based searchable encryption. Ref. [22] proposed a revocable and verifiable weighted ABE construction that enables collaborative access and efficient attribute updates in cloud environments. Ref. [23] presented an optimised dynamic attribute-based searchable encryption scheme that supports flexible attribute changes while maintaining low computational cost. Ref. [24] introduced FEASE, a fast and expressive asymmetric searchable encryption design implemented under a Type-3 pairing framework, achieving efficient keyword search with constant pairing operations. These developments highlight current progress in dynamic attribute encryption and Type-3 pairing applications.

Although these schemes have made significant progress in achieving fine-grained access control and searchable encryption, many of them still face practical limitations such as scalability issues, inefficient support for dynamic attribute updates, and vulnerability to keyword guessing attacks. In particular, most existing constructions assume a fixed attribute universe defined during system setup, making it difficult to incorporate new attributes without reinitialising public parameters or regenerating keys. Our work addresses these challenges by introducing a CP-DABKS scheme that supports scalable and dynamic attribute universe expansion through a lightweight commit-to-point mechanism while maintaining keyword search functionality and security guarantees. In addition, our construction leverages Type-3 pairings to improve computational efficiency and reduce the size of the ciphertext.

3. Preliminaries

This section outlines the core mathematical concepts and hardness assumptions that form the basis of our construction. We begin by establishing the notation: let p be a large prime. The set of integers modulo p is denoted by $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, and its multiplicative group, consisting of the non-zero elements, is $\mathbb{Z}_p^*$. The notation $x \in_R S$ indicates that an element x is sampled uniformly at random from a finite set S.

3.1. Bilinear Pairings

Following standard definitions in pairing-based cryptography [25,26], we recall the basic notions of symmetric and asymmetric bilinear maps used in this work.

Type-1 (symmetric) Pairing: Let $\mathbb{G}$ be a cyclic group of prime order p. A bilinear map is $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

Type-3 (asymmetric) Pairing: Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups of prime order p with generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. The pairing is $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Both pairing types satisfy the following fundamental properties:

- Bilinearity: For all $a, b \in \mathbb{Z}_p$ and valid group elements, the map is linear in each argument, i.e., $e(g^a, h^b) = e(g, h)^{ab}$.
- Non-degeneracy:
  - Type-1: For generators $g, h \in \mathbb{G}$, $e(g, g) \neq 1$ in $\mathbb{G}_T$.
  - Type-3: For generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, $e(g_1, g_2) \neq 1$ in $\mathbb{G}_T$.

  This ensures the pairing does not map all input pairs to the identity element.
- Efficient Computability: The function $e(\cdot, \cdot)$ can be evaluated efficiently for any inputs from the appropriate groups.

3.2. Access Structure

Following the standard definition of access structures in attribute-based encryption [6,27], let $P = \{P_1, P_2, \ldots, P_n\}$ denote a finite set of participants. A family of subsets $\mathbb{A} \subseteq 2^P$ is called *monotone* if, for any $B, C \subseteq P$, whenever $B \in \mathbb{A}$ and $B \subseteq C$, it follows that $C \in \mathbb{A}$. An *access structure* is therefore a monotone collection of non-empty subsets of P, written

as $\mathbb{A} \subseteq 2^P \setminus \{\emptyset\}$. The subsets in $\mathbb{A}$ are termed *authorised sets*, whereas those not in $\mathbb{A}$ are considered *unauthorised*.

3.3. Linear Secret Sharing Scheme

Following the standard definition in attribute-based encryption literature [6,28], a secret-sharing scheme Π over a participant set P is called *linear* over $\mathbb{Z}_p$ if the shares of all participants form linear combinations of the secret and random values in $\mathbb{Z}_p$. Formally, the scheme is associated with a share-generating matrix M of size $l \times n$, where each row i (for $i = 1, \ldots, l$) corresponds to a participant identified by a mapping $\rho(i)$.

To share a secret $s \in \mathbb{Z}_p$, the distributor selects a random vector $\vec{v} = (s, r_2, \ldots, r_n)^\top$, where $r_2, \ldots, r_n$ are uniformly chosen from $\mathbb{Z}_p$, and computes the share vector $\vec{\lambda} = M\vec{v}$. Each element λ_i in $\vec{\lambda}$ represents the share assigned to participant $\rho(i)$.

The defining reconstruction property of an LSSS is that, for any authorised set of participants represented by row indices I, there exist constants $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ such that the secret s can be recovered as:

$$\sum_{i \in I} w_i \lambda_i = s. \tag{1}$$

Intuitively, unauthorised subsets cannot derive any information about s without knowledge of all necessary shares.

3.4. Pedersen Commitment Scheme

We briefly recall the Pedersen commitment scheme [29], a well-known construction based on the hardness of the discrete logarithm problem. Let $\mathbb{G}$ be a cyclic group of prime order p with two independent generators g and h. The scheme consists of the following procedures:

- Setup**:** The parameters $(\mathbb{G}, p, g, h)$ are published.
- Commit**:** To commit to a value $m \in \mathbb{Z}_p$, the committer selects a random $r \in \mathbb{Z}_p$ and computes
$$C = g^m h^r \bmod p. \tag{2}$$
The output C serves as the commitment, which conceals both m and r.
- Open**:** To reveal the commitment, the committer discloses (m, r), and the verifier accepts if
$$C = g^m h^r \bmod p \tag{3}$$
holds.

The commitment value can also be denoted compactly as $C(m, r) = g^m h^r \bmod p$. Pedersen commitments provide perfect hiding, since C reveals no information about m, and computational binding, meaning that no efficient adversary can produce two distinct openings that correspond to the same C.

Definition 1 (Hiding). *Let $\pi = (Setup, Commit, Verify)$ be a Pedersen commitment scheme. A commitment scheme is perfectly hiding if, for every polynomial-time adversary A, the commitment value leaks no information about the underlying message. In other words, even an adversary with unbounded computational resources cannot distinguish between commitments to two different messages.*

This implies that the commitment reveals nothing about the committed value, ensuring that the committer's message remains fully hidden until it is later revealed with the corresponding opening information.

Definition 2 (Binding). *Let $\pi = (Setup, Commit, Verify)$ be a Pedersen commitment scheme. The commitment scheme achieves computational binding when no probabilistic polynomial-time adversary A can find two different openings yielding an identical commitment. That is, it is computationally infeasible for an adversary to find two different messages and corresponding randomness values that result in the same commitment output.*

This property guarantees that, after a commitment is generated, the committer cannot alter the committed value without being detected. The binding characteristic further ensures that each commitment corresponds to one unique message, thereby preventing the committer from unveiling it as a different value later.

3.5. Discrete Logarithm (DLOG) Assumption

We recall the discrete logarithm assumption, following the standard formulation in [30]. Let $\mathbb{G}$ be a cyclic group of prime order p with generator g. Given an element $h \in \mathbb{G}$, the discrete logarithm (DLOG) problem asks an adversary to determine $x \in \mathbb{Z}_p$ such that

$$g^x = h. \tag{4}$$

The DLOG assumption states that no probabilistic polynomial-time (PPT) adversary can efficiently recover x from (g, h) with non-negligible probability.

3.6. Computational Bilinear Diffie–Hellman (CBDH) Assumption

We next recall the computational Bilinear Diffie–Hellman assumption [25]. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map, and let $g \in \mathbb{G}$ be a generator. Given $(g, g^x, g^y, g^z) \in \mathbb{G}$ for random $x, y, z \in \mathbb{Z}_p^*$, the computational BDH problem asks to compute

$$e(g, g)^{xyz} \in \mathbb{G}_T. \tag{5}$$

The CBDH assumption holds if no PPT adversary can produce this value with more than negligible probability.

3.7. Decisional Bilinear Diffie–Hellman (DBDH) Assumption

The decisional version of the BDH assumption [25] is defined as follows. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map with generator $g \in \mathbb{G}$. Given a tuple (g, g^x, g^y, g^z, T) where $x, y, z \in \mathbb{Z}_p^*$ are random, and T is either $e(g, g)^{xyz}$ or a uniformly random element in $\mathbb{G}_T$, the adversary's task is to distinguish the two cases. The DBDH assumption asserts that no efficient adversary can decide whether $T = e(g, g)^{xyz}$ with non-negligible advantage over random guessing.

3.8. Division Decisional Diffie–Hellman (DDDH) Assumption

Following the formulation in [5], let $g \in \mathbb{G}$ be a generator, and let $y, s \in \mathbb{Z}_p^*$ be randomly chosen. Consider the tuple $(g, g^y, g^{s/y}, T)$, where $T \in \mathbb{G}$ is either g^s or a uniformly random element in $\mathbb{G}$. The Division Decisional Diffie–Hellman (DDDH) problem asks an adversary to determine whether $T = g^s$. The DDDH assumption holds if no probabilistic polynomial-time adversary can distinguish the two cases with non-negligible advantage.

3.9. Decisional q-Parallel Bilinear Diffie–Hellman Exponent (q-BDHE) Assumption

Following the formalisation used in [5], consider the tuple

$$\left(g, g^s, g^a, \ldots, g^{a^{2q}}, \{g^{sb_j}, g^{a/b_j}, \ldots, g^{a^{2q}/b_j}\}_{j=1}^{q}, \{g^{asb_k/b_j}, \ldots, g^{a^q sb_k/b_j}\}_{1 \le j \ne k \le q}\right), \tag{6}$$

where $a, s, b_1, \ldots, b_q \in \mathbb{Z}_p^*$ are randomly sampled. The Decisional q-Parallel Bilinear Diffie–Hellman Exponent (q-BDHE) problem requires determining whether a given element $T \in \mathbb{G}_T$ equals $e(g,g)^{a^{q+1}s}$ or represents a random value in $\mathbb{G}_T$. The q-BDHE assumption holds if no probabilistic polynomial-time adversary can make this distinction with non-negligible advantage.

4. CP-DABKS Scheme Definition and Security Definition

4.1. CP-DABKS Scheme Definition

Following the general structure of ciphertext-policy attribute-based keyword search systems [5], a CP-DABKS construction is defined as a suite of six polynomial-time algorithms that interact among the key generation centre (KGC), sender, receiver, and cloud server. The algorithms are briefly described below.

- **Setup:** Executed by the KGC, this algorithm initialises the system on input of the security parameter 1^λ and the initial attribute universe U. It outputs the public configuration parameters cp and the master secret key msk.
- **Key Generation:** Each entity generates its own cryptographic keys according to its role. The sender derives a pair (pk_S, sk_S) from cp; the cloud server produces (pk_{CS}, sk_{CS}) using the same parameters; and for each receiver with attribute set $U \subseteq \mathcal{U}$, the KGC issues an attribute secret key SK and an individual key pair (pk_R, sk_R) based on msk and cp.
- **Encryption:** Given a message m, a keyword w, and an access policy (M, ρ), the sender uses $(cp, pk_{CS}, pk_R, sk_S)$ to produce the ciphertext CT, which embeds both the encrypted message and keyword components consistent with the specified policy.
- **Trapdoor Generation:** A receiver possessing an attribute key SK computes a search token T_w for keyword w using (cp, sk_R, pk_S), enabling the cloud server to perform keyword testing without revealing w.
- **Testing:** The cloud server, on input (M, ρ), the token T_w, and ciphertext CT, verifies whether the receiver's attribute set satisfies the policy and whether the keyword matches. If successful, it returns a partially decrypted ciphertext CT'; otherwise it outputs $\perp$.
- **Decryption:** The receiver applies its private key sk_R to CT' to recover the plaintext message m.

4.2. Security Definition

The security framework of our CP-DABKS construction follows the conventional game-based formulation used in ciphertext-policy searchable encryption [5]. For completeness, we restate the relevant notions in simplified form.

4.2.1. CP-ABE-CPA Security

The chosen-plaintext security (CPA) of the CP-ABE construction is expressed through an interaction between a challenger and an adversary:

- **Initialisation:** The adversary $\mathcal{A}$ commits to a challenge access structure (M^*, ρ^*).
- **Setup:** The challenger executes the Setup algorithm and provides the public system parameters to $\mathcal{A}$.
- **Query Phase 1:** $\mathcal{A}$ adaptively issues secret key queries for sets of attributes S. The challenger responds only if S does not satisfy (M^*, ρ^*).
- **Challenge:** The adversary $\mathcal{A}$ provides two equal-length messages, m_0 and m_1. The challenger randomly selects $\beta \in 0, 1$, encrypts m_β with respect to (M, ρ), and returns the ciphertext to the adversary $\mathcal{A}$.
- **Query Phase 2:** This phase is a repetition of Phase 1, under the same constraint.
- **Guess:** The adversary outputs its guess β' for β.

The advantage of an adversary $\mathcal{A}$ is given by $\left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$. The scheme achieves CP-ABE-CPA security if this advantage remains negligible for all polynomial-time adversaries.

4.2.2. IND-CPA Security

The confidentiality of the system under chosen-plaintext attacks (CPA) is described by a security game in which the adversary plays the role of the cloud server:

- **Setup:** The challenger initialises the system, generating all common parameters and key pairs. The adversary, acting as the cloud server, submits its public key to the challenger.
- **Query Phase 1:** The adversary adaptively requests the encryption of any messages of its choice and receives the corresponding ciphertexts from the challenger.
- **Challenge:** The adversary selects two distinct messages m_0^* and m_1^* of identical length. The challenger samples a random bit $\beta \in \{0, 1\}$ encrypts m_β^*, and outputs the corresponding challenge ciphertext.
- **Query Phase 2:** The adversary may issue further encryption queries, as in Phase 1, but is forbidden from querying the challenge messages m_0^* or m_1^*.
- **Guess:** The adversary outputs its guess β' for β.

The adversary's success probability is measured as $\left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$. A scheme is IND-CPA secure when this probability is negligible for all efficient adversaries.

4.2.3. IND-KGA Security

This models the security against offline keyword guessing attacks by a cloud server:

- **Setup:** The challenger initialises the system parameters. The adversarial cloud server generates its key pair and provides the public component to the challenger.
- **Query Phase 1:** The adversary may request trapdoors for arbitrary keywords w and obtain ciphertexts for chosen terms. The challenger answers all such queries faithfully.
- **Challenge:** The adversary selects two keywords w_0^* and w_1^*, after which the challenger samples a random bit $\beta \in \{0, 1\}$ and issues the ciphertext associated with w_β^*.
- **Query Phase 2:** The adversary can continue to make trapdoor and ciphertext queries for any keyword, with the sole restriction that it cannot query w_0^* or w_1^*.
- **Guess:** The adversary outputs its guess β' for β.

The adversary's advantage is defined as $\left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$. The scheme achieves IND-KGA security if no polynomial-time adversary can gain a non-negligible advantage.

4.2.4. IND-CKA Security

The scheme achieves security against chosen-keyword attacks (CKA) in scenarios where the cloud server behaves maliciously:

- **Setup and Query Phase 1:** Identical to the setup and first query phase defined in the IND-KGA game.
- **Challenge:** The adversary chooses two challenge keywords, w_0^* and w_1^*. The challenger then samples a random bit $\beta \in \{0, 1\}$ and returns the ciphertext corresponding to the keyword w_β^*.
- **Query Phase 2:** The adversary may again issue queries as in Phase 1, with the exception that it cannot request trapdoors or ciphertexts for the challenge keywords w_0^* or w_1^*.
- **Guess:** The adversary outputs its guess β' for β.

The adversary's advantage is defined as $\left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$. The scheme achieves IND-CKA security if no polynomial-time adversary can gain a non-negligible advantage.

5. CP-DABKS Scheme with Dynamic Attribute Universe

This section begins by introducing the commit-to-point mechanism, a key technique that enhances the flexibility and scalability of attribute management in our scheme. We then present the construction of our CP-DABKS scheme in the Type-3 pairing setting. While the design is inspired by the core ideas of Guo et al. [5].

5.1. Commit-to-Point

We introduce a secure improvement in the Type-3 pairing setting to enhance practical performance in attribute handling.

In cloud environments, a user may possess multiple identity attributes. For instance, an AWS user can have dozens of identity attributes in the AWS Identity and Access Management (IAM) Identity Centre. Therefore, even when considering only IAM attributes for a cloud service with n users, the encryption process already requires handling a universe of $|U| = 34n$ attributes. Each attribute must be mapped to a pair of group elements (h_i, h_i'), with each element containing $616 + 1632 = 2248$ bits. Moreover, every time an attribute is modified or newly introduced, the attribute universe U is extended by appending a new element pair (h_i, h_i'), which is also incorporated into the master public key.

To avoid the inefficiency of searching, storing, and updating these elements explicitly, a one-way function can be employed to deterministically encode attributes into their associated group elements. A simple yet secure approach is to compute (h_i, h_i') as:

$$h_i = g_1^{H(\text{attr})} \tilde{g}_1, \quad h_i' = g_2^{H(\text{attr})} \tilde{g}_2 \tag{7}$$

Here, $H : 0,1^* \to \mathbb{Z}_p^*$ denotes a cryptographic hash (e.g., SHA-256) that provides collision resistance and produces outputs uniformly distributed in $\mathbb{Z}_p^*$, and $(\tilde{g}_1 = g_1^x, \tilde{g}_2 = g_2^x)$ are additional master public key elements generated using a randomly chosen $x \in \mathbb{Z}_p^*$ during Setup. The exponent x can be discarded once these elements are derived, as it effectively serves as a secret trapdoor for the one-way mapping above.

In subsequent phases, h_i' is raised to the exponent t during KeyGen_R and h_i is raised to $-r_i$ during Encrypt, both operations resemble Pedersen commitments, which are known to be computationally binding. Since the Pedersen commitment scheme also functions as a chameleon hash [31], the values $h_i'^t$ and $h_i^{-r_i}$ can be regarded as collision-resistant chameleon hashes.

Theorem 1. *Let the commit-to-point map be instantiated as a Pedersen commitment. Then h_i is computationally binding assuming the DLOG problem is hard.*

Proof. The Pedersen commitment is computationally binding; that is, no efficient adversary can generate two different openings (m, r) and (m', r') that correspond to the same commitment value. Let $m = H(\text{attr})$, and assume an adversary attempts to find such pairs (m, r) and (m', r') satisfying this condition.

$$g_1^m \, \tilde{g}_1 \;=\; g_1^{m'} \, \tilde{g}_1^{r'}. \tag{8}$$

Recall from Setup that $\tilde{g}_1 = g_1^{\xi}$ for some secret $\xi \in \mathbb{Z}_p$. Substituting this form into the equality and writing exponents additively gives

$$m + \xi \;\equiv\; m' + \xi r' \pmod{p}. \tag{9}$$

Rearranging terms yields

$$\xi(1 - r') \;\equiv\; m' - m \pmod{p}. \tag{10}$$

If $r' \not\equiv 1 \pmod{p}$, we can solve for ξ as

$$\xi \equiv \frac{m' - m}{1 - r'} \pmod{p}. \tag{11}$$

Thus, an adversary who outputs two valid openings would recover ξ, i.e., the discrete logarithm of $\widetilde{g}_1$ relative to g_1, thereby violating the DLOG assumption. The special case $r' \equiv 1 \pmod{p}$ can be excluded, as it would simplify the equation to $m \equiv m' \pmod{p}$, contradicting the distinctness of the openings. Therefore, finding two different openings is computationally infeasible, and the commitment is binding under the DLOG assumption. □

5.2. The Proposed CP-DABKS Construction in Type-3 Pairings

The proposed CP-DABKS design, realised on Type-3 pairings, is outlined below and consists of the following polynomial-time procedures:

- **Setup** $(\lambda, U) \to (cp, msk)$: The KGC initialises the scheme by taking the security parameter λ and the initial attribute universe U as inputs, then performing the following setup procedure:
  1. Choose cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, each of prime order p.
  2. Define generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.
  3. Establish a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.
  4. Sample random elements $\alpha, a \in_R \mathbb{Z}_p^*$ and $Q, h \in_R \mathbb{G}_1$.
  5. For each attribute i in the universe U, select a random exponent $t_i \in_R \mathbb{Z}_p^*$.
  6. Define two cryptographic hash functions: $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \mathbb{G}_T \to \{0,1\}^k$ for a suitable key length k.
  7. The common parameters cp are published as:

$$cp = \left(g_1, g_2, e(g_1, g_2)^{\alpha}, g_1^a, g_2^a, Q, h, H_1, H_2, \left\{h_i = g_1^{t_i}\right\}_{i \in U}, \left\{h_i' = g_2^{t_i}\right\}_{i \in U}\right) \tag{12}$$

  8. The Master Secret Key is set to $msk = g_2^{\alpha}$.
- **KeyGen**$_S$ $(cp) \to (pk_S, sk_S)$: A data sender generates its key pair by picking a random $z \in_R \mathbb{Z}_p^*$ and setting the public key $pk_S = g_2^z$ and the private key $sk_S = z$.
- **KeyGen**$_{CS}$ $(cp) \to (pk_{CS}, sk_{CS})$: The cloud server selects a random $x \in_R \mathbb{Z}_p^*$, defining its public key as $pk_{CS} = g_2^x$ and its private key as $sk_{CS} = x$.
- **KeyGen**$_R$ $(cp, msk, S) \to (pk_R, sk_R, SK)$: Given a receiver whose attribute set is $S \subseteq U$, the KGC performs the following:
  1. Choose two random values $u, v \in_R \mathbb{Z}_p^*$.
  2. Compute the primary decryption component $K = g_2^{\alpha} g_2^{au}$.
  3. Compute $L = g_2^u$.
  4. For every attribute $i \in S$, generate an attribute-specific key $K_i = h_i^u$.
  5. The receiver's attribute-based secret key is $SK = (K, L, \{K_i\}_{i \in S})$.
  6. The receiver independently produces a personal key pair, where the public key is $pk_R = g_2^v$ and the secret key is $sk_R = v$.
- **Encrypt** $((M, \rho), cp, pk_{CS}, pk_R, sk_S, w, m) \to CT$: To encrypt a message m with a keyword w under an LSSS access policy (M, ρ), where M is an $l \times n$ matrix, the sender executes:
  1. Policy Encryption:
     - Choose a random vector $\vec{v} = (s, y_2, \ldots, y_n) \in_R \mathbb{Z}_p^n$ to share the secret s.
     - For each row i of M $(1 \le i \le l)$, select a random $r_i \in_R \mathbb{Z}_p^*$.

– Compute the policy ciphertext components:

$$C' = g_1^s, \quad \forall i : \left(C_i' = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i}, \quad D_i' = g_1^{r_i}\right) \tag{13}$$

In this construction, $\lambda_i = M_i \cdot \vec{v}$, where M_i denotes the i-th row of the access matrix M. Let $C_1 = (C', \{C_i', D_i'\}_{i=1}^{l})$.

2. Message Encryption:
   – Pick a random $t \in_R \mathbb{Z}_p^*$.
   – Compute the message ciphertext:

$$C_0 = g_1^t, \quad C_m = m \oplus H_2\left(e(g_1, g_2)^{\alpha s} \cdot e(Q, pk_{CS})^t\right) \tag{14}$$

Let $C_2 = (C_0, C_m)$.

3. Keyword Encryption:
   – Select a random $r \in_R \mathbb{Z}_p^*$.
   – Compute the keyword ciphertext:

$$C_3' = g_1^r \tag{15}$$

$$C_4' = pk_S^r \tag{16}$$

$$C_w = H_2\left(e\left(H_1(w)^{sk_R} \cdot h^r, pk_S\right) \cdot e(Q, pk_{CS})^r\right) \cdot e(g_1, g_2)^{\alpha s} \tag{17}$$

Let $C_3 = (C_3', C_4', C_w)$.

4. The resulting ciphertext $CT = (C_1, C_2, C_3)$ is subsequently transmitted to the cloud server for storage.

- **Trapdoor** $(cp, SK, sk_R, pk_S, w') \rightarrow T_w$: The receiver computes a search trapdoor corresponding to the keyword w' as follows:

$$T_{w,1} = H_1(w')^{sk_R}, \quad T_{w,2} = \mathcal{E}_{pk_{CS}}(SK) \tag{18}$$

where $\mathcal{E}$ is a secure public key encryption algorithm. The generated trapdoor $T_w = (T_{w,1}, T_{w,2})$ is transmitted to the cloud server, while the secret key SK remains encrypted using the server's public key pk_{CS}.

- **Test** $((M, \rho), cp, T_w, CT, sk_{CS}) \rightarrow CT' / \perp$: Upon receiving a trapdoor, the cloud server:
  1. Decrypts $T_{w,2}$ using its private key sk_{CS} to recover the receiver's $SK = (K, L, \{K_i\})$.
  2. It verifies whether the attributes embedded in SK fulfil the access policy (M, ρ). If not, it aborts and outputs $\perp$.
  3. Let $I \subset \{1, 2, \ldots, l\}$ be the set of indices for which the attributes $\rho(i)$ are held by the receiver, and let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ denote a collection of reconstruction coefficients satisfying $\sum_{i \in I} \omega_i \lambda_i = s$.
  4. Computes:

$$Q_1 = \frac{e(C', K)}{\prod_{i \in I}\left(e(C_i', L) \cdot e(D_i', K_{\rho(i)})\right)^{\omega_i}} = e(g_1, g_2)^{\alpha s} \tag{19}$$

  5. Then, it verifies if the keyword matches by checking:

$$\frac{C_w}{Q_1} \stackrel{?}{=} H_2\left(e(T_{w,1}, C_4') \cdot e(h, C_3') \cdot e(Q^{sk_{CS}}, C_3')\right) \tag{20}$$

If the equation holds, the server identifies the message ciphertext C_2 and computes a partial ciphertext $CT' = (C_0, C_m)$ which it returns to the receiver. Otherwise, it outputs $\perp$.

- **Decrypt** $(cp, CT', sk_R) \rightarrow m$: The receiver finalises the decryption using its private key sk_R:

$$m = C_m \oplus H_2\Big(e(Q^{sk_R}, C_0)\Big) \tag{21}$$

This step recovers the original plaintext message.

5.3. Correctness

The correctness of the construction over Type-3 pairings can be demonstrated as follows:

The Test algorithm:

$$\begin{aligned} Q_t &= \frac{e(C', K)}{\prod_{i\in I}\Big(e(C'_i, L)e(D'_i, K_{\rho(i)})\Big)^{w_i}} \\ &= \frac{e(g_1^s, g_2^{\alpha} g_2^{at})}{\prod_{i\in I}\Big(e(g_1^{a\lambda_i} h_{\rho(i)}^{-r_i}, g_2^t)e(g_1^{r_i}, h'^t_{\rho(i)})\Big)^{w_i}} \\ &= \frac{e(g_1, g_2)^{\alpha s} e(g_1, g_2)^{ast}}{\prod_{i\in I} e(g_1, g_2)^{ta\lambda_i w_i}} \\ &= \frac{e(g_1, g_2)^{\alpha s} e(g_1, g_2)^{ast}}{e(g_1, g_2)^{ast}} \\ &= e(g_1, g_2)^{\alpha s} \end{aligned} \tag{22}$$

and

$$\begin{aligned} \frac{C_5}{Q_t} &= H_2\Big[e(H_1(w)^{sk_S} h^r, pk_R)e(Q, pk_{CS})^r\Big] \\ &= H_2\Big[e(H_1(w)^{sk_S}, pk_R)e(h^r, pk_R)e(Q, g_2^{sk_{CS}})^r\Big] \\ &= H_2\Big[e(H_1(w)^{sk_R}, pk_S)e(h, pk_R^r)e(Q^{sk_{CS}}, g_2^r)\Big] \\ &= H_2\Big[T_{w,1}e(C_4, h)e(Q^{sk_{CS}}, C_3)\Big] \end{aligned} \tag{23}$$

within the Decrypt phase:

$$\begin{aligned} \frac{ct'}{H_2[e(Q^{sk_R}, C_1)]} &= \frac{mH_2[e(Q, pk_R)^b]}{H_2[e(Q^{sk_R}, C_1)]} \\ &= \frac{mH_2[e(Q, g_2^{sk_R})^b]}{H_2[e(Q^{sk_R}, C_1)]} \\ &= \frac{mH_2[e(Q^{sk_R}, g_2^b)]}{H_2[e(Q^{sk_R}, C_1)]} \\ &= m \end{aligned} \tag{24}$$

6. Security Analysis

Theorem 2. *The proposed CP-DABKS scheme in Type-3 pairings is CP-ABE-CPA secure under the decisional q-parallel BDHE assumption, IND-CPA secure under the CBDH assumption, IND-KGA secure under the DBDH assumption, and IND-CKA secure under the division DDH assumption.*

Proof. The CP-DABKS design of Guo et al. [5], implemented on Type-1 symmetric pairings, was proven CP-ABE-CPA secure under the q-parallel BDHE assumption, IND-CPA secure

via the CBDH assumption, IND-KGA secure under the DBDH assumption, and IND-CKA secure according to the division DDH assumption. Migrating a construction from the symmetric (Type-1) to the asymmetric (Type-3) pairing framework consequently modifies how these foundational hardness assumptions are instantiated. Abe et al. [32] demonstrate that assumptions proven secure under the generic bilinear group model for Type-1 pairings also remain valid when instantiated in Type-3 pairings. Therefore, the security of the proposed scheme remains based on the same assumptions when implemented in a Type-3 pairing setting. □

7. Performance Analysis

In this section, the performance of the Type-3 pairings CP-DABKS scheme is analysed by performing the actual tests using the real-world email dataset. All experiments were executed on a Windows 11 Home 22H2 system featuring an Intel Core i5-1135G7 processor (2.4 GHz). The implementation was carried out in Java with the MIRACL Core library, employing the BLS12381 curve for Type-3 pairing operations.

For simplification, the analysis focuses on a single user, John Arnold. In particular, 3852 emails from John Arnold were examined. From these emails, the 15 most frequently used keywords by John Arnold were selected, as shown in Table 1. Among them, 667 emails (including duplicates) contained the keyword "forwarded" in their body, therefore, "forwarded" was chosen as the keyword for this study. To run the KeyGen_R algorithm, the receiver is assigned all the attributes, i.e., $attr$ = "ReadOnly, Hidden, System, Directory, Archive, Device, Normal, Temporary, SparseFile, ReparseFile, ReparsePoint, Compressed, Offline, NotContentIndexed, Encrypted, IntegrityStream, Virtual, Pinned, Unpinned, RecallOnOpen, RecallOnDataAccess". The analysis involved testing the performance of the proposed scheme's Test algorithm using the keyword "forwarded" with different thresholds. The threshold parameter t defines the minimum number of attributes that must satisfy the ciphertext policy for a valid keyword test and decryption. For example, when $t = 3$, a user key must contain at least three attributes that meet the policy conditions. A lower t value provides broader accessibility, while a higher t value enforces stricter access control and may reduce the number of successful matches. In our experiments, the scheme continued to return correct keyword search results under different threshold settings, demonstrating stable performance and consistent practicality. The reason for using only the Test algorithm for benchmarking is that the Test algorithm is run multiple times, unlike KeyGen algorithms that are only run once. The Encrypt algorithm and Trapdoor algorithm are also run multiple times but the results show no significant changes with respect to different thresholds.

The analysis results are presented in Figure 1 illustrating these results in a chart format. Experimental findings show that the execution time of the Test algorithm grows approximately linearly as the threshold parameter increases. There are a total of 667 emails associated with the keyword "forwarded". When the threshold value is set to 2, the average time to test a single email is 41.36 ms. Conversely, when the threshold value is set to 12, the average time to test a single email increases to 190.04 ms. This indicates that the proposed CP-DABKS scheme, using Type-3 pairings, is efficient and suitable for practical implementation. Optimisation of the scheme's performance is reserved for future work to enhance its practical applicability, specifically to reduce the time required for the Test algorithm and incorporate other search functionality.

Table 1. Word Frequency.

Word	Frequency
Forwarded	362
I	314
Hey	138
Please	109
Can	94
Andy	80
You	74
Hello	66
How	58
Thanks	58
Just	54
Are	50
Dave	50
What	48
Jeff	48

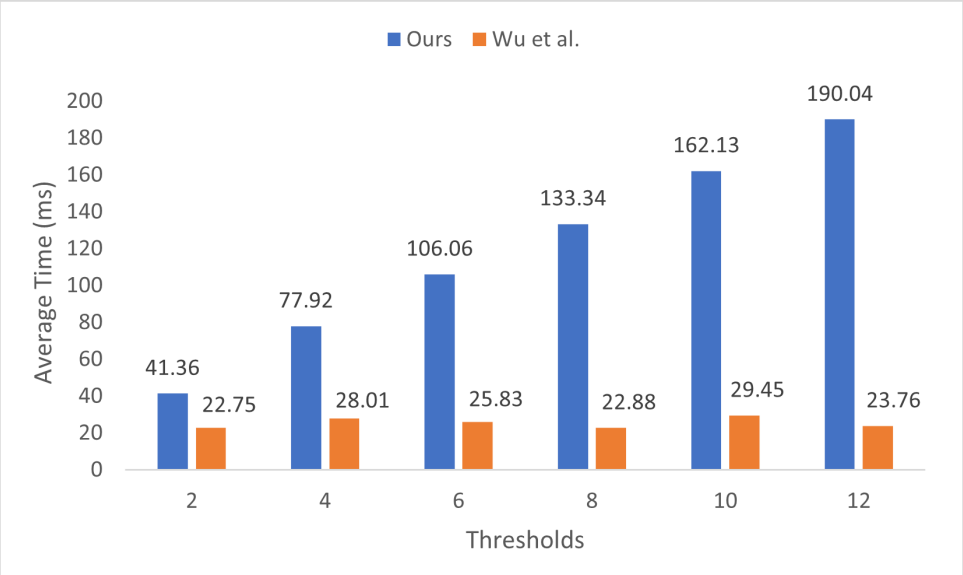

Figure 1. Performance of *Test* Algorithm and Wu et al.'s scheme [33] Under Different Thresholds.

For comparison, Wu et al.'s multi-authority ABKS scheme [33] demonstrates shorter Test execution times across all threshold values, with averages ranging between 22.75 ms and 29.45 ms. This difference, however, stems from the underlying design scope of the schemes. In Wu et al.'s construction, the threshold value corresponds to the minimum number of attributes required under a single authority, whereas the proposed CP-DABKS defines the threshold across a global multi-authority access structure. Hence, the proposed scheme performs additional pairing and policy-evaluation operations per test, resulting in higher computation time. Nevertheless, this overhead is expected and acceptable given that the proposed scheme supports cross-authority coordination and fine-grained attribute control that Wu et al.'s model does not address. Furthermore, by employing Type-3 pairings and a commit-to-point mechanism, the proposed scheme achieves stronger security

assurances and scalability, demonstrating a balanced trade-off between expressiveness and efficiency suitable for real-world deployment.

In Table 2, a comparison is made between the communication costs of the Type-3 pairings CP-DABKS scheme, Wu et al.'s ABKS scheme [33], and the Type-1 pairings CP-DABKS scheme proposed by Guo et al. [5]. Let $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, $|\mathbb{G}_T|$, and $|\mathbb{Z}_p|$ denote the size of the elements in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, and $\mathbb{Z}_p$, respectively. Here, U denotes the complete attribute universe, A represents the subset assigned to a user, and l corresponds to the number of rows in the access-policy matrix, under the assumption that point compression is applied. If Guo et al.'s scheme [5] uses a Type A curve [26], then the size of $|\mathbb{Z}_p|$ is 768 bits, $|\mathbb{G}_1| = |\mathbb{G}_2| = 1537$ bits, and $|\mathbb{G}_T| = 3074$ bits. For the proposed Type-3 pairings CP-DABKS scheme that uses the BLS12-381 curve, the size of $|\mathbb{Z}_p|$ is 255 bits, $|\mathbb{G}_1| = 382$ bits, $|\mathbb{G}_2| = 764$ bits, and $|\mathbb{G}_T| = 4584$ bits. In comparison to the constructions of Guo et al. [5] and Wu et al. [33], our CP-DABKS scheme exhibits reduced overall communication overhead in its public parameters (cp), key pairs, and ciphertexts. Guo et al.'s cp size is 9216 bits plus 1536 bits multiplied by the size of the attribute universe $|U|$, whereas Wu et al.'s is 6916 bits plus $1537|U|$ bits. In contrast, the proposed scheme's cp size is fixed at 9886 bits, independent of $|U|$. Consequently, even with a small attribute universe, both Guo et al. and Wu et al.'s schemes result in larger parameters. Furthermore, thanks to the secure amendment of the commit-to-point operation in our CP-DABKS scheme, the system is no longer constrained by a predefined attribute universe U, providing better scalability and practical efficiency.

Table 2. Communication Efficiency Comparison.

Scheme	cp	Sender/ Cloud Server Key Pair	Receiver Key Pair	Ciphertext	Trapdoor
Guo et al. (2020) [5]	$(4+\|U\|)\|G_1\|+\|G_T\|$	$\|G_1\|+\|Z_p\|$	$(3+\|A\|)\|G_1\|+\|Z_p\|$	$(4+2l)\|G_1\|+2\|G_T\|$	$\|G_T\|+2\|G_1\|$
	9222 + $1537\|U\|$ bits	4611 bits	5379 + $1537\|A\|$ bits	12,296 + $3074l$ bits	6148 bits
Wu et al. (2022) [33]	$(2+\|U\|)\|G_1\|+\|G_T\|+\|Z_p\|$	$\|G_1\|+\|Z_p\|$	$(3+2\|A\|)\|G_1\|+\|Z_p\|$	$(2+l)\|G_1\|+\|G_T\|$	$\|G_T\|$
	6916 + $1537\|U\|$ bits	4611 bits	5379 + $3074\|A\|$ bits	6148 + $1537l$ bits	3074 bits
The Proposed Scheme	$5\|G_1\|+3\|G_2\|+\|G_T\|$	$\|G_2\|+\|Z_p\|$	$(3+\|A\|)\|G_2\|+\|Z_p\|$	$(1+2l)\|G_1\|+3\|G_2\|+2\|G_T\|$	$\|G_T\|+2\|G_1\|$
	9886 bits	1019 bits	2547 + $764\|A\|$ bits	11,842 + $764l$ bits	5348 bits

Overall, when comparing the three schemes side-by-side, distinct design trade-offs can be observed. Guo et al.'s [5] Type-1 pairing construction achieves acceptable efficiency but suffers from linear growth of public parameters with the attribute universe size $|U|$. Wu et al.'s [33] multi-authority design offers lower computation cost but lacks scalability and dynamic attribute support. By contrast, the proposed Type-3 pairing CP-DABKS scheme combines compact public parameters with resistance to KGA and channel-free trapdoor transmission, making it more suitable for large-scale and frequently updated attribute environments.

7.1. Trade-Offs in Applicable Scenarios

The experimental and analytical results highlight a balance between scalability, security, and computational efficiency. The use of Type-3 pairings provides faster pairing operations and smaller element sizes compared with Type-1 settings, directly reducing communication and storage overhead. Meanwhile, the commit-to-point mechanism enables dynamic attribute expansion without regenerating global parameters, improving long-term flexibility at the cost of slightly higher setup complexity. Therefore, the proposed CP-DABKS scheme is particularly advantageous for cloud systems that require continuous

attribute updates and KGA resistance, whereas Type-1 or single-authority ABKS variants remain preferable in lightweight or static environments.

7.2. Limitations and Reproducibility Considerations

This study evaluates the proposed scheme using the Enron Email Dataset [34], a widely used public corpus that reflects realistic message formats and keyword distributions in enterprise communication. Although only one dataset was employed, the experimental results remain representative, since the main performance factors, such as pairing operations, exponentiations, and access-policy evaluations, are determined by the cryptographic design rather than the textual content. To further validate reproducibility, future evaluations will extend to the Avocado Research Email Collection [35] under the same parameter settings and threshold variations. We expect similar linear growth trends in testing time as a function of the threshold parameter t, since the computational workload scales with the number of pairing and exponentiation operations rather than dataset characteristics. These results suggest that the proposed scheme's practicality and scalability are independent of a particular dataset.

8. Future Work

In future work, we intend to implement and evaluate the CP-DABKS scheme within practical cloud infrastructures through integration with the AWS Java SDK [36]. In particular, we aim to utilise the `com.amazonaws.auth.policy` package, which provides APIs for defining and evaluating access conditions based on AWS attributes. This integration serves as a first step toward deploying CP-DABKS within AWS-based access control infrastructures.

We also highlight a key advantage of our scheme over Guo et al.'s original CP-DABKS construction. In Guo et al.'s approach, the MPK must include a unique pair of group elements for each attribute in the universe. For large-scale systems such as AWS IAM, which supports dozens of attributes per user and potentially tens of thousands system-wide, the MPK grows linearly with the number of attributes. This results in significant storage and distribution overhead. By contrast, the proposed commit-to-point mechanism enables attribute-dependent components to be generated dynamically at runtime, ensuring that the MPK size remains fixed and independent of the attribute universe dimension. This property makes our scheme more suitable for deployment in practical environments with dynamic and large-scale attribute sets.

Another direction involves extending the scheme's expressiveness beyond a basic (l, n)-threshold access structure. In the current design, the access policy (M, ρ) defined during *Encrypt* must be satisfied during *Test* before the cloud server is able to execute keyword search. However, this structure only supports a single-layer threshold policy. Future work will explore recursive composition of threshold policies to realise more general monotonic access structures. For a structure with N levels, the ciphertext would require N instances of the element pairs $(C'_i, D'_i)_{i=1}^{l}$, while the *Test* algorithm would compute the token Q_t recursively using a distinct set of constants w_i for each level.

We also intend to explore performance optimisations, particularly for the *Test* algorithm, which remains the most computationally intensive component in our implementation. By reducing its runtime, we aim to enhance the overall practicality of the scheme for large-scale applications.

9. Conclusions

This paper presented a secure and efficient CP-DABKS scheme with dynamic attribute universe. The proposed scheme mitigates keyword guessing attacks while eliminating

the requirement for a secure communication channel when transmitting trapdoors. While inspired by the framework of Guo et al. [5], our construction introduces significant enhancements by leveraging the commit-to-point mechanism to support dynamic attribute universes, as well as the asymmetry and performance benefits of Type-3 pairings.

The proposed construction was implemented and benchmarked using the Enron Email dataset. Experimental results confirm its practicality and robustness for cloud environments that demand fine-grained access control with efficient keyword search capabilities. We also outlined several promising directions for future development, including real-world integration with AWS infrastructure and the expansion of the access structure to support recursive threshold policies.

Author Contributions: Conceptualization, K.-M.C., S.-H.H. and S.-Y.T.; formal analysis, S.-H.H., S.-Y.T. and S.-C.T.; funding acquisition, S.-H.H.; investigation, K.-M.C. and S.-Y.T.; methodology, K.-M.C. and S.-Y.T.; validation, S.-H.H., S.-Y.T. and S.-C.T.; writing—original draft preparation, K.-M.C.; writing—review and editing, S.-H.H., S.-Y.T. and S.-C.T.; supervision, S.-H.H., S.-Y.T. and S.-C.T.; project administration, S.-H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Telekom Malaysia Research & Development Grant (RDTC/221045).

Data Availability Statement: The original contributions presented in this study are included in the article.

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

References

1. Byun, J.W.; Rhee, H.S.; Park, H.A.; Lee, D.H. Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In *Workshop on Secure Data Management*; Springer: Berlin/Heidelberg, Germnay, 2006; pp. 75–83.
2. Yau, W.C.; Heng, S.H.; Goi, B.M. Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In *International Conference on Autonomic and Trusted Computing*; Springer: Berlin/Heidelberg, Germnay, 2008; pp. 100–105.
3. Yang, K.; Jia, X.; Ren, K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems. In Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, Hangzhou, China, 8–10 May 2013; pp. 523–528.
4. Sun, W.; Yu, S.; Lou, W.; Hou, Y.T.; Li, H. Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *27*, 1187–1198. [CrossRef]
5. Guo, L.; Li, Z.; Yau, W.C.; Tan, S.Y. A decryptable attribute-based keyword search scheme on eHealth cloud in Internet of Things platforms. *IEEE Access* **2020**, *8*, 26107–26118. [CrossRef]
6. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
7. Galbraith, S.D.; Paterson, K.G.; Smart, N.P. Pairings for cryptographers. *Discret. Appl. Math.* **2008**, *156*, 3113–3121. [CrossRef]
8. Hwang, Y.H.; Lee, P.J. Public key encryption with conjunctive keyword search and its extension to a multi-user system. In *International Conference on Pairing-Based Cryptography*; Springer: Berlin/Heidelberg, Germnay, 2007; pp. 2–22.
9. Wang, C.; Li, W.; Li, Y.; Xu, X. A ciphertext-policy attribute-based encryption scheme supporting keyword search function. In Proceedings of the Cyberspace Safety and Security: 5th International Symposium, CSS 2013, Zhangjiajie, China, 13–15 November 2013; Proceedings 5; Springer: Berlin/Heidelberg, Germnay, 2013; pp. 377–386.
10. Miao, Y.; Ma, J.; Liu, X.; Wei, F.; Liu, Z.; Wang, X.A. m^2-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting. *J. Med Syst.* **2016**, *40*, 246. [CrossRef] [PubMed]
11. Miao, Y.; Ma, J.; Liu, X.; Li, X.; Jiang, Q.; Zhang, J. Attribute-based keyword search over hierarchical data in cloud computing. *IEEE Trans. Serv. Comput.* **2017**, *13*, 985–998. [CrossRef]
12. He, H.; Zhang, J.; Li, P.; Jin, Y.; Zhang, T. A lightweight secure conjunctive keyword search scheme in hybrid cloud. *Future Gener. Comput. Syst.* **2019**, *93*, 727–736. [CrossRef]
13. Yin, H.; Zhang, J.; Xiong, Y.; Ou, L.; Li, F.; Liao, S.; Li, K. CP-ABSE: A ciphertext-policy attribute-based searchable encryption scheme. *IEEE Access* **2019**, *7*, 5682–5694. [CrossRef]

14. Cao, M.; Wang, L.; Qin, Z.; Lou, C. A Lightweight Fine-Grained Search Scheme over Encrypted Data in Cloud-Assisted Wireless Body Area Networks. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 9340808. [CrossRef]
15. Cui, J.; Zhou, H.; Xu, Y.; Zhong, H. OOABKS: Online/offline attribute-based encryption for keyword search in mobile cloud. *Inf. Sci.* **2019**, *489*, 63–77. [CrossRef]
16. Miao, Y.; Liu, X.; Choo, K.K.R.; Deng, R.H.; Li, J.; Li, H.; Ma, J. Privacy-preserving attribute-based keyword search in shared multi-owner setting. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 1080–1094. [CrossRef]
17. Sun, J.; Xiong, H.; Nie, X.; Zhang, Y.; Wu, P. On the security of privacy-preserving attribute-based keyword search in shared multi-owner setting. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 2518–2519. [CrossRef]
18. Pan, X.; Li, F. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. *J. Syst. Archit.* **2021**, *115*, 102075. [CrossRef]
19. Zhang, K.; Li, Y.; Lu, L. Privacy-Preserving Attribute-Based Keyword Search with Traceability and Revocation for Cloud-Assisted IoT. *Secur. Commun. Netw.* **2021**, *2021*, 9929663. [CrossRef]
20. Yan, L.; Wang, G.; Yin, T.; Liu, P.; Feng, H.; Zhang, W.; Hu, H.; Pan, F. Attribute-based searchable encryption: A survey. *Electronics* **2024**, *13*, 1621. [CrossRef]
21. Tang, Y.; Chen, Y.; Luo, Y.; Dong, S.; Li, T. VR-PEKS: A verifiable and resistant to keyword guess attack public key encryption with keyword search scheme. *Appl. Sci.* **2023**, *13*, 4166. [CrossRef]
22. Li, X.; Wang, H.; Ma, S.; Xiao, M.; Huang, Q. Revocable and verifiable weighted attribute-based encryption with collaborative access for electronic health record in cloud. *Cybersecurity* **2024**, *7*, 18. [CrossRef]
23. Khan, S.; Khan, S.; Waheed, A.; Mehmood, G.; Zareei, M.; Alanazi, F. An optimized dynamic attribute-based searchable encryption scheme. *PLoS ONE* **2024**, *19*, e0268803. [CrossRef] [PubMed]
24. Meng, L.; Chen, L.; Tian, Y.; Manulis, M.; Liu, S. FEASE: Fast and Expressive Asymmetric Searchable Encryption. In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), Philadelphia, PA, USA, 14–16 August 2024; pp. 2545–2562.
25. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
26. Ben, L. On The Implementation On Pairing-Based Cryptosystems. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2007.
27. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Proceedings 24; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
28. Lewko, A.; Okamoto, T.; Sahai, A.; Takashima, K.; Waters, B. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Proceedings of the Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, France, 30 May–3 June 2010; Proceedings 29; Springer: Berlin/Heidelberg, Germany, 2010; pp. 62–91.
29. Pedersen, T.P. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 129–140.
30. Menezes, A.J.; Van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996; Volume 17.
31. Krawczyk, H.; Rabin, T. Chameleon hashing and signatures. *Cryptol. Eprint Arch.* **1998**. Available online: https://eprint.iacr.org/1998/010 (accessed on 28 August 2025).
32. Abe, M.; Hoshino, F.; Ohkubo, M. Design in type-i, run in type-iii: Fast and scalable bilinear-type conversion using integer programming. In Proceedings of the Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2016; Proceedings, Part III 36; Springer: Berlin/Heidelberg, Germany, 2016; pp. 387–415.
33. Wu, Y.; Li, X.; Liu, Z. Attribute-based keyword searchable encryption scheme for multi-authority in cloud storage. In Proceedings of the 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, 11–14 November 2022; pp. 933–939.
34. Klimt, B.; Yang, Y. The Enron corpus: A new dataset for email classification research. In Proceedings of the 15th European Conference on Machine Learning (ECML 2004), Pisa, Italy, 20–24 September 2004; Springer: Berlin/Heidelberg, Germany, 2004.
35. Oard, D.; Webber, W.; Kirsch, D.A.; Golitsynskiy, S. *Avocado Research Email Collection*; LDC2015T03; Web Download; Linguistic Data Consortium: Philadelphia, PA, USA, 2015. [CrossRef]
36. Amazon Web Services, Inc. AWS SDK for Java. 2023. Available online: https://aws.amazon.com/sdk-for-java/ (accessed on 12 October 2023) .