Received: May 31, 2023. Revised: July 8, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Efficient Cloud Data Sharing and Accessing Control Based on Public-Key Encryption with Authentication Search

S. Subbalakshmi[1]* **K. Madhavi[1]**

[1]*Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Anantapur, Ananthapuramu, Andhra Pradesh, India.*
* Corresponding author's Email: subbalakshmi.phd18@gmail.com

Abstract: It is becoming increasingly common to outsource data to the cloud. We recommend that the data owner encrypt his or her data ahead of outsourcing it to the cloud to ensure the confidentiality of the data. However, effective search for encrypted information is an essential and complex issue for cloud data sharing. In the past, searchable encryption (SE) has been proposed in many studies as the most promising solution for the efficient retrieval of encrypted data. However, these mechanisms are open to internal attacks that can be restored by assuming keywords from an open trapdoor (TPD) in an autonomous manner. This raises new security concerns, especially data confidentiality and security of cloud-based shared applications. In this paper, we present an efficient mechanism for the public key encryption method with authentication search (PKE-AS) which securely encrypts data using public-key cryptography and validates recipient authentication before retrieval with a shared secret key (SSK) exchange. The shared key is used by the sender to encrypt the keyword and by the receiver to generate a search TPD. The key idea of the proposal is to make sure that the system is safeguard the user data from intrusive keyword guessing attacks. This contribution provides a secure communication model for secure data sharing, searching and accessing by employing the key exchange protocol among the users. The comparison of PKE-AS with existing methods shows low executing time in analysis with different keyword search with encryption, TPD and Verification. It shows an average of 0.52ms less execution time in comparison results. The analysis of the experiment results demonstrates that the proposed method is suitable for practical use in cloud data sharing.

Keywords: Cloud data sharing, Access control, Public-key encryption, Authentication, Searchable encryption.

1. Introduction

Cloud computing has been conceived as the next-generation computing paradigm. The available applications and resources in the cloud environment will receive on-demand services via the Internet. In the past few years, the vast advancement in computing and communication makes cloud computing very admired which includes Google-Cloud, Windows-Azure, Amazon-Simple-Storage-Service, Baidu-Cloud, etc. [1-4]. It provides a scalable kind of storage network technology that accumulates and shares user data on various data servers in the cloud. Due to their many benefits, storage services of the cloud are extensively utilized in various applications [5, 6]. This comprises data volume, data availability, data sharing, and regular backup of data on a scale. A lot of advantages of cloud storage services advance client familiarity and service quality, and users can use cloud data at any time and without depending on any specific device.

While cloud storage services provide a lot of convenience to users, there are still many issues and challenges. When users submit their sensitive data to the cloud repository, the uploaded information can be compromised by various network attacks or intrusion in critical organizations [7-9]. In the meantime, users have lost the ability to effectively manage their data. It is essential to preserve the sensitiveness of data by employing a simple and secure method to encrypt the source data before uploading it to the cloud [10, 11]. However, how should users manage and retrieve encrypted information when they try to search for files that contain certain keywords is still a practical question. In the past, there were two ways to resolve this problem is suggested [12]. One of them is to copy the encrypted data locally and copy the encryption request. The first way is to download the entire encrypted data locally, and then download the required query for decryption. This process requires downloading a huge number of unnecessary files which increases excessive overhead and making the decryption expensive. This method is practically impracticable due to the high-cost complexity. Another problem is that the user sends the secret key (SEC-KEY) to the cloud server to open the question, but does not fully trust the cloud server.

In 2000, Song [13] first introduced search technologies, which became an important stage in the development of search encryption. Searchable encryption (SE) is an innovative technology that allows the user to select encrypted data sent from a cloud-encrypted server. There are two main types of SE technology in terms of encryption. The primary one is "Symmetric Search Encryption (SSE)" and the next one is public key (PUB-KEY) encryption using keyword search (PEKS). Many SSE methods have now been proposed [14, 15] because of their better effectiveness. However, users need to share the private key (PVT-KEY) securely to encrypt the data in the SSE and it is not appropriate for multi-user data sharing circumstances. PEKS resolved the difficulty of distributing SEC-KEYs creation by SSE, in evaluated with PEKS and SSE shows a wider scope of applicability. However, PEKS has the disadvantage of being inherently resistant to "Keyword Guessing Attacks (KGA)".

A keyword will be utilized as unrestrained. But for the real-time application, such a scenario might not be true always. Consumers frequently utilize restricted forms of keywords due to their habit, so it makes the original plural space a suffix and less entropy. Incidentally, the attacker is able to presume what words are included in the TPD search following a few guesses. First, the attacker assumes all the keys of the user are empty and then creates one key ciphertext after another to predict the match. The TPD requested by the attacker is checked one by one along with the key ciphertext he created. If the same situation happens by chance, the attacker can acquire the key knowledge the consumer is looking for, which will reveal the privacy of the user. Because the cloud server has the user's search TPD, these attacks can easily be mounted on the cloud server. These attacks are often referred to as internal KGA as IKGA.

In this paper, we present an efficient PUB-KEY encryption and authentication search (PKE-AS) mechanism that securely encrypts data employing PUB-KEY cryptography and validates recipient authentication before retrieval with an SSK exchange. In PKE-AS, the data sender not only encrypts keywords but also uses SSK to authenticate the receiver, so the search TPD can only match that data sender according to the bilinear diffie-hellman key exchange assumption. The proposed framework of PKE-AS is shown in Fig. 1.

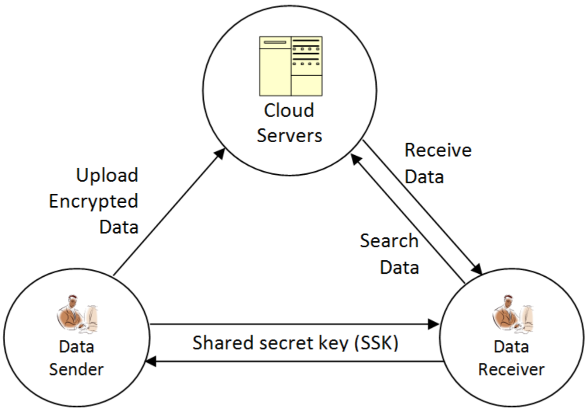

Figure. 1 Framework of PKE-AS

The deigned framework of PKE-AS has three interrelated entities that perform their actions. It is defined as the data sender (DS) who is the owner of the data, data receiver (DR) who downloads the data, and cloud server (CS) who do storage and verification of the authentic search. We observed that in PKE-AS, both DS and DR hold a couple of PUB-KEY / PVT-KEY. If the shared key can be counted without interactions, the common key can be seen as the SEC-KEY for the SSE method.

Therefore, the proposed PKE-AS mechanism calculates the SSK to be exchanged using the Diffie-Hellman (DH) key method [16, 17] for non-interactive authentication, and to encrypt keywords the SSE method in [8] is implemented. This removes time-consuming pairing from the old PKE method. To satisfy both ciphertext and TPD, we will prove that our method for IKGA in multi-user configuration is financially secure. In terms of security and computational efficiency, we evaluate our methods with some ancillary methods, as we do several testing to reveal the effectiveness of our plans to maintain privacy.

In the next section, we will briefly discuss the related works and some cryptographic primitives. In section 3, the proposal PKE-AS is presented and its security and privacy measures for data storage, search and retrieval is given. In section 4, the efficiency of the PKE-AS is evaluated in comparison with the other related methods. The summarization of the paper is given in section 5.

2. Related works

Searching encryption data have been widely utilized and studied in recent years by employing the searchable SE method. Nowadays most people keep their data in the cloud because of their unrestricted cloud storage and flexibility in the facilities. In order to get sure of the information security consumers usually symmetric or asymmetrically encrypt it before sending it to the cloud for sharing. It was noted before, that it makes certain and secrecy of the user information. However, some problems arise if someone tries to retrieve encrypted files uploaded to the cloud.

Since the availability of data in the cloud is in an encrypted form, so for consumers, it will be not easy to find such encrypted information simply. It can resolve this issue in two manners. First, the encrypted information is received by the consumer locally and then executes to decrypts the information, and then performs searches for an input keyword using a simple word. This method is safe but ineffective. If the received files contain a lot of information, they spend a lot of time compiling and also consume a lot of resources.

Another solution is to decrypt scripts that are encrypted in the cloud and retrieve simple scripts from the cloud. Hence, this response reveals the background of the retrieved information and risks the security of information and the privacy of individuals. Thus, how users are able to study key terms in digital signage in the cloud has sparked the interest of several scholars [18, 19].

SE is a cryptographic measure that allows authorized consumers to access data in a specific way in the cloud, such as key queries. Its purpose is to have the ciphertext server return the file format of interest to the user without knowing the cipher text material. Depending on the method of encryption, the encryption test can be divided into PEKS and SSE.

2.1 PEKS working mechanism

In 2004, Boneh et al. [18] established the PEKS structure for the first time. PEKS is relying on a broad spectrum of encryption methods in support of the three entities having DS, DR, and CS. The sender encrypts their documents and sorts them with PUB-KEYs, and submits them to the remote server. The acquisition of information, the acquisition of the necessary self-sufficient keys, can facilitate the activity analysis. It creates a TPD that wants to scan the keyword with PVT-KEY and send it to the server. After receiving the TPD, the integration server allows the user to check whether the password contains the keyword without knowing that the record label is the keyword. The service provider then returns the query to the owner, and in the end, the receiver can decrypt the text and send it to the server. PEKS is best suited for use in insecure internet settings. No part coding or part decoding is required to negotiate the key first.

The overall PEKS program consists of four possible polynomial-time methods [18].

- *Key_Gen* (*G*): This method implement by DR. It requires a secure input as G and generates the *PUB-KEY* and *PVT-KEY*.
- *PEKS* (*PUB-KEY, W*): This method implement by DS. *PUB-KEY* and keyword *W* are required as inputs, to specify the ciphertext keyword *S* of *W*.
- *TPD* (*PVT-KEY, W)*: The generation of TRD with keyword is implemented by this method. It requires *PVT-KEY* and query *W* to correctly match and execute to generate TPD_W for query key *W*.
- *Verify* (*PUB-KEY, S,* T_W): This method is implemented by CS to test the TPD access. It requires *PUB-KEY,* the ciphertext *S* of the keyword *W'* and the TPD_W for a query keyword *W*. If (*W* = *W* '), this method returns "true"; otherwise, "false" as result.

The PEKS framework relies on the PUBLIC-KEY configuration employing the bilateral mapping. The method allows any user to create searchable objects with the recipient's PUB-KEY, but only PVT-KEY users can create a TPD key for a query. Their design requires a safe way to transform the TPD search, but it seems to be expensive to build a secure path for execution.

Baek et al. [19] established PEKS methods that eliminate restrictions on the secure path for execution. The server's PUB-KEY and PVT-KEY are required for construction, and only the server selected by the sender can be searched. Security has been demonstrated in arbitrary oracle models in the BDH problem. Finally, Rhee et al. [20] designed a new PEKS method on PKI to improve Baek's model [14] and restrict attackers to find the link between ciphertext and TPD.

Park et al. [21] developed the initial PEKS method, which supports the binding of keyword search and PUB-KEY encryption and offers two classifications depends on the DBDH and the DBDHI issues. However, their designs require a lot of communication and storage costs. Hwang and Lee [22] advanced the work proposed in [21] and developed a novel concept known as multi-user PUB-KEY encryption with additional search keywords to preserve links and storage locations. Finally, Zhang et al. [23] designed a PEKS program that supports linking to a subset of search keywords and enhances performance in [21] tasks that can support supports keyword search only.

Lv et al. [24] designed a clear and secure PEKS approach that supports communication, interaction, and rejects search-based action on component combinations. This condition was safe in the experiment, which can be expanded to help search in a condition. Tan and Chen [25] set up a PEKS program called PUB-KEY encryption with a password registered, which allowed the user to post a keyword for the main topic for only the words subject registered with the sender. Their construction was stronger than predicted by offline keywords. Hu et al. [26] proposed encrypted PUB-KEY encryption, enhanced security against the prediction of keywords by a given test structure, and could decrypt the keyword from the ciphertext.

Fang et al. [27] provided a proven proof that SCF-PEKS is safe from selective selection attacks, ciphertext keyword prediction attacks. They provided a safe route without a PEKS program, without discussions under popular notion. Next, Shaon Yang [28] reinforces security models for keyword attack prophecy based on the work of [27] and describes cases in which the attacker is a malicious service. Recently, Lu et al. [29] extended the work of [28] also unable to resist internal KGA and proposed new development by Fang et al. [27] suggests a system of resistance against internal and external attacks.

Zhang et al. [30] established a new PEKS platform that supports automated search engine optimization and provides two specialized configurations that can maintain strong property protection and provide high efficiency in searches to outsource database data. Huang and Li [31] established a validation rule for PUB-KEY with a key search program in which the sender not only tags the keyword but also validates it. Their program was safe from the attack of internal keywords. Recently Wu et al. [32] established the Diffie-Hellman general SEC-KEY, based on the new PEKS framework, to achieve strong protection against file attack attacks and words in the attack prophecy predictions on existing PEKS systems.

2.2 PEKS with searchable symmetric encryption (SSE)

SSE agrees to the other member entities to perform external storage of its information on another sever entity while remaining eligible to perform selected searches. It is considered to be one of the most promising solutions to balance data confidentiality and availability. However, most of the existing experimental encryption methods do not have the required properties, so they cannot meet the requirements of high search efficiency and strong security at the same time. This question has been the center of investigation activity in current years, which over the years has been an interesting question of how to effectively obtain labeled data. However, in almost all PEKS programs, an in-house attacker can retrieve keywords from a specific TPD by manipulating their rankings online. So, how to counteract PEKS keyword attacks remains a major challenge in cloud data sharing.

It is a type of encryption that relies on symmetric encryption. In recent times, several articles have been developed to design encrypted data recovery methods. So, to increase the quality of SE and to further satisfy the user's interest in similar file systems Wang et al. [33] established a state-of-the-art symmetric encryption program in which documents obtained by one main study are aggregated accordingly. To be more efficient in this program, symmetric storage order coding was introduced. As for the popularity of sharing secure information is increased, so to have accurate search it is essential to have multiple keywords in the search queries.

Cao et al. [34] set out to force the usage of multi-word search on encrypted information. It employs similar links to find several kinds of information as probable and determine the correspondence among information and search keys utilizing similarity within the product. To decreased spelling recovery breakdown it enhanced search coding of multi-keyword by introducing fuzzy function for searching. The main method is to represent keywords as a vector unigram. This allows the spelling of words as words that are similar to the correct words by determining the Euclidean distance.

Recent studies of multi-keyword searches in multi-owner models [35], enrich SSE. In the program of Yin et al. [30], a set of data owners firmly shares the first two l-bit primes q_1, $q_2 \in Z_q$ with $q = q_1 . q_2$, where q_1 is used by the data owner to encrypt the secure index, and q_2 is reserved by privileged information users to SE query keywords.

It pre-defines the keyword vocabulary, $KV = (w_1, w_2, \ldots, w_n)$, where every keyword has its permanent location. Data client D_i retrieves keyword $W_{i\,,\,j} = (w_1, w_4)$ from information file $F_{i,j}$ and secure its position. The proposal avoids the effect of dropping phrases into individual files. Du et al. [28] also recommend SSE client capability with Boolean support. It resolving such issues that do not only allow users with information to re-authorize each other's queries without affecting the normal use of other people's information but also reduces user interaction.

2.3 Insider keyword guessing attack (KGA)

KGA is a possible password encryption problem that poses an open door for intruders. Byun et al. [36] started an investigation on KGA designing PEKS programs. This attack is caused by the low-entropy property of the keyword space. This attack allows the attacker to correctly guess the keyword added to the TPD keyword. In KGA [24], attackers can be divided into two types: external attackers and internal attackers based on the mode of intrusion. In general, the keyword area is considered very large at least. Alternatively, in practical functions, it is typically it is not very big. The selection of regular keywords depends on minimum entropy areas. Hence, an intruder can predict keywords in a document by executing a KGA method [37].

To be precise, during the KGA attack, the opponent tests encrypt, and verify the encrypted text with that TPD. When the experiment is successful, the partner knows the keyword contained in the TPD. Since individuals often select keywords that are frequently utilized and simple to memorize, the cloud servers are proficient to recognize the keys being used. Let's consider that forgiven the secure mail server, client Alice mails an encrypted message to another client named Bob. The opponent can use the KGA attack on PEKS code can detect e-mail passwords if there is TPD sent from Bob to an e-mail server. This lets the other party know the subject of the email, which leaks user privacy. This attack is usually carried out through CS or other tasks within CS supervision. Therefore, it is termed as "Inside Keyword Guessing Attack (IKGA)".

In opposition to the KGA's attack, investigators came up with a lot of ideas and ideas. In general, KGA functions for two causes. Initially, protesters can obtain a TPD, and second, it can be tested liberally. Therefore it prevent the attackers from spreading to external attackers, such as creating a trusted route between the receiver TPD and the cloud server, to prevent the KGA attacks from happening, only the server can prevent the TPD; or restricting the testing of unauthorized opponents, that is, designated testers PEKS [38] (that is, only designated servers can be tested), authorized PEKS [39] (that is, only authorized servers can be tested). However, no method can carry out an internal opponent's KGA attack. So the obvious question is how to build a reliable PUB-KEY SE program that is protected from hackers for predicting KGA is a very open and demanding question it needs to be addressed.

Table 1. Notations list

Notations	Description
PEKS	Public-key encryption keyword search
PKE-AS	Public Key Encryption method with Authentication Search
SSK	Shared secret key
TPD	Trapdoor
KGA	Keyword Guessing Attacks
DS, DR	Data sender, Data receiver
CS	Cloud server
ω	Search keyword
E_ω	Encrypted Keyword Ciphertext
TPD_ω	Keywords Ciphertext by Trapdoor
V_ω	Verification status of keyword

3. Proposed PKE-AS system

Since the introduction of PEKS, many researchers have been studying its safety, such as [15, 18, 19, 21]. However, to our knowledge, no proposal can effectively protect the KGA from internal opponents [27, 29, 31, 36]. In this part, we introduced the concept of public-key encryption and authentication search (PKE-AS), which is intended to counteract internal KGA. In the following section the Table. 1 notation are used.

3.1 PKE-AS model

The design PKE-AS model is based on the derive PEKS model which consists of three prime entities to perform data uploading by DS, data receiving by DR, and middleware CS to store and verify the authentication receiver. The communication flow between these three entities is shown in Fig. 2.

The methods operated by these parties are very similar to the PEKS method; the PEKS encryption method currently needs the DS to enter its SEC-KEY password, and the TPD generation method, and verification method into the DS PUB-KEY as a

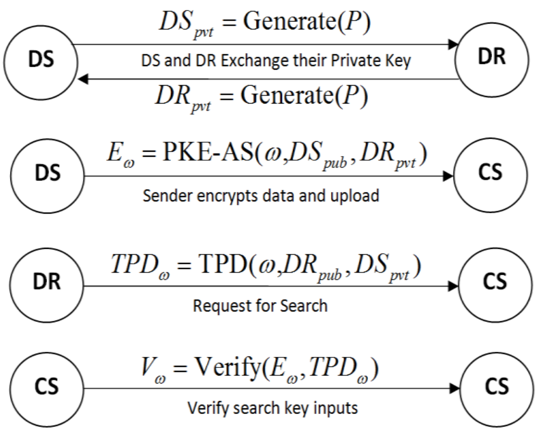
DS_{pvt} = Generate(P)
DS
DS and DR Exchange their Private Key
DR
DR_{pvt} = Generate(P)
DS
E_ω = PKE-AS(ω,DS_{pub},DR_{pvt})
Sender encrypts data and upload
CS
DR
TPD_ω = TPD(ω,DR_{pub},DS_{pvt})
Request for Search
CS
CS
V_ω = Verify(E_ω,TPD_ω)
Verify search key inputs
CS

Figure. 2 Flow of secure communication of PKE-AS model

component of the input. In general, it examines the practical ways in which the building of secure communication is facilitated.

The methods operated by these parties are very similar to the PEKS method; the PEKS encryption method currently needs the DS to enter its SEC-KEY password, and the TPD generation method, and verification method into the DS PUB-KEY as a component of the input. In general, it examines the practical ways in which the building of secure communication is facilitated.

The above-mentioned action flow of PKE-AS requires only a few parameters as input to generate a key for further authentication and search. We will discuss the functions of these actions below. There are four main steps in the actions performed between DS, DR, and CS. The first step is to create a verification key for DS and DR, the second step is to encrypt data for DS to upload, the third step defines the mechanism for DR to search for encrypted data, and finally verifies the DS keyword that DR requests the keyword encrypted with CS.

3.1.1. Authentication key generation

To create a securely authorized receiver, we consider using the diffie-hellman key exchange protocol (DH-KEP), which provides a way to create a secret shared key between DS and DR using a public channel. For example, the exchange of key symbols is expected on complex problems such as "modular exponentiation", "primitive roots", and "discrete logarithm" difficulties.

The DH-KEP function is a global phenomenon that is easy to implement but difficult to modify. It establishes the mathematical formation as a component of the cyclic field as G = ‹p›. here G represents the key generator factor. The generated key element of DS as p^a and DR as p^b selected from a range of elements which belongs to $a, b \in \{1,2, \dots, |G|\}$ which being exchange between DS and DR to establish a common level of an element as p^{ab} = $(p^a)^b$ = $(p^b)^a$ to have a secure data sharing. The program is useful because of exponentiation transforms and it is difficult to read the normal part of p^{ab} from p^a and p^b.

Suppose that M is a method that at the input security factor I^m, where "$m \in \mathbb{N}$", illustrates a cyclic set G of the corresponding position and p as a key creator of G. Here, the two participants are DS and DR who supposed to exchange their keys for the authentication method is,

<u>**DS Secure Key Generation**</u>		<u>**DR Secure Key Generation**</u>
Sample $(G, p) \leftarrow$ M(1^m)		
Sample $a \leftarrow U(\mathbb{Z}_{\lvert G \rvert})$	$\xrightarrow{(G,\, p,\, p^a)}$	Sample $b \leftarrow U(\mathbb{Z}_{\lvert G \rvert})$
$ds_k \leftarrow (p^b)^a = p^{ab}$	$\xleftarrow{p^b}$	$dr_k \leftarrow (p^a)b = p^{ab}$

Here the method for generating the sender's key pairs takes the global parameter of the system as P and generate a pair of the key as DS_{pub} and DS_{pvt} , similarly, the receiver also generates a set of key pair as DR_{pub} and DR_{pvt}. The creation of key of DS employs *Generate(P)* having a randomly select $a \longleftarrow Z_p$, and set $DS_{pub} \coloneqq p^a$ and $DS_{pvt} \coloneqq a$, and the key of DR having a randomly select $b \longleftarrow Z_p$, and set $DR_{pub} \coloneqq p^b$ and $DR_{pvt} \coloneqq b$.

3.1.2. Authentication key generation

The process of keyword encryption is performed by the PKE-AS encryption method which acquires a keyword ω, the receiver's PUB-KEY as DR_{pub} and the DS's SEC-KEY as DS_{pvt} as the parameter, and generate a resulted ciphertext as E_ω of the keyword ω as E_ω=$PKE\text{-}AS(\omega, DS_{pub}, DR_{pvt})$ in the following steps of execution as given in Eq. (1) to Eq. (5).

- Compute the keys

$$\text{sk'}||\text{sk"} = H\left((\text{DR}_{pub})^{DS_{pvt}}\right) = H\left(g^{ab}\right) \quad (1)$$

- Compute

$$X = E_{sk'}(\omega) \quad (2)$$

$$sk = f_{sk''}(X) \quad (3)$$

- Select a random string $S \in \{0,1\}^{n-m}$and set,

$$U = S||F_{sk}(S) \quad (4)$$

$$\text{Set, } E_\omega = X \oplus U \quad (5)$$

- Finally, return E_ω.

The generated E_ω finally upload to CS for sharing by the receivers.

3.1.3. Data search function

In order to perform a search of the data published by the DS receiver generate an encrypted TPD_ω with taking search keywords created using Eq. (1) along with DS_{pub} and DR_{pvt} as inputs and compute as, $TPD_\omega = TPD(\omega, DR_{pub}, DS_{pvt})$ given in Eq. (8).

$$X = E_{sk'}(\omega), \text{ and} \tag{6}$$

$$sk = f_{sk''}(X) \tag{7}$$

$$TPD_\omega = X||sk'' \tag{8}$$

The generated TPD_ω submit to CS for verification of the input search keywords.

3.1.4. DR search verification by CS

In order to have authenticated search CS verify the TPD_ω request by DR with the ciphertext of DS uploaded. The verification method takes the input TPD_ω and E_ω to validate $V_\omega = Verify(E_\omega\ ,\ TPD_\omega\ )$ if V_ω is *1* then verification of keyword pass else if *0* then it is unauthorized search. It computes $U = T_{chiper} \oplus X$ and interprets it as S || V_ω. If $V_\omega = F_{sk}\ (S)$, is *1* then matched, else *0* then failed.

3.2 Authentication and search suitability

Let the DR key set be DR_{pub} , DR_{pvt} = (g^b, b) and the DS key set be DS_{pub} , DS_{pvt} = (g^a, a). Then, the key $sk' \| sk'' = H(g^{ab})$ can be produced by utilizing the DS and DR keys. Let T_{chiper} be a cipher script of key information ω created by the DS and TPD_ω be the matching search TPD produced by the DR.

As per the encryption of keywords method, Z and U must have two texts and a random text given as S$\in$ $\{0,\ 1\}^{x\text{-}y}$ so that $T_{chiper} = Z \oplus U$, $Z = E_k\ (\omega)$, and $U = S||F_{sk}\ (S)$, where $sk = f_{sk''}(Z)$. For an accurate TPD of keyword ω, it has to be in the type of TPD_ω = $Z||sk$, where $Z = E_{sk}\ (\omega)$, and $sk = f_{sk''}(Z)$ the $U = T_{chiper} \oplus Z$.

Let S be the initial x--y bits of U and V_ω be the final y bits. So, the $V_\omega = F_{sk}\ (S)$ will hold like 1 or 0. Therefore, the similar key information in the cipher script will show alike the same TPD. If V_ω =0 is obtained, the authenticity search is invalid and the search is rejected; if V_ω = 1, the authenticity of the search user matches the search keyword, and the user can be allowed to search and download. The following is an experimental evaluation of the proposed PKE-AS to prove the effectiveness of the proposal.

4. Experiment evaluation

We examine the efficiency of the proposed PKE-AS by comparing it with other related methods in this section. The comparison analysis has made with HL-PEKS method-1 [40], PAEKS method-2 [41], PAEKS method-3 [42], and PEKS Method-4 [13], which are based on bilinear sets of computation. In these methods along with the set of G, there is a set G_T having a bilinear mapping e derived as of (G X G) to G_T.

We utilize dataset of Enron Email Dataset [43] for evaluation of the proposed method. We construct 3 different keywords related to the email subject and employ the method to compute the execution time with different keywords. It consists of 33400 documents, where we considered 11520 documents in related to the 3 keywords for the efficiency evaluation.

Table 1 shows the statistical results of the internal keyword guessing attack (IKGA) and the usage of the keywords comparison in terms of multi-user effectiveness, TPD creation, verification with authentication, and two other characteristics of protection.

In the table, we denote "*A*" as the authentication function described in the proposed PKE-AS. Other symbols, such as "*E*" for modular exponentiation, "*P*" for bilinear pairing, "*H1*" for an exceptional hash method that relates any character string to the set of elements, and "*H2*" for traditional hash function correspondingly for encryption, TPD and authentication.

In order to analyze the efficiency, we first evaluated the length of the key and the ciphertext, and then compared the HL-PEKS method-1, the encryption of each method in PAEKS, the data search function of TPD, and the execution time method of DR search verification of HL-PEKS method-1, PAEKS method-2, PAEKS method-3, and PEKS method-4.

Fig. 3 displays the comparison length of each one factor of HL-PEKS method-1, PAEKS method-2, PAEKS method-3, and PEKS method-4 with the proposed PKE-AS. Except for PEKS method-4, the other three methods involve DS PUB-KEY and PVT-KEY in the encryption method for keywords and method for TPD creation correspondingly. The figure shows that the PKE-AS proposed has a shorter TPD and ciphertext than other methods. For

Table 2. Comparison of efficiency of various methods

	Computation Factors For			Security Properties For	
Proposals	**Encryption**	**TPD**	**Verification**	**Multiuser**	**IKGA**
Proposed PKE-AS	E+H2+3A	E+H2+2A	A	Y	Y
HL-PEKS Method-1 [40]	3E+H1+H2+P	2E+H1	P	-	Y
PAEKS Method-2 [41]	3E+H1	E+H1+P	2P	Y	Y
PAEKS Method-3 [42]	3E+H1	E+H1+P	2P	N	Y
PEKS Method-4 [13]	3E+H1+H2+P	E+H1	P	Y	N

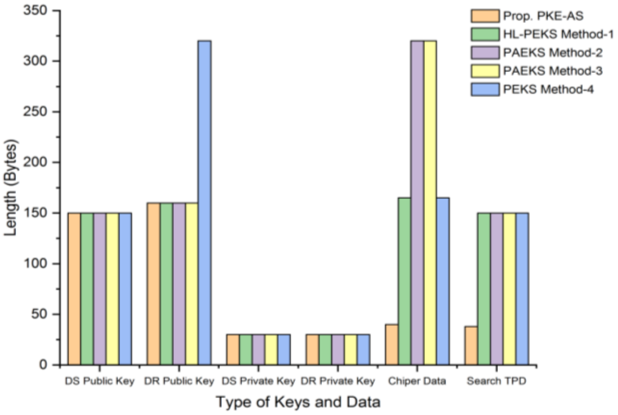

Figure. 3 Comparison of parameter length (Bytes)

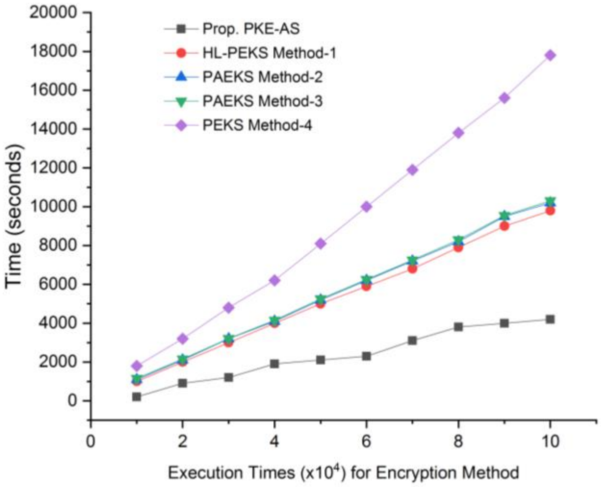

Figure. 4 Execution time of the encryption method

other factors, this method has a comparable length to other methods.

The mechanism of key pairing calculations between these operations typically takes the most time. So, as per [26] the creation of *H1* compared with traditional hash functions and its computational efficiency is usually lower is observed. It is straightforward to build a recognition function as an effective hash method in a random oracle model. According to the observation of execution time, we observe that the mechanism of keywords evaluation has to be faster in comparison to the past methods. To do the encryption and TRD generation, the advantages of our solution are not obvious. In terms of safety, PEKS Method-4 cannot resist IKGA. PEKS method-3 is able to mitigate IKGA, however, it is not safe in a multi-user configuration. HL-PEKS Method-1 does not show its security in multi-user settings.

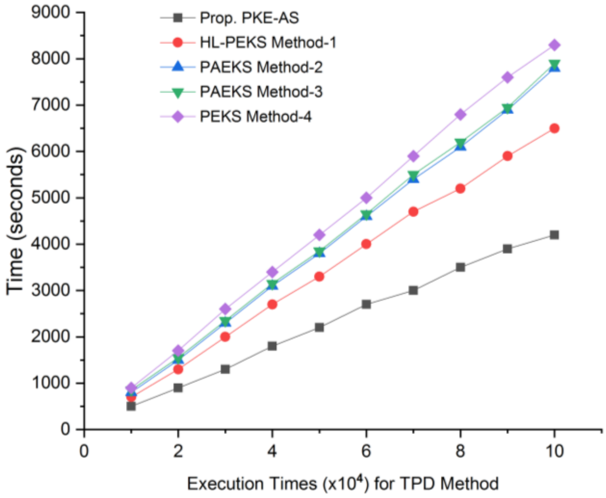

Figure. 5 Execution time of the TPD method

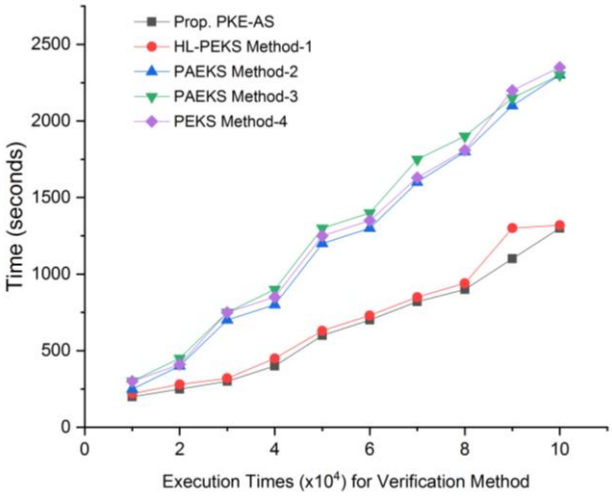

Figure. 6 Execution time of the verification method

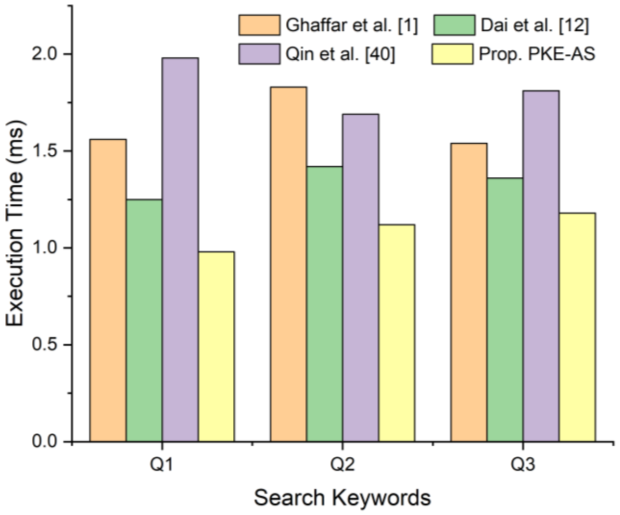

Figure. 7 Execution time (ms) comparison of the state-of-the arts method

To evaluate the effectiveness of this approach, we execute our proposed method in a Windows 8 operating environment with 6 GB of primary memory and an Intel Core i3 processor. We implement a library of Java Pairing Based Cryptography (jPBC) and select type A pairing with the curve $y^2 = x^3 + x$ for the workspace W_q, where the prime number $q = 3 \bmod 4$.

It executes and records every method at diverse execution times in seconds. The variations of the results are shown in Fig. 4, 5, and 6, correspondingly to justify the effectiveness of the work. It run and log each method at different execution times in seconds to evaluate the time complexity of each level of processing.

Fig. 4, 5, and 6 show the calculations of PAEKS method-2 and PAEKS method-3, which have almost identical experimental outcomes. Experimental outcomes illustrate that the proposed PKE-AS encryption method and TDP creation method are somewhat quicker than the comparison methods. But the verification of keywords in the proposed solution is faster than other solutions compared.

The result shown that using the three proposed methods for reducing execution time for accessing data achieve better than existing approach. The average of less than 2 sec for least query and 6 sec for more number of queries clearly enhance the total cost time for searching all matching and accessing indexing.

To evaluate the efficiency, we compared the proposed PKE-AS with state-of-the arts methods of Ghaffar et al. [1], Dai et al. [12] and Qin et al. [40] with relate to execution time assessment for searching the documents for accessing with encryption of the keywords as given in Fig. 7.

Fig. 7 shows an improvisation of the proposed PKE-AS in the execution time in compare with the state-of-the methods. The proposed PKE-AS shows least time of execution due to less number of iteration for performing keywords encryption, TPD and verification. It takes 0.58, 0.27 and 1ms less time for Q1; 0.71, 0.30, 0.57ms less time for Q2; and 0.36, 0.18, 0.63 less for Q3 in compare to methods of [1], [12], [40], it conclude a significant enhancement in the proposed method.

5. Conclusion

In this paper, we present public key encryption employing an efficient authentication retrieval mechanism that securely encrypts data using PUB-KEY cryptography and validates recipient authentication before retrieval with a shared SEC-KEY exchange. The methodology of DH key exchange protocol is utilized for creating a key to be shared among the DS and the DR. The key to being shared be able to consider as the SE Method using SEC-KEY as symmetric key, which is utilized by the DS or DR for search key encryption and generate TRD lookups. So, according to the experimental evaluation hypothesis, the proposed PKE-AS achieves both TDP and ciphertext creation, so that it can effectively withstand internal KGA. A comparison analysis with the state-of-art methods with measure of execution time for shows PKE-AS take less time for execution. For each executed search keyword, it takes an average of 0.51ms less time. It shows the fastness of the keyword search with authentication to accessing the shared data in cloud using this method. The results show the suitability of the proposal for the future practical application for the encrypted searching, data sharing and access control in Cloud.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, S. Subbalakshmi; writing—original draft preparation, writing—review and editing, S. Subbalakshmi; visualization, supervision, K. Madhavi.

References

[1] Z. Ghaffar, S. Shamshad, K. Mahmood, M. S. Obaidat, S. Kumari and M. K. Khan, "A Lightweight and Efficient Remote Data Authentication Protocol Over Cloud Storage

Environment", *IEEE Transactions on Network Science and Engineering*, Vol. 10, No. 1, pp. 103-112, 2023.

[2] A. Bakas, H. V. Dang, A. Michalas, and A. Zalitko, "The Cloud we Share: Access Control on Symmetrically Encrypted Data in Untrusted Clouds", *IEEE Access*, Vol. 8, pp. 210462-210477, 2020.

[3] S. Fugkeaw, "Secure Data Sharing With Efficient Key Update for Industrial Cloud-Based Access Control", *IEEE Transactions on Services Computing*, Vol. 16, No. 1, pp. 575-587, 2023.

[4] M. Konstantopoulos, P. Diamantopoulos, N. Chondros, and M. Roussopoulos, "Distributed Personal Cloud Storage without Third Parties", *IEEE Transaction Parallel Distribution System*, Vol. 30, pp. 2434-2448, 2019.

[5] I. Gupta, A. K. Singh, C. N. Lee, and R. Buyya, "Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions", *IEEE Access*, Vol. 10, pp. 71247-71277, 2022.

[6] A. Chiquito, U. Bodin, and O. Schelén, "Attribute-Based Approaches for Secure Data Sharing in Industrial Contexts", *IEEE Access*, Vol. 11, pp. 10180-10195, 2023.

[7] M. Xiao, H. Li, Q. Huang, S. Yu, and W. Susilo, "Attribute-Based Hierarchical Access Control With Extendable Policy", *IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 1868-1883, 2022.

[8] K. Yang, J. Shu, and R. Xie, "Efficient and Provably Secure Data Selective Sharing and Acquisition in Cloud-Based Systems", *IEEE Transactions on Information Forensics and Security*, Vol. 18, pp. 71-84, 2023.

[9] D. Kim and K. S. Kim, "Privacy-Preserving Public Auditing for Shared Cloud Data With Secure Group Management", *IEEE Access*, Vol. 10, pp. 44212-44223, 2022.

[10] E. Chen, Y. Zhu, K. Liang, and H. Yin, "Secure Remote Cloud File Sharing With Attribute-Based Access Control and Performance Optimization", *IEEE Transactions on Cloud Computing*, Vol. 11, No. 1, pp. 579-594, 2023.

[11] M. M. Sandoval, M. H. Cabello, H. M. M. Castro, and J. L. G. Compean, "Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud", *IEEE Access*, Vol. 8, pp. 170101-170116, 2020.

[12] W. Dai, S. Tuo, L. Yu, K. K. R. Choo, D. Zou, and H. Jin, "HAPPS: A Hidden Attribute and Privilege-Protection Data-Sharing Scheme With Verifiability", *IEEE Internet of Things Journal*, Vol. 9, No. 24, pp. 25538-25550, 2022.

[13] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data", In: *Proc. of International IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, pp. 44-55, 2000.

[14] B. Chen, L. Wu, L. Li, K. K. R. Choo, and D. He, "A Parallel and Forward Private Searchable Public-Key Encryption for Cloud-Based Data Sharing", *IEEE Access*, Vol. 8, pp. 28009-28020, 2000.

[15] S. Zhang, T. Yao, W. Liang, V. K. A. Sandor, and K. C. Li, "An Efficient Privacy-Preserving Multi-Keyword Query Scheme in Location Based Services", *IEEE Access*, Vol. 8, pp. 154036-154049, 2020.

[16] F. Bao, R. H. Deng, and H. Zhu, "Variations of Diffie-Hellman problem", In: *Proc. of International Conf. on Information and Communications Security*, Springer, pp. 301-312, 2003.

[17] M. Abdalla, M. Bellare, and P. Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES", In: *Proc. of Topics in Cryptology - CT-RSA 2001*, Springer, pp. 143-158, 2001.

[18] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search", In: *Proc. of International Conf. on Advances in Cryptology*, Springer, pp. 506-522, 2004.

[19] J. Baek, R. S. Naini, and W. Susilo, "Public key encryption with keyword search revisited", In: *Proc. of International Conf. on Computational Science and its Applications*, Springer, pp. 1249-1259, 2008.

[20] H. S. Rhee, J. H. Park, W. Susilo, and D. H Lee, "Improved searchable public key encryption with designated tester", In: *Proc. of International Symposium on Information, Computer, and Communications Security*, pp. 376-379, 2009.

[21] J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search", In: *Proc. of International Workshop on Information Security Applications*, Springer, pp. 73-86, 2004.

[22] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system", In: *Proc. of the International Conf. on Pairing-Based Cryptography*, Springer, pp. 2-22, 2007.

[23] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset

keywords search", *Journal of Network Computer Application*, Vol. 34, pp. 262-267, 2011.

[24] Z. Lv, C. Hong, M. Zhang, and D. Feng, "Expressive and secure searchable encryption in the public key setting", In: *Proc. of International Conf. on Information Security*, Springer, pp. 364-376, 2014.

[25] Q. Tang and L. Chen, "Public-key encryption with registered keyword search", In: *Proc. of European Public Key Infrastructure Workshop*, Springer, pp. 163-178, 2009.

[26] C. Hu and P. Liu, "An enhanced searchable public key encryption scheme with a designated tester and its extensions", *Journal of Computer*, Vol.7, pp. 716-723, 2012.

[27] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random oracle", *Information Science*, Vol. 238, pp. 221-241, 2013.

[28] Z. Y. Shao and B. Yang, "On security against the server in designated tester public key encryption with keyword search", *Information Process*, Vol. 115, pp. 957-961. 2015.

[29] Y. Lu, G. Wang, and J. Li, "Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement", *Information Science*, Vol. 479, pp. 270-276, 2019.

[30] R. Zhang, R. Xue, T. Yu, and L. Liu, "PVSAE: A public verifiable searchable encryption service framework for outsourced encrypted data", In: *Proc. of IEEE International Conf. on Web Services*, San Francisco, CA, USA, pp. 428-435, 2016.

[31] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks", *Information Science*, Vol. 403, pp. 1-14. 2017.

[32] L. Wu, B. Chen, S. Zeadally, and D. He, "An efficient and secure searchable public key encryption scheme with privacy protection for cloud storage", *Software Computer*, Vol. 22, pp. 7685-7696, 2018.

[33] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data", In: *Proc. of IEEE International Conf. on Distributed Computing Systems*, Genoa, Italy, pp. 253-262, 2010.

[34] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", *IEEE Transaction Parallel Distribution System*, Vol. 25, No. 1, pp. 222-233, 2014.

[35] H. Yin, Z. Qin, J. Zhang, L. Ou, F. Li, and K. Li, "Secure conjunctive multi-keyword ranked search over encrypted cloud data for multiple data owners", *Future Generation Computer System*, Vol. 100, pp. 689-700, 2019.

[36] J. W. Byun, H. S. Rhee, H. A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data", In: *Proc. of the Workshop on Secure Data Management*, Springer, pp. 75-83, 2006.

[37] W. C. Yau, S. H. Heng, and B. M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes", In: *Proc. of International Conf. on Autonomic and Trusted Computing*, Springer, pp. 100-105, 2008.

[38] C. Hu and P. Liu, "A secure searchable public key encryption scheme with a designated tester against keyword guessing attacks and its extension", In: *Proc. of International Conference on Computer Science, Environment, Ecoinformatics, and Education*, Springer, Vol. 215, pp. 131-136, 2011.

[39] S. Ma, Q. Huang, M. Zhang, and B. Yang, "Efficient public key encryption with equality test supporting flexible authorization", *IEEE Transaction Information Forensics and Security*, Vol. 10, pp. 458-470, 2015.

[40] B. Qin, Y. Chen, Q. Huang, X. Liu, and D. Zheng, "Public-key authenticated encryption with keyword search revisited: security model and constructions", *Information Sciences*, Vol. 516, pp. 515-528, 2020.

[41] M. Noroozi and Z. Eslami, "Public key authenticated encryption with keyword search: revisited", *IET Information Security*, Vol. 13, No. 4, pp. 336-342, 2019.

[42] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks", *Information Sciences*, Vol. 403-404, pp. 1-14, 2017.

[43] Enron Email Dataset: “www.kaggle.com/datasets/wcukierski/enron-email-dataset/”.