International Journal of Communication Networks and Information Security
ISSN: 2073-607X, 2076-0930
Volume 15 Issue 04 Year 2023

Study on A Proposed Scheme for Generating Inverted Encryption Index Structure Based on Public Homomorphic Encryption

Ahmed Nashaat Shakir*
Department of Information Technology, College of Computer Science and Information Technology, Kirkuk University, Kirkuk, Iraq
ahna2005@uokirkuk.edu.iq
Zubaidah Abdulhakeem Majeed
Department of Computer Science, College of Computer Science and Information Technology, Kirkuk University, Kirkuk, Iraq
zubaidah.abdulhakeem@uokirkuk.edu.iq

Article History

Received: 1 August 2023
Revised: 18 September 2023
Accepted: 16 October 2023

Abstract

This research article focuses on the formidable challenge of efficiently searching through encrypted data in cloud environments, particularly as an extended number of users adopt encryption for their sensitive Information. The inverted index has proven to be a robust and effective searchable index structure in this context. However, striking a balance between preserving user privacy and enabling conjunctive multi-keyword searches remains a significant hurdle for existing solutions. In response to this challenge, the authors propose an innovative public-key-based encrypted file system. This system follows conjunctive multi-keyword searches but also eliminates the restrictive one-time-only searching limitation that has been a drawback in previous approaches. The proposed solution goes beyond conventional methods by safeguarding the search pattern, a critical aspect of user privacy. Their approach involves the integration of a probabilistic trapdoor-generating mechanism, adding an extra layer of security. To fortify their technique and adhere to more stringent security standards, the authors introduce an oblivious transmission control mechanism. This mechanism enhances the overall security posture of the system, ensuring robust protection against potential threats. The simulation results presented in the article demonstrate the practical proposed technique in real-world applications. Despite the additional security measures, the approach incurs reasonable overhead, making it a viable and efficient solution for cloud-based encrypted data searches.

Keywords: Data Privacy, Multi-user Environments,Homomorphic Encryption, Indexing, Secure Information Retrieval

1. Introduction

In the landscape of cloud computing, the utilization of cloud storage services for storing and accessing personal information has seen a significant upswing. However, to safeguard data privacy, the imperative of encrypting sensitive data before transmission to the cloud has become increasingly apparent. While data encryption effectively shields data from unauthorized access, it introduces challenges in performing activities such as searching for specific keywords within the encrypted

data. The conventional method of downloading the entire set of encrypted files and subsequently decrypting them proves to be an unproductive utilization of information and communication resources. Thus, there arises a critical need for a search method that seamlessly integrates cloud services and encryption keys [1].

Existing techniques for enabling direct search over encrypted data, commonly known as searchable encryption, employ a secure index for the entire document collection. This secure index, akin to searching plaintext documents, is a standard procedure in searchable encryption systems. During the search stage, the secure index is invoked on the cloud server side. The utilization of secured index-based data encryption ensures the independence of each document's encryption, thereby enhancing search scalability [2].

However, prevalent searchable encryption systems are predominantly based on either inverted indices or self-designed indexes. While inverted indices offer advantages over self-designed indexes, current systems still grapple with limitations. Searches using inverted indices can be highly effective, particularly for large datasets, as the inversion lists matching the query term guide the search directly to related documents. Despite this potential, existing schemes encounter drawbacks such as compromised privacy during keyword searches and the need for index rebuilding post-search. Furthermore, most current schemes lack support for conjunctive multi-keyword searches, a common and resource-intensive query type [3].

To address these limitations comprehensively, we propose an innovative public key-based encrypted file system that incorporates attribute values. This approach strategically combines the robust search performance of the inverted index while eliminating the one-time-only searching restriction inherent in prior solutions. In our system, the search pattern is protected, and a probabilistic trapdoor-generating mechanism is employed to facilitate conjunctive multi-keyword searches. Unlike methods reliant on expensive pairing processes, our public key-based approach employs efficient operations such as multiplications and exponentiations, rendering it more efficient. Additionally, we introduce an effective oblivious transmission control mechanism to conceal network operations from both the cloud and the network, meeting stricter security standards. Simulation results affirm the practicality and efficiency of our technique for real-world applications, demonstrating reasonable overhead [4].

While Curtmola et al. pioneered the concept of a secure inverted index for searchable encryption, current inverted index-based systems grapple with two primary drawbacks. Firstly, privacy is compromised during keyword searches, necessitating index rebuilding post-search. Secondly, most existing schemes lack support for conjunctive multi-keyword searches. In its reaction, we offer an inverted index-based searchable encryption system with a secure set-crossing mechanism [5]. Our system ensures private and secure matching between the secure index and the query trapdoor. We have devised a novel trapdoor-generating mechanism to covertly combine query-related inverted lists without disclosing the fetched inverted list to the cloud server [6].

Building upon the foundation laid by existing searchable encryption methodologies, our proposed system represents a leap forward in addressing the inherent challenges and limitations. By strategically combining the efficiency of inverted indices with a public key-based encrypted file system, we pave the way for enhanced search scalability, privacy preservation, and support for conjunctive multi-keyword searches. In doing so, our approach not only overcomes the shortcomings of current systems but also sets a new standard for the secure and efficient exploration of encrypted data within cloud environments. The seamless integration of attribute values in our public key-based encrypted file system offers a novel dimension to the search process, providing a practical solution that goes beyond the constraints of previous one-time-only searching restrictions. Moreover, the incorporation of a probabilistic trapdoor-generating mechanism ensures a robust defense against privacy compromises during keyword searches. Leveraging efficient operations such as multiplications and exponentiations further distinguishes our approach by minimizing computational overhead.

In addition to these advancements, our system introduces an oblivious transmission control mechanism, a critical innovation that conceals network operations from both the cloud and the network. This mechanism not only fortifies the security standards but also aligns with the evolving landscape of cloud-based data storage and retrieval. Simulation results validate the viability of our technique in real-world applications, showcasing not only its practicality but also its efficiency with

reasonable overhead. The proposed searchable encryption scheme based on an inverted index, enriched by a private set intersection protocol, addresses the fundamental drawbacks of current inverted index-based systems. By ensuring private and secure matching between the secure index and the query trapdoor, our system offers a level of security unparalleled in the current landscape. The novel trapdoor-generating mechanism covertly combines query-related inverted lists, a key innovation that further enhances the privacy and efficiency of the search process.

In essence, our work contributes a comprehensive solution to the challenges of searching through encrypted data in the cloud, promising enhanced security, efficiency, and privacy preservation in comparison to existing methodologies.

Our contributions include:

(1) This text introduces an innovative public-key searchable encryption solution that utilizes an inverted index, effectively overcoming the limitation of one-time-only searches present in earlier methods. In contrast to existing inverted index-based techniques, which only allow single-keyword searches, our system facilitates conjunctive multi-keyword searches through a single trapdoor.

(2) We have developed a probabilistic trapdoor creation mechanism that eliminates the linkability of trapdoors while retaining indexing and trapdoor privacy. Our system also includes an effective oblivious interchange format to conceal the access pattern, which enhances security [7].

(3) Our technique is much more efficient than current public-key searchable cryptographic algorithms as it only requires multiplication and exponentiation, which are inexpensive pairing operations [8].

2. Related Works

In the article [9], An innovative Privacy-Preserving Searchable Encryption (PPSE) technique is presented, utilizing both public and private blockchains to tackle the difficulties associated with cloud data outsourcing. The proposed solution minimizes storage needs, boosts transaction speed, and enhances data protection by storing encrypted indices in a private blockchain while outsourcing associated documents to a public blockchain. A smart contract enforces a secondary verification access control, restricting user access to the private blockchain for privacy. This system improves both the security of encrypted data and the efficiency of queries, outperforming current solutions in both aspects.

The article [10] describes a PEKS scheme with post-quantum security, robust against attacks, supporting conjunctive searches, and implemented efficiently in C++, completing tasks in 22.5 milliseconds at a 256-bit security level with a compact 5-kilobyte public key.

In this paper [11], in era of increased data storage and cloud reliance, a dynamic multi-keyword ranked search method ensures secure searches of encrypted cloud data with access control and efficient B+tree indexing. Assumed 'Honest-but-Curious' cloud server, Merkle B-tree verifies results, validated by real-world experiments for efficiency and precision.

The paper [12] examines the evolving landscape of data privacy in cloud infrastructures, emphasizing the role of encryption techniques. Traditional encryption methods pose challenges for efficient data retrieval, leading to innovative solutions like integrating Homomorphic Encryption (HE) into Searchable Encryption (SE) schemes. The analysis categorizes existing HE-based SE schemes, explores HE types used in SE and outlines their impact on search processes and additional functionalities. Notably, Partially Homomorphic Encryption is widely adopted. The study underscores the prevalence of index-based SE schemes, support for ranked search, and multi-keyword queries, and suggests future research directions, including integrating other encryption schemes, addressing functional gaps, and leveraging advancements in Fully Homomorphic Encryption.

The paper [13] explores the impact of cloud infrastructures on data storage and access, emphasizing concerns over the privacy of sensitive information. Encryption techniques, though widely used, present challenges for efficient data retrieval. Innovative solutions, such as integrating Homomorphic Encryption (HE) into Searchable Encryption (SE) schemes, address these limitations. The analysis categorizes existing HE-based SE schemes, delves into HE types used in SE, examines their influence on search processes and additional functionalities, and outlines future research directions. Notably, there's a rise in HE adoption, particularly Partially Homomorphic Encryption,

and the prevalence of index-based SE schemes supporting ranked search and multi-keyword queries. Future exploration areas include integrating other encryption schemes, addressing functional gaps like fuzzy keyword search, and leveraging advancements in Fully Homomorphic Encryption.

The paper [14], this algorithm tackles the efficiency challenge in searchable encryption for queries across multiple data owners. Unlike existing schemes for single data owner scenarios, it introduces a method based on key aggregation, ensuring consistent ciphertext processing. By mutually inverting elements in a finite field, it generates index keys and facilitates efficient query trapdoor generation for encrypted indexes from different owners. Theoretical analysis and simulations show enhanced query efficiency and protection of query indexes and trapdoors.

3. Methodology

The following text describes the DGHV homomorphic data encryption, the DGHV homomorphic encryption scheme, named after its creators Brakerski, Vaikuntanathan, and Vaikuntanathan, to ensure secure computation on encrypted data while maintaining confidentiality throughout the processing pipeline. This method is used to encrypt data in a way that allows computations to be performed on the encrypted data without needing to decrypt it first. The security of the encryption scheme is based on the (ρ, η, γ)-Approximate- Greatest Common Devisor (GCD) problem, which involves finding the greatest common divisor of two numbers with some approximation. The parameter generation method accepts a security parameter as input and calculates various sets of parameters, including the shared key parameters. The key distribution procedure involves choosing an odd-bit integer p and drawing samples from D(p), which is a set of possible remainders when dividing p by certain values. The resulting public key and secret key are used to encrypt and decrypt data, respectively. To ensure the security of the encryption scheme, a homomorphic secure hash algorithm is used, which satisfies the homomorphic and collision-free properties. This algorithm allows for computations to be performed on encrypted vectors while preventing collisions between different vectors.

Let p be an odd-bit integer, D be a set of polynomially many examples, and q and r be random integers. Then, we can define:

(1) γ as the value of O(λ^5)

(2) ρ as a parameter for the (ρ, η, γ)-Approximate-GCD problem

(3) η as another parameter for the (ρ, η, γ)-Approximate-GCD problem

(4) x as the output of pq + r, where p is a random -bit uneven number

(5) The parameter generation method ParamGen takes a security parameter λ as input and calculates the following sets of values:

(6) λ

(7) 2

(8) O(2)

(9) O(5)

(10) +

The shared key parameters are then outputted as h, 0, and i. The key distribution procedure KeyGen(λ) starts by choosing the odd-bit integer p, which has the value p = (value not specified in the text). It then draws (+1) samples x0,..., x from D,(p), and relabels them. Finally, the public key is pk = hx0, x1,..., x I and the secret key is sk = p.

The encryption algorithm Encrypt(pk, m) selects a random subset S to encrypt a bit m among 0 and 1. Let G be an order p carry a greater group, and generators (g1, g2,..., gn) exist. The homomorphic secure hash algorithm of something like a vector b = (b1, b2,... bn) is given by:

$$sH(b) = sQn, si=1\ sg\ sbi\ si\ s \qquad (1)$$

Then, H(b) meets the requirement stated in [9]:

H(r1b1 + r2b2) = H(b1) r1H(b2) r2 for any two vectors b1, b2, and random integers r1, r2.

It is also collision-free, meaning that it is hard to find b1, b2, b3, r1, and r2 (where b3 ≠ r1b1 + r2b2) for any polynomial-time algorithm which satisfies H(b3) = H(b1) r1H(b2) r2.

3.1 System Model

The proposed system comprises four key components: the key-generation hub, data consumers, cloud services, and data owners.

(1) Key-generation hub: The key-generation center (KGC) is a fully trusted facility responsible for generating public and distributing private keys to individuals, cloud servers, and data owners.

(2) Data owners: This group creates and encrypts private/confidential data before sharing it with authorized parties via the cloud.

(3) Cloud server: With large storage capacity and computational capabilities, cloud servers are ideal for hosting the encrypted data. We assume the cloud server is "honest but curious," meaning it runs the method truthfully while attempting to analyze the data in an interesting way.

(4) In this scheme, we use the "Frequently Used Ciphertext Model" to protect the keywords from the cloud server. The server can only access the ciphertexts, while the data consumers can perform keyword searches on the ciphertexts via spot instances. The server creates a hash table using the ciphertexts, but it cannot access the user's keywords since it does not have the user's secret key [8].

The system comprises four main components: the key-generation hub (KGC), data consumers, cloud services, and data users. The KGC is a fully trusted facility tasked with generating and distributing public and private keys to individuals, cloud servers, and data owners.

Data owners encrypt private data using a standard proxy re-encryption mechanism and employ the Public-Key Encryption with Keyword Search (PEKS) method for each keyword (wi, pk). The encrypted data is then sent to the cloud server in the form of E(file)||PEKS(w1, pk)|| . . . ||PEKS(wm, pk).

To enable keyword searches on the ciphertexts without revealing the user's secret key, the server creates a hash table using the ciphertexts but is unable to access the user's keywords. This method relies on a verified public-key encryption technique with a keyword search in a multi-user climate. The technique consists of several algorithms, including Setup(1), KeyGen(1, id), PEKS(wi, pk), and Verify(Rf, proof).

In a multi-user setting, a public-key authentication mechanism is combined with a specific keyword using the DGHV homomorphic encryption technique. The key-generation component performs ParamGen0 to obtain its shared key params0 and KeyGen0 to generate sk0 = p and pk0 = hx0, x1. The KGC then chooses a homomorphic hashing H and a concussion hash H1: 0, 1s 0, 1sQ(Q). The KGC transfers the secret key sk = sk0 = p to the servers while establishing the decryption key pk = (params0, pk0, H, H1) too though.

KeyGen (id, 1). If a user

$$Ci1 = [w_i + r_i q_i d + w_i + r_i q_i d \, X \, i \in S \, x_i]x_0, c_{i2} = (r_i g_i d) \qquad (2)$$

The searchable ciphertexts for the term wi are as follows:

$$C_{Ti} = (C_{i1}, C_{i2})(C_{i1}, C_{i2})$$

Test(CTi, CTj) (CTi, CTj). The server checks to see if the two ciphertexts (CTi, CTj) are compatible after receiving them.

$$H(Ci1 \bmod p) \times Cj2 \; H(Cj1 \bmod p) \times Ci2 = 1 \qquad (3)$$

It returns 1 if wi = wj; else, it returns 0.

Table 1. Showing the Encryption of Files with Corresponding Keyword Searchable Ciphertexts using PEKS in a Multi-user Setting

PEKS (w_1)	**E($file_1$)**	**E($file_2$)**			
PEKS (w_2)	E($file_3$)	E($file_4$)	E($file_s$)		
PEKS (w_3)	E($file_1$)	E($file_2$)	E($file_3$)	E($file_4$)	E($file_5$)
…	…	…	…	…	…
PEKS (w_{m-2})	E($file_6$)	E($file_7$)	E($file_{10}$)		
PEKS (w_{m-1})	E($file_2$)	E($file_3$)	E($file_s$)	E($file_8$)	E($file_9$)
PEKS (w_m)	E($file_1$)	E($file_2$)	E($file_{10}$)		

CTi(i = 1, 2,.., mZ-IndexBuild)'s function. The server uses the Test(CTi,CTj) algorithm with the ciphertexts CTi(i = 1, 2,..., m) to check whether the keywords in the two ciphertexts (CTi,CTj) are consistent, where PEKS(wi) = CTi.

3.2 A Scheme for Authorized Data Structures Based on Inverted Encryption

3.2.1 Data Structure

Description of the data structure used in the scheme

Rows represent encrypted phrases PEKS(wi) = CTi

Columns represent encrypted files E(filei)

Each row contains a verification proof pre-proof (vwi) and vector vwi = [vi1, vi2,..., vin]

Set vij = 1 if the term wi is present in the E(filej), otherwise set vij to 0

3.2.2 Verification Process

Description of the verification process used in the scheme

Use of pre-proof (vwi) and vector vwi to locate keywords in encrypted files

Creation of vector v0 using Rf = E(filei)

Verification that search result is exhaustive: v0 = binary(value(vw1)) &... & binary(value(vwl))

Output 1 if the equation is true, 0 if not

3.2.3 Example

An example to illustrate the verification process

Use of Test(CTi, CTj) algorithm to discover associated PEKS(w1, w2)

Verification using equations H(value(vw1)) = prepoof (w1) and H(value(vw2)) = prepoof (w2)

Construction of v0 and verification of completeness condition

3.2.4 Scheme Provisions

Explanation of subset constraints and completeness conditions

Subset requirement satisfied by verification of equation (3)

Explanation of how ciphertexts containing the same keyword are handled

Use of equation (7) to demonstrate how to handle cases where wi = wj

3.2.4.1 Search Result Validation

To ensure completeness in our system, the server only provides encrypted files for which the relevant locations are identical to 1, as described in [22]. Equations (3) and (4) guarantee the rectification of the search result (4), while Equations (5) and (6) allow users to verify its accuracy, Equation (5) publishes the pre-proof (wi), which, if true, suggests that the returned value (vwi) is also true.

4. Results and Discussion

4.1 Security Analysis

The security of estimating large integers is a challenging task. One approach is to use a set of randomly chosen integers (x0, x1,..., x) to identify the "common nearest divisor" of multiple copies of a large integer, p. To achieve this, an accurate and trustworthy oracle is needed, which can be obtained through a spontaneous and accuracy-amplification stage. The BinaryGCD method can then use this oracle to determine p [18].

To reduce the parameters involved, our random self-reduction technique can correctly anticipate the redundant bits of a number, such as the quotient in a "low-bits noise integer," to guess an encrypting bit in a randomized "strong s0 s-bits noisy ciphertext" [24]. We can increase the noise sound in the random integers to eliminate any non-randomness, adjusting the approach's security near the difficulty of the approximate-GCD problem [19].

Theorem 1 shows that the suggested system to solve the approximation GCD problem can be used to create an advantageous attack A against algorithm B, with a success chance of at least 2/1.

Setting the parameters (, 0,,,) as suggested in Section 4, the running durations of B and A are both 1/ and are exponential with respect to one another. The proof is shown by demonstrating how well p can be recovered by challenger B using the same method as [25].

Z is defined as z = qp(z)p + rp using qp(z) but instead rp(z) to represent the quotient as well as remaining portion of z with regard to p. (z).

4.2 Algorithm 1

The Learn-LSB subroutine is a key component of Algorithm 1, which is used in cryptography for a variety of applications. In this subroutine, the input is a parameter z that is between 0 and 2, along with a public key pk consisting of a series of randomly chosen integers x0, x1,..., xn. The algorithm performs the following steps for each value of j:

(1) Choose a random noise value rj, a bit value wj, and a random subset of indices Sj.

(2) Calculate a new value CTj as z + wj + rj + (rj6kSj xk).

(3) Invoke the A function on the public key pk and the value CTj, resulting in a value aj.

(4) Set values aj, bj, parity(z), and wj.

(5) Repeat steps 1-4 for each value of j.

(6) Take the majority vote of the bj values and publish the result.

Once the above steps are completed, the next step is to recover the value of p. This can be accomplished by transforming the subroutine into an oracle for the least significant bit of qp(z) and using the Binary GCD Algorithm. Specifically, the Binary GCD Algorithm is applied to two integers sz1 and sz2, where sz1 is the value of qp(z) for the current iteration and sz2 is a previous value of qp(z) that has been transformed into an oracle. This method is highly effective and has been widely used in cryptography for many years.

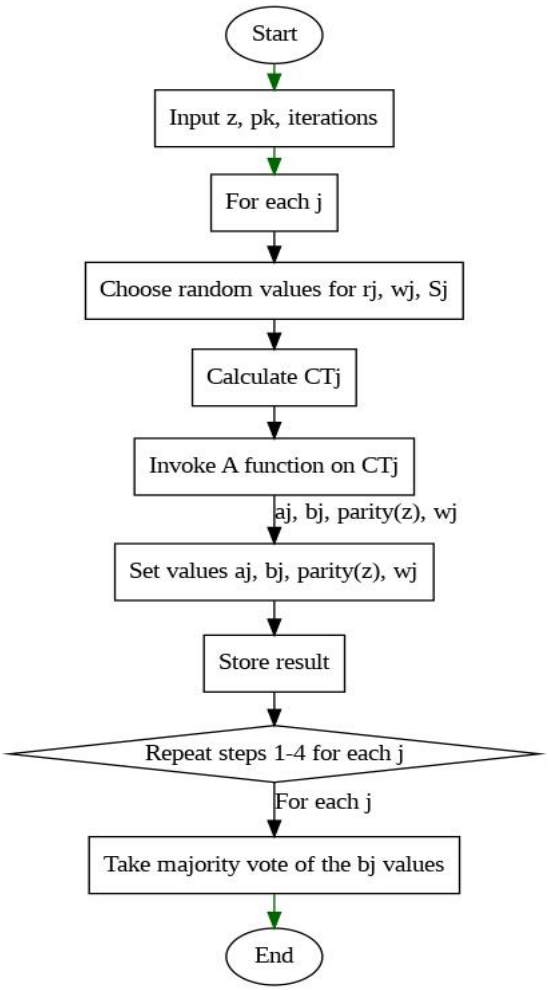

Figure 1. Flowchart depicting the Learn-LSB subroutine in Algorithm 1.

4.3 Algorithm 2

Inputs:

$z \in \{0,1\}^k$ with $|rp(z)| \leq 2^\lambda$

pk = (x_0, x_1,..., x_t)

Output:

The value of p

Set $z_1 = z + 2^\lambda$

For j = 1,2,...,t do the following:

a. Choose a noise value $r_j \in \{-2^{(2\lambda)}, ..., 2^{(2\lambda)}\}$ and a random bit $w_j \in \{0,1\}$

b. Choose a random subset $S_j \subseteq \{1,2,...,t\}$ such that $j \notin S_j$

c. Set $CT_j = z_j + w_j + r_j + \sum_{k \in S_j} x_k$

d. Compute aj = A(pk, CT_j)

e. Set bj = parity(aj), wj = aj mod 2, and sj = $2^{(2(j-1))}$ bj

Use the Binary GCD Algorithm to recover p from s_1, s_2, ..., s_t

(1) The Approximate GCD problem is often considered difficult, but in this case, it does not provide any additional information to the attacker.

(2) Despite having the secret key p, the connection cannot extract any information from the ciphertexts Ci1 and Ci2. The missing secret key skid makes it impossible for the server to decrypt the ciphertexts.

(3) The random noise ri is unrelated to the plaintext, ensuring that the ciphertexts are unreliable and preventing the server from learning anything new about the keyword wi.

(4) The text could benefit from providing more definitions and explanations of technical terms to improve clarity and understanding.

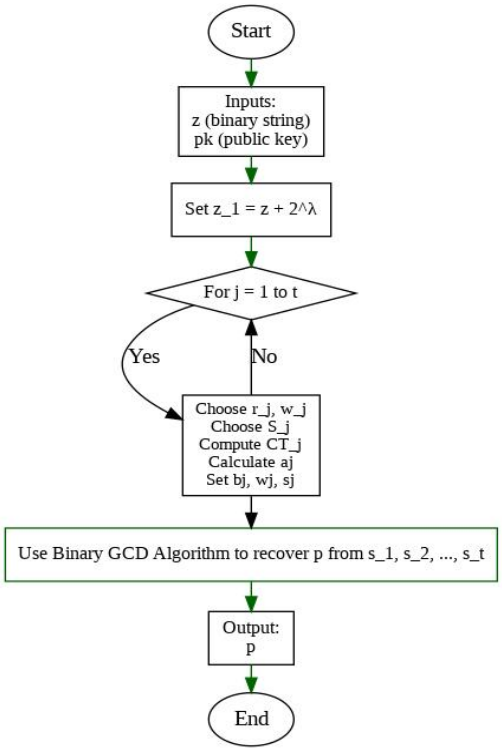

Figure 2. Flowchart Illustrating the Cryptographic Process for Secure Key Recovery using the Learn-LSB Subroutine and Binary GCD Algorithm

4.4 Performance Evaluation

This section assesses how effective our system is primarily by comparing it to other indicator accessible cryptographic algorithms, its functionality, computational cost, and experimental results. Suppose that |DO| represents the overall total of authorised data owners, n the overall total of determines, m the overall total of competitive keywords, t the overall total of key words and phrases queried, with d the overall total all search results [17].

First, we compare our system's functionality and computational complexity to those of other analogous systems in Table 2 and Table 3.

Table 2. Functional Comparison

Schemes	Multiple Keywords	Veriiable	OIB	Data Update
Re-dPEKS	No	No	No	Yes
Re-dtPECK	Yes	No	No	Yes
VMKD016	Yes	Yes	No	Yes
Our Scheme	Yes	Yes	Yes	Yes

Table 3. Computational Cost Comparition

Schemes	KeyGen	PEKS	Test	Verify
Re-dPEKS	(2|DO|+2)E	(2m+1)E+mP	(t+1)E+H+tP	\
Re-dtPECK	(|DO|+2)E	(m+5)E+2P	(t+4)E+(t+2)P	\
VMKD016	(|DO|+1)E	E+nH+3P	2P+3E	(d+1)E+dH+2P
Our Scheme	(|DO|)H	m(H+2A+2M)	m(D+2H+2M)t	tL

It goes without saying that our plan enhances search capabilities for encrypted data. It can accomplish the aforementioned functionalities concurrently, whereas the other three cannot, as seen in Table 1, which illustrates this. Our method permits your cloud server to implement an inverted encryption clustering technique requiring the need for a question trapdoor, greatly enhancing searchability [23].

Table 4 demonstrates that now the computational cost of the My strategy has simpler KeyGen, PEKS, Test, as well as Verify procedures than do competing approaches. Because our approach uses homomorphic encryption and therefore only requires basic addition, multiplication, & division operations, as opposed to the three other techniques' usage of bilinear pairing operations as a kind of cryptography.

Table 4. Comparison of Index-based Searchable Encryption Protocols

Schemes	Index Structure	Index Size	Search Complexity	Data Update Structure
Z-IDX	FIle Keyword	O(n)	O(n.m)	Bloom Filter
PPSED	FIle Keyword	O(n.m)	O(n.m)	Masked Index
MKPSE	Keyword File	O(n.m)	O(m2)	Index i
DEPKS	Keyword File	O(m2)	O(m log m)	Index C
Our Scheme	Keyword File	O(n.m)	O(m)	Z-Index

We compare index structure's search efficiency to a number of related searchable encryption systems, as well as the results are shown in Table 4. Generally speaking, there are two categories of

index structures: The phrase "keyword-file" and "file-keyword" are both referred to as the index file. Table 4 displays overall able to gain an understanding for schemes that are similar to ours but less sophisticated, because of the structure of the index. We introduce the Z-Index clustering algorithm in proposed scheme, which can allow search strategy & results verification to guarantee the accuracy and comprehensiveness of search results [19].

Comparatively, it demonstrates whether our plan is more effective when all the elements in Table 3 are taken into account.In order to createThe VMKDO16 scheme basic steps encrypt data set F using the standard public key encryption method before signing anything like the decryption set. The cost of computing the signature for each system files block increases. The index would then be constructed for the file collection that use the specified keyword set. Several bilinear pairings and exponential operations make up the majority of the process. To complete the PEKS method, our scheme only requires two multiplications, two additions, and just a hash operation. As a result, the scheme VMKDO16 has a substantially larger computational workload for the PEKS algorithm than do our schemes.Basically, the PEKS technique is impacted by that of the keyword m, and as m increases, so does the computational load just on algorithm. But, since addition & multiplication only require a little amount of computing, our technique performs nearly unaffected [20].

Whereas the computation complexity of the VMKDO16 scheme is practically constant, it rises according to the number of t in our system. This is the case because the VMKDO16 scheme is built using an attribute encryption method, It implies that the number of search phrases has a big impact on how quickly the trapdoor is produced. No matter the amount of keywords, the Test technique only needs three power exponential operations including 2 permutation and substitution operations during the Test phase. Given that the encrypted indices include keywords and that our system is based on homomorphic cryptographic methods, it will expand exponentially as more keywords are searched. Hence, the VMKDO16 scheme will outperform our technique whenever t is large enough [26].

The advent of cloud computing and the proliferation of online services have led to a significant amount of personal data being stored on the internet. This data requires limited access and privacy protection, which has become a growing concern for individuals [27]. To fully realize the potential of cloud computing, it is necessary to have technologies that can support safe online data management. Traditional information security measures rely on strong passwords and safe data transfer to ensure secure transmission of data to the server and prevent unauthorized access [28].

However, when data is decrypted and put into plaintext operation on the server, it becomes vulnerable to shady service providers and malevolent attackers. For instance, an unencrypted private photo album stored online can be accessed by a system administrator. Encrypting data using cryptographic ciphers can hinder access to the encrypted database for both the user and the server [28].

To address these issues, it is desirable and essential to create systems that retrieve data using encoded information while safeguarding users' confidentiality and preserving the usefulness or ability of the data. This can be achieved through techniques such as homomorphic encryption and distance preserving randomization [23].

The proliferation of digital cameras and mobile phones has led to a significant increase in the volume of digital photos, which now constitute a substantial portion of personal data. These images can be easily managed and stored online, allowing for convenient access from any location. This paper focuses on the problem of content-based image search in privacy-sensitive contexts, such as personal online photo albums, where confidentiality of data is of utmost importance in order to prevent unauthorized access, including by network operators. To address this issue, we investigate two main categories of techniques: homomorphic encryption and distance-preserving randomization. Our study is motivated by significant technological trends in the field, and we discuss related work in the following section [29].

In this study, we explore the challenging research questions surrounding privacy-preserving content-based image search. Achieving a balance between effectiveness and safety is critical for applications that manage personal image collections. Such secure online programs must be highly efficient and require minimal user interaction. We examine two primary categories of solutions: symmetric encryption and search index and visual feature randomization. We provide quantitative comparisons of these two categories of methodologies in terms of search precision, security potency,

and computational effectiveness. Feature/index randomization approaches use deterministic distance-preserving randomization to achieve high efficiency, but at the cost of disclosing some information about data distribution among the randomized features. The homomorphic-based approach offers greater security but is excessively demanding in terms of computational difficulty, knowledge load, and user participation, making it unsuitable for real applications. Both approaches surpass traditional searches that don't provide privacy protection in terms of search accuracy. We anticipate that this comparative analysis will guide the development of effective and safe privacy-preserving methods for personal privacy image search in various private web applications with diverse safety and effectiveness requirements [30].

In response to the increasing demand for greater storage and processing capacity from businesses and consumers, new technologies have emerged, including cloud computing. This technology enables users to outsource and process data remotely based on their needs, with the added benefit of pay-per-use computational power and storage space, providing greater flexibility and affordability.

However, the data outsourced through cloud computing, such as private emails, financial records, medical reports, and pictures, are often sensitive and confidential. Thus, it is strongly recommended to encrypt such data before outsourcing them to safeguard against any attacks on the cloud system. However, when a user searches the exported and encrypted data, the cloud server cannot use standard search techniques because the data are encrypted. In the literature, several researchers have proposed different solutions to address this issue, with varying levels of success and security. The majority of these approaches rely on the vector space model, where each vector in the index represents a page, and the index is composed of a set of vectors that represent components of a feature space of size n. The advantage of this vector model is that, with the right cryptosystem, it is possible to secure the index while still retaining the ability to search over it.

Also, vector-based techniques have two fundamental limitations. Firstly, as the amount of collected data increases, the search process becomes ineffective. In this case, both the size and number of vectors grow, resulting in an extensive index table. During the search process, the server computes an arithmetic average between each vector representation and the query vector. When the query dataset is large, the program needs to load and traverse the entire index, rendering the search useless. Secondly, this method is worthless in real-world applications as index updating is disabled when new terms are introduced to the data collection. Despite the fact that the information retrieval system speeds up the process of searching by providing users direct access to files containing the search terms, it has received little attention in the literature due to its vulnerability. An inverted index is built in this way, pointing to a set of documents containing each phrase that appears in the collection. Unfortunately, this structure reveals how a set of phrases and a specific document are related, which violates the keyword privacy constraint and may result in the exposure of sensitive information also it may pose potential danger in IoT environments [32].

5. Conclusion

This study recommends using homomorphic encryption in combination with validated public key encryption in multi-user environments. Our proposed technique enables the server to build a secure index for reversed encryption, reducing computational complexity to O, which is comparable to a single-term search. Our approach allows multiple users to validate the accuracy and comprehensiveness of search engine results while running encrypted keyword queries on encrypted data. Despite security concerns related to the use of a random oracle, our technique is secure. Unlike the vector model, which has drawbacks such as slow search speed and data updating issues, our proposed technique enhances search efficiency and adds naturalness to the selection process. The suggested safe IP version serves as the basis for the SIIS approach, and a second secured intermediate representation is used to securely manage each user's access privileges to the data. Our experimental research shows that our proposed approach outperforms methods based solely on the vector space framework.

References

[1] T. Adame, A. Bel, B. Bellalta, J. Barcelo, M. Oliver, "IEEE 802.11AH: the Wi-Fi approach for M2M communications," *IEEE Wireless Communications Magazine*, vol.21, no.6, pp.144-152, 2014.

[2] E.-J. Goh et al., "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *in Advances in Cryptology - EUROCRYPT 2004, ser. Lecture Notes in Computer Science,* C. Cachin and J. Camenisch, Eds. Springer Berlin Heidelberg, vol. 3027, pp. 506-522. 2004.

[4] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Applied Cryptography and Network Security, ser. Lecture Notes in Computer Science*, J. Ioannidis, A. Keromytis, and M. Yung, Eds. Springer Berlin Heidelberg, vol. 3531, 2005, pp. 442-455.

[5] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," *in Advances in Cryptology - CRYPTO 2007, ser. Lecture Notes in Computer Science,* A. Menezes, Ed. Springer Berlin Heidelberg, vol. 4622, pp. 535-552, 2007.

[6] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *2010 Proceedings IEEE INFOCOM*, Mar. 2010, doi: https://doi.org/10.1109/infcom.2010.5462196.

[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multikeyword ranked search over encrypted cloud data," in *INFOCOM, 2011 Proceedings IEEE*, pp. 829-837. Apr. 2011.

[8] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," *in Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pp. 383-392,Jun.2011.

[9] N. Cao, Z. Yang, C. Wang, K. Ren, and W. Lou, "Privacy-preserving query over encrypted graph-structured data in cloud computing," *International Conference on Distributed Computing Systems*, pp. 393-402, Jun.2011.

[10] Y. Lu, "Privacy-preserving logarithmic-time search on encrypted data in cloud," in *NDSS.The Internet Society*, 2012.

[11] R. Du, C. Ma, and M. Li, "Privacy-Preserving Searchable Encryption Scheme Based on Public and Private Blockchains," *Tsinghua Science and Technology,* vol. 28, no. 1, pp. 13-26, 2023, doi: 10.26599/TST.2021.9010070.

[12] V. Yousefipoor and T. Eghlidos, "An efficient, secure and verifiable conjunctive keyword search scheme based on rank metric codes over encrypted outsourced cloud data," *Computers and Electrical Engineering*, vol. 105, p.108523. 2023, doi: 10.1016/j.compeleceng.2022.108523.

[13] M. Hozhabr, P. Asghari, and H. H. S. Javadi, "Dynamic secure multi-keyword ranked search over encrypted cloud data," *Journal of Information Security and Applications*, vol. 61, p. 102902, Sep. 2021, doi: 10.1016/j.jisa.2021.102902.

[14] I. Amorim and I. Costa, "Homomorphic Encryption: An Analysis of its Applications in Searchable Encryption," 2023, [Online]. Available: http://arxiv.org/abs/2306.14407.

[15] G. Xu, W. Ji, Y. Wang, X. Shi, Q. Huang, and Y. Gan, "Searchable encryption algorithm based on key aggregation of multiple data owners in data sharing," *Journal of Information Security and Applications*, vol. 78, p. 103600, 2023, doi: 10.1016/j.jisa.2023.103600.

[16] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "Scifi ─ A system for secure face identification," *in Proc. IEEE Symp. Sec*, pp. 239-254, May.2010.

[17] D. E. Yan Huang, L. Malka, and J. Katz, "Efficient privacy-preserving biometric identification," *in Proc. 18th Network Distrib*, pp. 1-9, 2011.

[18] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," *CiteSeer X (The Pennsylvania State University)*, Apr. 2009, doi: https://doi.org/10.1109/icassp.2009.4959888.

[19] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *IEEE Transactions on Image Processing,* vol. 15, no. 7, pp. 2061-2075, Jul. 2006.

[20] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 905-917, Oct. 2006.

[21] H. Kim, J. Wen, and J. D. Villasenor, "Secure arithmetic coding," *IEEE Transactions on Signal Processing,* vol. 55, no. 5, pp. 2263-2272, May. 2007, doi:https://doi.org/10.1109/tsp.2007.892710.

[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.

[23] Y. Zhao, X. Chen, H. Ma, Q. Tang, and H. Zhu, "A new trapdoor indistinguishable public key encryption with keyword search," *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, vol. 3, no. 1/2, pp. 72-81, 2012.

[24] D. E. Knuth, *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms.* Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1997.

[25] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *in Advances in Cryptology EUROCRYPT 99, ser. Lecture Notes in Computer Science*, J. Stern, Ed. Springer Berlin Heidelberg, vol. 1592, pp. 223-238, 1999.

[26] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," *in INFOCOM, 2010 Proceedings IEEE*, pp. 1-9, Mar. 2010.

[27] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: verifiable attributebased keyword search over outsourced encrypted data," *in Proc. IEEE INFOCOM*, Apr./May 2014, pp. 522–530, 2014.

[28] W. Sun et al., "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Trans. Parallel Distrib. Syst*., vol. 25, no. 11, pp. 3025–3035, Nov. 2014.

[29] C. Guo, X. Chen, Y. Jie, F. Zhang, M. Li, and B. Feng, "Dynamic multi-phrase ranked search over encrypted data with symmetric searchable encryption," *IEEE Trans. Services Comput*., pp. 1–12, Oct. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8089767/, doi: 10.1109/TSC.2017.2768045.

[30] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers, " *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 24-43, 2010.

[31] M. N. Krohn, M. J. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," *in Proc. IEEE Symp. Secur. Privacy*, pp. 226-240 May.2004.

[32] N. Katuk and I. R. Chiadighikaobi, "An Enhanced Block Pre-processing of PRESENT Algorithm for Fingerprint Template Encryption in the Internet of Things Environment," *International Journal of Communication Networks and Information Security(IJCNIS)*, vol. 13, no. 3, Apr. 2022.