Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 4, pp. 1377–1396
DOI : 10.47974/JIOS-1607

A new bilinear pairing-based searchable encryption scheme

M. H. Noorallahzadeh †
Department of Mathematics
Faculty of Mathematical Sciences
University of Qom
Qom
Iran

R. Alimoradi *
Department of Computer Science
Faculty of Science
University of Qom
Qom
Iran

Abstract

The searchable encryption scheme is employed to address privacy and data confidentiality issues in outsourced databases. It enables users to send encrypted search queries containing a keyword to a server, which then returns all encrypted documents containing the specified keyword without learning additional information about the queries or documents. Boneh et al. [2] proposed a scheme for searching encrypted keywords without leaking information about the keywords and original documents. A unique technique for public-key encryption with keyword search (PEKS) has been presented in this paper, leveraging the power of bilinear pairings, offering better performance than [2] due to not using pairing operation in the encryption. Our scheme can also search conjunctive keywords simultaneously without leaking information and does not require the establishment of a secure channel, known as the secure channel free scheme. Additionally, we introduce a highly efficient SSE scheme that enables advanced conjunctive search capabilities. The innovative approach has been thoroughly examined and verified to be secure.

Subject Classification: *94A60.*

Keywords: *Searchable encryption, Cryptography based on pairings, Conjunctive keyword search.*

† *E-mail:* **Mh.noorallahzadeh@stu.qom.ac.ir**
* *E-mail:* **r.alimoradi@qom.ac.ir (Corresponding Author)**

1. Introduction

With the expansion of cloud services, users increasingly store their data on remote (untrusted) servers and access it through wireless networks or mobile phones. To protect privacy, data stored on remote servers must be encrypted. However, searching encrypted data without compromising the privacy of original documents and keywords seemed challenging at first. A design is required to enable user searches while providing privacy safeguards. In 2000, SONG et al. introduced the first symmetric searchable encryption (SSE) scheme [4], single keyword search support. This paper presents an SSE scheme that supports Conjunctive search and makes it more practical. The concept of public-key encryption with keyword search (PEKS) was first proposed and pioneered by a team of researchers in 2004, as documented in the work of [2]. The concept of PEKS was initially introduced by Boneh et al. [2] and has since been the subject of extensive research, drawing upon the versatility of bilinear pairings which have also been effectively used in other cryptographic applications such as efficient digital signatures [3]. The primary objective of a PEKS scheme is to enable a user to search for specific keywords within encrypted data without revealing the keywords themselves to the server. This pioneering work established the groundwork for further advancements in this specialized field of cryptography.

The scheme works as follows: Suppose Alice has a smartphone to check her emails. He wants to receive important emails individually and quickly. To achieve this, he wants to select the most important emails from his inbox. To maintain confidentiality, all emails sent to Alice are encrypted, ensuring that only Alice can access them. In this scenario, the server uses a "trapdoor" provided by Alice to determine what emails it wants to receive. For example, if Alice wants to receive instant emails from the server, The user generates a search token for the keyword "Instant" and sends it to the server. The email is then encrypted using a standard encryption algorithm, with additional encryption using the PEKS scheme. Alice can use keywords to retrieve the emails she wants from the server, ensuring that the server and other parties do not know more about the content of the email. This article describes our contributions in this field.

Building on prior work in conjunctive keyword search [9], our scheme enhances this capability by allowing simultaneous keyword searches without leaking information and without requiring a secure channel (a secure channel free scheme)

We provide a concise overview of the fundamental concepts that form the basis of our scheme. These concepts include bilinear pairings, which were foundational in the development of Identity-Based Encryption [1], complexity assumptions, and the definition of PEKS.

We present an optimized PEKS scheme by eliminating unnecessary mathematical operations in the encryption operation and removing the secure channel compared to [2].

We present a highly efficient SSE scheme that outperforms the approach described in [4], providing enhanced practical applicability. The structure of this document is as follows: Section 2 gives an overview of the relevant prior research in this domain. Section 3 introduces the necessary background information and preliminaries. In Section 4, we describe the details of our proposed schemes. Lastly, Section 5 summarizes the paper's key findings and contributions.

2. Related works

The concept of searchable encryption was initially introduced by Song, Wagner, and Perrig [4]. This paper describes cryptographic schemes for searching on encrypted data. The primary objective of this work is to mitigate security and privacy risks associated with the storage of data on servers, including mail servers and file servers. The cryptographic techniques proposed in this paper are rigorously analyzed and proven to be secure, offering verifiable confidentiality guarantees for the encrypted data, query isolation for searches, and controlled searching. Song's approach [4] relied on sequential scanning, which entails searching through the entire encrypted document. However, researchers discovered that sequential scanning leads to significant information leakage to the server. Additionally, it is inefficient and contributes to increased computational overhead. Mr. Goh later proposed an efficient scheme using Bloom filters to define security for searchable encryption [6]. The scheme explores the concept of secure indexes, which are data structures that enable the testing of word inclusion in an index without revealing any information about the index contents. It introduces z-idx, an efficient and secure index construction algorithm, and discusses its potential applications, including private database query schemes, encrypted searchable audit logs, and secure set membership tests. Unlike standard index constructions that use hash tables, The secure indexing approach introduced in this work enables users to test the index only for a specific keyword or term, using a specialized "trapdoor" mechanism. This trapdoor can be generated solely with a secret key, ensuring that no information about the document contents is revealed without the proper authorization. Conventional index constructions relying on hash tables are unsuitable for indexing encrypted documents, as they compromise the semantic security of the system by exposing details about the underlying data., which is not the case with secure indexes. These secure indexes can be used to build private database query schemes that enable private queries to be performed using a semi-trusted third party. This scheme requires substantial memory and search time and is susceptible to false positives. Most published searchable encryption designs are

based on a symmetric key setting, where the user encrypts and stores data on a remote server for keyword-based searches. However, these schemes are not suitable for practical applications like e-mail routing due to the challenges of establishing a symmetric key [5]. There is ongoing research in this area, with PEKS differing from previous solutions such as the public information retrieval (PIR) solution, as noted in [2]. This scheme discusses the PEKS, a technique that enables a gateway to examine encrypted data for specific keywords without learning any other information. An example scenario given is email routing, where certain keywords are used to determine which device should receive the email. This technology allows this routing to occur even when emails are encrypted for privacy reasons. Public-key encryption with keyword search is a cryptographic mechanism that allows a user to provide a specialized key to an intermediary entity, such as a gateway. This key enables the gateway to assess whether a specific keyword is present in an encrypted email or message, without gaining access to any other information about the content. Building upon the foundational work presented in [2], the first PEKS scheme utilizing bilinear pairings and supporting conjunctive search functionalities was proposed by Park, Kim, and Lee [12]. The proposed scheme outlines a method for PECKS. This approach permits the search for particular keywords within encrypted documents without disclosing any information about the documents themselves. It is built on the foundation of PECKS, allowing the search for documents containing a set of keywords in specific fields. PEKS is valuable in various settings where secure keyword searches in encrypted data are necessary without revealing any data information. The document highlights two such environments: email gateways and email servers. For instance, if Bob wishes to send an encrypted email to Alice using her public key, and Alice only wants emails containing specific keywords (e.g., "urgent" or "finance") to be directed to her pager or PDA, this encryption scheme would be beneficial. Similarly, email servers store numerous encrypted emails, and if Alice wants to retrieve emails containing specific keywords (e.g., "finance") without revealing any email information, this scheme would be advantageous. Golle, Staddon, and Waters [7] introduced a solution for conjunctive keyword search based on a robust secret key system. Their scheme allows the search for documents containing all specific keywords without revealing any information about the documents except for those that match the search. This solution involves creating a secret key for each keyword and using it to encrypt the corresponding document. Subsequently, the search is conducted using the intersection method, where the server learns which documents matched individual keywords but not which documents contained all the keywords. However, this solution is limited by its heavy secret key system. However, this scheme's efficiency is hindered by a large number of pairing operations. Waters et al. [8] The proposed PEKS scheme relying on bilinear pairings has been shown to have practical

applications in the development of searchable encrypted audit logs. This approach enables the creation of secure and searchable audit records, where authorized users can efficiently locate and retrieve specific entries without compromising the confidentiality of the entire audit log. Despite its introduction for the realization of PEKS, the scheme in [2] is almost impractical for large databases due to its use of complex mathematical operations and high computational overhead. This paper aims to enhance design efficiency by reducing complex mathematical operations, while also addressing other issues outlined in [2].

3. Preliminaries

Before introducing our method and outlining its findings, we will briefly outline the framework, definitions, and security models for the PEKS and SSE schemes.

Figure 1 illustrates the general architecture of a PEKS scheme. The data owner generates public/secret keys, encrypts keywords using the PEKS algorithm, and sends the resulting ciphertexts to the server. To perform a search, the user generates a Trapdoor for a specific keyword and sends it to the server, which then uses the Test algorithm to find matching ciphertexts without learning the underlying keyword

3.1 *Peks structure and, definitions*

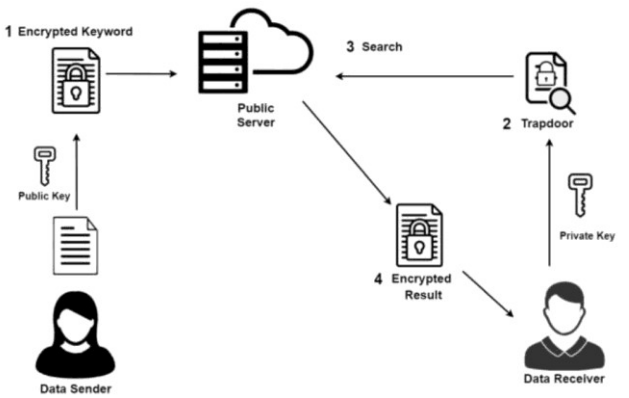

Figure 1
PEKS structure [11]

Negligible function: f: N → R is termed negligible if it diminishes to zero more rapidly than any positive polynomial. In other words, $f(x) < 1/g(x)$ for sufficiently large x and any positive polynomial g.

Definition 1: The PEKS scheme comprises four polynomial time algorithms as follows:

1. KeyGen(1^{λ}): Executed by the data receiver (or trusted entity), this probabilistic algorithm generates a public/private key pair (pk,sk) with a security parameter λ as input.
2. Trapdoor(sk, W): Executed by the data receiver, this deterministic algorithm produces a trapdoor T_W , this algorithm takes two key inputs: sk, the receiver's private key and W, the target keyword. These inputs are utilized within the core functionality of the system to enable secure and controlled access to the encrypted data.
3. PEKS(pk, W): Executed by the data sender, this algorithm returns ciphertext S = PEKS(pk, W), which represents the searchable encryption of W. The algorithm operates by accepting two primary inputs: pk , the receiver's public key and W the specific keyword.
4. Test(pk, S, T_W): This deterministic operation is performed by the server. It accepts three inputs: pk, S = PEKS(pk, W), and T_W = Trapdoor(sk, W). The algorithm yields the output 1 ("Yes") if W matches the provided input, and 0 ("No") otherwise.

Consistency:

The condition Test(pk, S, T_W) = 1 should be satisfied for all keywords W, where (pk, sk) = KeyGen, S = PEKS(pk, W), and T_W = Trapdoor(sk, W). This consistency property ensures the reliable and accurate operation of the system.

3.2 *SSE structure and, definitions*

Figure 2 illustrates the fundamental workflow of a Symmetric Searchable Encryption (SSE) scheme. In this model, the data owner uses a single secret key (sk) to both build the encrypted index (BuildIndex) and generate search tokens or trapdoors (Trapdoor), enabling the server to perform searches directly on the symmetrically encrypted data.

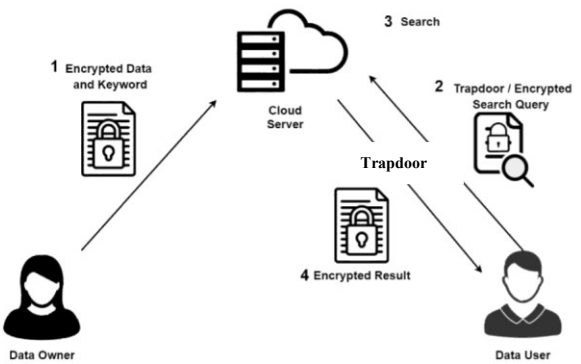

Figure 2
SSE structure [11]

Definition 2: The SSE scheme comprises four polynomial time algorithms as follows:

1. KeyGen(1λ): Executed by the data receiver (or trusted entity), This probabilistic algorithm generates a secret key (sk) and takes as input a security parameter λ
2. Trapdoor(sk, W): Executed by the data receiver, this deterministic algorithm returns a trapdoor TW and takes the sk and keyword W as input.
3. PEKS(sk, W): Executed by the data sender, this probabilistic or deterministic algorithm returns ciphertext S = SSE(sk, W), which represents the searchable encryption of W, and takes sk and word W as input.
4. Test(sk, S, T_W): This deterministic operation is performed by the server. It accepts three inputs: the secret key sk, a ciphertext S = SSE(sk, W), and T_W = Trapdoor(sk, W′). The algorithm yields the output 1 ("Yes") if W matches the provided input W′, and 0 ("No") otherwise.

Consistency:

The system is designed such that the Test(sk, S, T_W) operation will consistently output 1 for any given keyword W. This property ensures the reliable and accurate functioning of the system, allowing for the seamless retrieval of relevant data based on the provided search criteria., where (sk) = KeyGen(1^{λ}), S = SSE(sk, W), and T_W = Trapdoor(sk, W).

3.3 *Security Model for Symmetric-key Searchable Encryption*

To illustrate the security model of symmetric searchable encryption, we introduce Figure 3 in this figure:

Setup phase: In this phase, the challenger first executes the key generation algorithm and randomly chooses a value between zero and one. The attacker also produces as many keywords as they need.

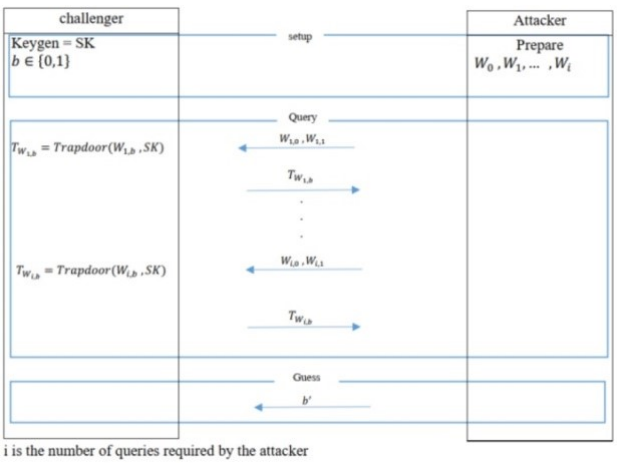

Figure 3
Security model of symmetric searchable encryption

Search Query Phase: The adversary is granted the ability to perform search queries using any number of keywords they have generated, as well as create as many keyword-trapdoor pairs as desired. The challenger, in response, selects a specific keyword with the aid of a randomly generated value and generates a corresponding trapdoor, which is then transmitted to the adversary.

Guessing phase: The adversary must guess whether the selected random value is zero or one.

Figure 3 visualizes the standard security model for Symmetric Searchable Encryption (SSE), often referred to as indistinguishability against chosen-keyword attack (IND-CKA). The model is a game between an adversary and a challenger. The adversary's goal is to distinguish between the trapdoor of a keyword they chose and a trapdoor for a randomly generated keyword, thereby testing if the scheme leaks any information about the queried keywords

3.4 *The Bilinear Map*

Suppose that the orders of groups G_1 and G_2 are both equal to a large prime number q. A bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ satisfies the following properties:

1. Bilinearity: The map $\hat{e}$ is bilinear, meaning that for any elements U, V in the group G_1 and any integers a, b, the equation $\hat{e}$(aU,bV) = $\hat{e}(U,V)^{ab}$ holds
2. Non-degeneracy: The generator P of the group G_1 is mapped by the bilinear map $\hat{e}$ to the generator of the group G_2. This ensures that the bilinear map $\hat{e}$ is non-trivial and captures meaningful information about the groups G_1 and G_2.

   These properties of the bilinear map $\hat{e}$ are crucial in various cryptographic applications, as they provide a strong algebraic structure that can be leveraged to construct secure and efficient cryptographic schemes.
3. Computable: For any elements U and V belonging to the mathematical structure G_1 There exists an efficient algorithm to calculate this map $\hat{e}$(U,V).

G_1 can be represented as a group of points on a specific type of mathematical object, known as an elliptic curve.

3.5 *The Boneh et al. [2] scheme is as follows:*

The KeyGen(1^{λ}) algorithm is responsible for generating the public and private keys for the Public Key Encryption with Conjunctive Keyword Search (PECKS) scheme. Here's a more detailed explanation of the steps involved:

The algorithm first constructs a mathematical structure called a group G_1, where the cardinality (order) of G_1 is a large prime number, denoted as q. This means that G_1 contains q distinct elements.

The group G_1 is generated by an element g, which means that every element in G_1 can be represented as a power of g. In other words, G_1 = <g>, where <g> represents the group generated by the element g.

Next, the algorithm creates a bilinear pairing function $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, where G_2 is another group with the same large prime order q as G_1. The bilinear pairing eˆsatisfies the following properties:

The algorithm then randomly selects an element x from the set Z_q^*, which represents the set of integers modulo q that are invertible. It then computes y = g^x, where the exponentiation is performed within the group G_1.

The pk is defined as (λ, q, g, $\hat{e}$, G_1, G_2, y), where λ is the security parameter.

The sk is simply the randomly selected element x.

Finally, the algorithm returns the key pair (pk, sk).

Trapdoor(sk,w): $T_W \leftarrow H_1(w)^x$; Return T_W

The Trapdoor(sk, w) algorithm takes the sk and a w as input, and generates a Tw for the w. The steps are as follows:

The algorithm computes Tw = $H_1(w)^x$, where H_1 is a hash that maps the keyword w to an element in the group G_1, and the exponentiation is performed within G_1 using the x.

The algorithm returns the computed Tw.

PEKS(pk, w): r ← Z_q^*; A ← g^r; $H_2(e(H_1(w), y)^r)$; Return (A,B)

The PEKS(pk, w) algorithm accepts pk and a w as input, and generates a conjunctive searchable encryption of the w. The steps are as follows:

The algorithm randomly selects an element r from the set Z_q^*, which represents the set of integers modulo q.

It computes A = g^r, where the exponentiation is performed within the group G_1 using the generator g.

It then computes B = $H_2(e(H_1(w), y)^r)$, where H_2 is another hash, $\hat{e}$ is the bilinear pairing function, and the exponentiation is performed within the group G_2.

The algorithm returns the pair (A, B) as the conjunctive searchable encryption of the keyword w.

Test(T_W,(A,B)): If $H_2(e(T_W,A))$ = B return 1 “Yes” Else 0 “No”

The Test(Tw, (A, B)) algorithm accepts a Tw and a conjunctive searchable encryption (A, B) as input, and determines whether the keyword associated with the Tw matches the keyword associated with the conjunctive searchable encryption (A, B). The steps are as follows:

The algorithm computes $H_2(e(T_W,A))$, where $\hat{e}$ is the bilinear pairing function and the computation is performed within the group G_2.

It then checks if the computed value is equal to the value B from the conjunctive searchable encryption (A, B).

If the values are equal, the algorithm returns 1 "Yes", indicating a match. Otherwise, it returns 0 "No".

The PECKS design consists of the following four algorithms:

Key Generation (KeyGen(1^λ)): This algorithm generates a key pair, denoted as A_{pub} and A_{priv}, respectively.

PECKS Encryption (PECKS(A_{pub}, $\mathcal{D}$)): This algorithm accepts the A_{pub} and a document $\mathcal{D}$ as input, and produces a conjunctive searchable encryption of the document.

Trapdoor Generation (Trapdoor(A_{priv}, $\mathcal{Q}$)): Given the A_{priv} and a query $\mathcal{Q}$, this algorithm generates a T_Q for the query.

Test (Test(A_{pub}, $\mathbb{S}$, T_Q)): This algorithm takes the A_{pub}, a conjunctive searchable encryption $\mathbb{S}$ = PECKS(A_{pub}, $\mathcal{D}$), and a T_Q = Trapdoor(A_{priv}, $\mathcal{Q}$) as inputs. It outputs 1 'Yes' if the conditions {(W_{I_1} = Ω_1), (WI2W_{I_2} = Ω_2), . . ., and (W_{I_t}= Ω_t)} are satisfied, and 0 'No' otherwise.

In various real-world scenarios, establishing a secure connection between the data owner and a recipient can be impractical or involve high expenses. To address this challenge, it would be beneficial to develop a scheme that does not require the setup of a secure communication channel.

Consequently, we aim to enhance our existing cryptographic scheme by introducing a new approach, referred to as the SCF-PEKS[1] scheme. Novel approach eliminates the need for a secure channel between the involved parties.

The key concept underlying SCF-PEKS is the utilization of a key pair associated with the server. The data owner employs the server's and the receiver's public key to generate a ciphertext. In this scheme, the receiver retains the ability to transmit a trapdoor, but this time, the Tw can be sent through a public channel.

The SCF-PEKS scheme consists of the following six operations:

Parameter Generation (KeyGenParam(λ)): This algorithm accepts a security parameter λ and returns a cp.

Server Key Generation (KeyGenServer(cp)): This algorithm generates the server's key pair, denoted as (pk_S, sk_S)).

Receiver Key Generation (KeyGenReceiver(cp)): This algorithm generates the receiver's key pair, denoted as (pk_R, sk_R).

SCF-PEKS Encryption (SCF−PEKS((cp, pk_R, pk_S,w)): This algorithm takes the common parameter cp, the receiver's public key pk_R, the server's key pk_S, and w as inputs, and returns $\mathbb{S}$.

Trapdoor Generation (Trapdoor(cp, sk_R,w)): This algorithm takes the common parameter cp, the receiver's sk_R, and w as inputs, returns a T_w for w.

Test (Test(cp, T_w, sk_S,S)): This algorithm takes the common parameter cp, a trapdoor T_w, the server's sk_S, and $\mathbb{S}$ as inputs. It returns the symbol "Yes" if the keyword w matches the keyword w_0 used to generate the ciphertext $\mathbb{S}$, and "No" otherwise.

3.6 *Security Model for Public-key Encryption with keyword search*

The prevailing security criteria for PEKS schemes is adaptive indistinguishability against chosen keyword attack (IND-CKA), as defined by Boneh et al. [2]. They presented a PEKS scheme that relies on the decision

[1] Secure Channel-Free Public Key Encryption with Keyword Search

Diffie-Hellman and employs a random oracle model in its construction. Within the PEKS ciphertext, the adversary is unable to discern which keyword is encrypted with a probability only negligibly higher than that of a random guess. Figure 4 illustrates the IND-CKA security game, which formalizes the indistinguishability under chosen-key attacks for the proposed encryption scheme. This game demonstrates how an adversary interacts with the encryption oracle to test the scheme's security properties.

IND-CKA game:

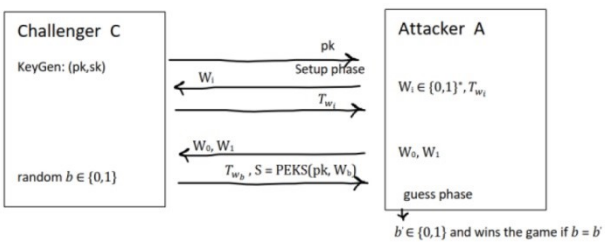

Figure 4
IND-CKA game

Key Generation (KeyGen): The challenger executes this algorithm, which generates a pk and a corresponding sk. The challenger retains the sk, while providing the pk to the attacker. The adversary receives the public key pk from the challenger. The attacker can adaptively request to generate a T_w for all W chosen from the set {0,1}*. The challenger responds to these requests by generating the requested trapdoors and providing them to the attacker.

In this attack model, the attacker has the public key and ability obtain T_w for all keywords of their choosing. The goal is to assess the security of the scheme under this adaptive query scenario.

Key Generation (KeyGen): The challenger executes this algorithm to generate (pk,sk). The attacker receives pk from the challenger. The attacker asks the challenger for T_W for all keyword W.

At this stage, it is important to emphasize that, given the public key, an attacker has the capability to generate numerous keywords and compute the corresponding trapdoors. However, the attacker is prohibited from requesting the trapdoor of these keywords from the challenger. Suppose an attacker, denoted as A, seeks to be presented with a challenge involving two words, W_0,W_1, and submits them to the challenger (In this attack model, the attacker has access to the pk and can obtain trapdoors for all keywords of their choosing, except for the specific keywords W_0 and W_1.). The challenger then proceeds to randomly selects a bit b from the set {0,1} and transmits S to the attacker, where S = PEKS(pk, W_b). The attacker has pk and can obtain Tw for all keywords of their choosing, without any restrictions.

Guess Phase: In the final phase, the attacker A, guess b' ∈ {0, 1}. The attacker wins if the guess b' matches the challenger's randomly selected bit b.

Attacker's Advantage:

The advantage of the attacker A is defined as follows:

Adv $\varepsilon_{,A}(\lambda) = |\Pr[b = b'] - 1/2|$

This represents the absolute difference between the probability that the attacker's guess b' is correct (b = b') and the probability of a random guess, which is 1/2.

The smaller the attacker's advantage, the more secure the underlying cryptographic scheme is considered to be.

Definition 3: A PEKS scheme is considered IND-CKA (Indistinguishability under Chosen Keyword Attacks) secure if, for any polynomial-time attaker A, the attacker's advantage Adv $\varepsilon_{, A}(\lambda)$ is negligible.

The BDH problem is considered to be intractable, meaning that there is no polynomial-time algorithm capable of solving it with non-negligible probability. In simpler terms, solving the BDH problem is not feasible.

3.7 *Security Model for Public Key Encryption with Conjunctive Keyword Search (PECKS):*

Security Model (ICC Game):

The challenger first executes this algorithm to generate a key pair (A_{pub}, A_{priv}). The challenger then provides the public key A_{pub} to the attacker.

The attacker is allowed to dynamically request the trapdoor T_Q from the challenger for all query Q of their choice.

Subsequently, the attacker submits two documents D_0 and D_1 to the challenger, with the condition that these documents have not been queried before.

The challenger proceeds to randomly selects a value b from {0, 1} then presents the attacker with the PECKS(A_{pub}, D_{b}) encryption.

The attacker retains the ability to request T_Q for any query Q of their choice, except for the queries related to the documents D_0 and D_1.

Finally, the attacker outputs a value b' from the set {0, 1}, and wins if $b = b'$.

The advantage is defined as follows:

$$\text{Adv}_{\text{A}}(\lambda) = |\Pr[b = b'] - 1/2|$$

Definition 4: A Public Key Encryption with Conjunctive Keyword Search (PECKS) design is semantically secure if, for any adversary A, the attacker's advantage $\text{Adv}_{\text{A}}(\lambda)$ is a negligible of the security parameter λ, as defined by the ICC game.

In the ICC game, the attacker is allowed to adaptively request trapdoors for any queries of their choice, except for the specific documents used in the challenge. The attacker then attempts to distinguish the encryption of one of the two challenge documents provided to the challenger.

If the attacker's advantage AdvA(λ) is negligible, it means that the PECKS design is secure against this type of attack, and the attacker cannot gain a significant advantage in successfully identifying the challenger's selection.

3.8 *Security Model for SCF-PEKS:*

Indistinguishability of SCF against Chosen Keyword Attack (IND-SCF-CKA):

This notion ensures that the server is unable to determine which PEKS ciphertext corresponds to which keyword. Even if an external attacker gains access to all the trapdoors for the keywords transmitted through the public channel, they are unable to draw any conclusions about the PEKS ciphertexts.

Attack Model (Game):

The IND-SCF-CKA security is defined between a challenger and an attacker:

Game: A represents an outside attacker (including the receiver).

Phase 1: A runs key generation algorithms KeyGenReceiver(cp) and, KeyGenServer(cp) and generates (pk_R, sk_R) and (pk_S, sk_S). only sk_S is secret from Outside attacker A.

Phase 2: The outside attacker can query any number of keywords and receives a corresponding trapdoor Tw.

Phase 3 (Additional Challenge): The outside attacker chooses a new keyword pair (w_0^*, w_1^*). After receiving this, the challenger randomly selects a bit β ∈ {0, 1} and returns the PEKS ciphertext $S^* = \text{SCF} - \text{PEKS}(\text{cp}, pk_R, pk_S, w_\beta^*)$ to the attacker.

Phase 4: The outside attacker continues to query several trapdoors.

Phase 5: Guess: The attacker outputs a guess b' ∈ {0, 1} for the value of b in the initial challenge, and a guess β' ∈ {0, 1} for the value of β in the additional challenge.

The advantage is defined as |Pr[b = b'] - 1/2| + |Pr[β = β'] - 1/2|, which should be negligible for the scheme.

Theorem 1: *The SCF-PEKS scheme is IND-SCF-CKA secure in the random oracle framework If the BDH problem is computationally hard.*

Proof: The evaluation of the Scenario is akin to the one described in [10].

Definition 5: The SCF-PEKS design is semantically secure if the advantage AdvA(λ) is a negligible function for any polynomial-time adversary A according to the game.

4. Our new schemes

4.1 *Our new PEKS scheme:*

Consider the bilinear pairing $\hat{e}$: G1 × G1 → G2, where (G1,·) and (G2,+) are cyclic groups of prime order p, and H :$Z_p^* \rightarrow \{0,1\}^{\log p}$ is a hash function. Let g be a generator of G1, and μ = $\hat{e}$(g, g).

The design is peresented as follows:

KeyGen: Randomly choose $x \in Z_p^*$, and output $pk = g^x$ and $sk = x$ (Public key and secret key).

Trapdoor: Given the W and, sk = x as input, and, $Tw = g^{(H(w)+x)^{-1}}$ as output.

PEKS algotithm: Randomly choose $r \in Z_p^*$, take $pk = g^x$ and a W as input, compute $A = g^{r(H(w)+x)}$, $S = H(r\mu)$ and output (A, S).

Test: Given the public key g^x, ciphertext (A, S), and trapdoor Tw, output 1 "Yes" if $H(\hat{e}(Tw, A)) = S$, otherwise 0 "No".

Consistency:

Consistency is proved as follows:

$$\begin{aligned} H(\hat{e}(T_W, A)) &= H(r\mu) \\ \hat{e}(T_W, A) &= (r\mu) \\ \hat{e}(g^{(H(w)+x)^{-1}}, g^{r(H(w)+x)}) &= \hat{e}(g^{(H(w)+x)^{-1}}, h^{(H(w)+x)}) \ , \ h = g^r \\ &= \hat{e}(g, g^r) \\ &= r\mu \end{aligned}$$

Efficiency:

The proposed scheme eliminates the need for pairing operations within the PEKS algorithm. Additionally, our design uses only one hash function, resulting in high efficiency.

Table 1
Comparisons between Boneh et al.[2] and our scheme

Schemes	Trapdoor	PEKS	Test
Boneh et al. [2]	H + P	2H + ê+ 2P	H + ê
Our scheme	H + P	H + P	H + ê

H: Hash function, P: Power operatio , ê: Bilinear pairing

Theorem 2: *The PEKS scheme is semantically secure against a chosen keyword attack, assuming BDH is intractable in the random oracle model.*

Security Proof: The analysis of the Game is similar to the one provided in [2]. According to [14], we compare some PEKS schemes with our scheme.

Table 2
Comparison of several PEKS schemes

schemes	Boneh et al. [2]	Baek, Safavi-Naini, and Susilo [10]	Wu et al. [13]	Our scheme
PEKS	ê+ 2P	2ê+ 2P	ê+ 5P	P
Trapdoor	P	P	5P	P

Test	ê	ê+ P	3ê+ 6P	ê
Search	Single	Single	Single	Single
KGA	vulnerable	vulnerable	resistant	vulnerable
SCF	No	Yes	No	No

As previously mentioned, the concept of PEKS was first introduced by Boneh et al. However, most PEKS designs are not resistant to offline Keyword Guessing Attacks (KGA). To address this issue, Wu et al. [13] introduced a KGA-resistant scheme.

The issue of Keyword Guessing Attack (KG) presents a significant threat in the context of PEKS. Byun et al. [15] were the first to introduce this attack on specific PEKS schemes. This attack enabling an attacker to accurately guess the keyword encoded in each keyword trapdoor. The KG attack classifies attackers into two types: outside attackers and inside attackers [16].

Baek et al. [10] proposed a PEKS design that safeguards against external attackers who may attempt to intercept the communication between the server provider and the recipient. This scheme ensures that even if these malicious entities manage to obtain the keyword ciphertexts exchanged over the public channel, they are unable to carry out the decryption process.

Conversely, an inside attacker typically refers to a malicious server provider with access to encrypted keywords from any sender and trapdoor information from the receiver. Resisting inside attackers is particularly challenging, especially in secure channel-free PEKS schemes.

In Section 4.3, we shall enhance our scheme and introduce an alternative scheme that eliminates the requirement for a secure channel.

4.2 *Enhance our PEKS scheme with multi-keyword search.*

In certain scenarios, both conjunctive and disjunctive search operations become necessary. It is important to note that the structure outlined above naturally supports disjunctive keyword search: the recipient can send trapdoors combined with the specific symbol "or" [T_{w_1}"or" T_{w_2}"or" ,..., "or" T_{w_n}] to the server (where W = $w_1, \dots, w_n$). The server then tests which PEKS ciphertexts match T_{w_1} or T_{w_2} or ... or T_{w_n}.

At one point, it was believed that a conjunctive keyword search could be achieved by sending [T_{w_1}"and" T_{w_2}"and",..., "and" T_{w_n}] to the server. However, this method leaks unnecessary information to the server, as published by Golle, Staddon, and Waters [7].

To address this issue, we propose an enhancement to enable secure conjunctive search.

KeyGen: A random selection of $x \in Z_p^*$ results in the output of pk= g^x and sk= x (Public key and secret key), similar to the "KeyGen" introduced in 4.1.

Trapdoor: Receives the sk= x and W as input, and $T_W = g^{(H(W)+x)^{-1}}$ as output, similar to the "Trapdoor" introduced in 4.1.

PECKS: A random selection of $r \in Z_p^*$, is followed by the receipt of the pk= g^x and output a W=(w_1,…,w_n). This results in the computation of $A_1 = g^{H(w_1).r}, g^{xr}, \dots, A_n = g^{H(w_n).r}. g^{xr}$, S= H(rμ) and the output ($A_1, \dots, A_n$, S)

Test: Receives the public key g^x as input, the ciphertext $A_1, \dots, A_n$, and the trapdoor T_W. It outputs "Yes" if $H(\hat{e}(T_W, A_i)) = S$ and $0 \leq i \leq n$; otherwise "No", similar to the "Test" introduced in 4.1.

Corollary 1: *Our PECK scheme (assuming DBDH is intractable) is semantically secure. The proof is similar to the one given in [12].*

4.3 *Our secure channel free PEKS (SCF-PEKS) design.*

The core idea behind PEKS is centered on the server having its own set of private and public keys. When generating a ciphertext, the sender utilizes both the server's pk and the receiver's pk. Subsequently, the receiver can transmit a Tw through a public channel to access information linked to the encrypted keyword. Upon reception of the Tw, the server can verify if the provided PEKS ciphertexts correspond to the trapdoor by leveraging its sk. This model is formally outlined as:

Definition 6: A SCF-PEKS design comprises the subsequent operations: KeyGenParam(k): This operation produces a common parameter cp when provided with a security parameter k. Key Generation for the Server (cp): By utilizing the cp as input, this procedure generates a key pair (sk_s, pk_s) for the server.

The Receiver's Key Generation process involves creating a key pair (sk_R, pk_R) based on the common parameter cp.

SCF-PEKS(cp, pk_s, pk_R, w): function utilizes the cp, the pk_s, the pk_R, and w to generate a PEKS ciphertext S. which represents the encrypted form of the keyword w. This process can be represented as S = SCF-PEKS(cp, pk_s, pk_R, w).

The Trapdoor(cp, sk_R, w) operation generates a Tw for the w using the cp and sk_R.

In the Test(cp, Tw, sk_s, S): operation, cp, a Tw for the w, the sk_s, and S = SCF-PEKS(pk_s, pk_R, w_0) are provided as input to determine if the keyword matches the encrypted representation.

It is important to note that a system administrator, is responsible for generating common parameters to establish long-term parameters within the system.

The bilinear pairing $\hat{e}: G_1 \times G_1 \rightarrow G_2$ is defined, where (G_1,·) and (G_2,+) are cyclic groups of prime order p. The hash $H:\{0,1\}^* \rightarrow G_2$ is represented by H, and g serves as a generator of G_1

The scheme includes the following steps:

- Key Generation for the Server: Randomly selecting $y \in Z_p^*$ results in $pk_s = g^y = Y$ and, $sk_s = y$ (Server's public-key and secret-key).

- Key Generation for the Receiver: Randomly selecting $x \in Z_p^*$ leads to $pk_R = g^x$ and, $sk_R = x$ (Receiver's public-key and secret-key).
- Trapdoor: $T_w = g^{H(w)-x}$
- SCF–PEKS: $U = g^{r_1 \cdot H(w)+x}$,$V = g^{r_2}$,$C = H(r_2(r_1+1)H(w)\hat{e}(g, Y))$, random choice of $r_1, r_2 \in Z_p^*$

Test: If $H\left(\hat{e}(U.T_w, V^y)\right) = C$, then the output is "Yes"; otherwise, it is "No".

Consistency:

To prove consistency, we can show that:

$$\begin{aligned}\hat{e}(U.T_w, V^y) &= \hat{e}\left(g^{r_1 \cdot H(w)+x}.g^{H(w)-x}, Y^{r_2}\right) \\ &= r_2(r_1+1)H(w)\,\hat{e}(g, Y)\end{aligned}$$

Security Proof: The security proof is similar to the one provided in [10].

4.4 *Our New Conjunctive SSE Scheme:*

1. Searchable Symmetric Encryption (SSE) schemes provide protection against keyword guessing attacks and demonstrate superior performance compared to PEKS schemes. The following SSE scheme allows for conjunctive keyword search, enabling efficient searches over encrypted document collections that have been outsourced, all while reducing information leakage. Efficient SSE schemes are capable of supporting conjunctive queries, which entail multiple keywords and necessitate their presence in the documents being searched.

Secure Searchable Encryption (SSE) schemes are crafted to offer confidentiality-preserving search functionalities for encrypted data. These schemes empower a user to conduct searches on encrypted data while keeping the content hidden from the data owner. This is made possible by encrypting the data in a manner that permits searching without the need for decryption.

Consider the following scenario: Let $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, where $(G_1, \cdot)$ and $(G_2, +)$ cyclic groups with order p. Within G_1, g serves as a generator. Additionally, let $D = (D_1, D_2 \ldots, D_m)$ denote the set of documents, where m represents the total number of documents. Furthermore, $W_j = (w_1, w_2 \ldots, w_n)$ represents the list of keywords associated with each document. Our SSE scheme encompasses four polynomial time algorithms as described below:

KeyGen: random choice of $x \in Z_p^*$, output sk = x

BuildIndex(sk , W_i): random choice of $r_i \in Z_p^*$, $S_j = E_{sk}(w_j)$, $w_j \in W_i$, $I_i = (r_i g, r_i S_i, \ldots, r_i S_n)$, $E_{sk}(.)$ is a symmetric encryption algorithm.

Trapdoor(sk,$l_1, \ldots, l_d, w'_1, \ldots, w'_d$): random choice of $a \in Z_p^*$, $t = \sum_{j=1}^{d} E(w'_j)$, $T = (ag, at, l_1, \ldots, l_d)$. l_i is the location of the ith keyword.

Test (I_i , T): Outputs "Yes" if $\hat{e}(at, r_i g) = \hat{e}(\sum_{j=1}^{d} r_i S_{l_j}, ag)$; otherwise "No"

Consistency:

Consistency is easily proved as follows:

$\hat{e}(at, r_i g) = \hat{e}(t, g)^{ar_i}$, $\hat{e}\left(\sum_{j=1}^{d} r_i S_{l_j}, ag\right) = \hat{e}(\sum_{j=1}^{d} S_{l_j}, g)^{ar_i}$

Therefore, if it is $t = \sum_{j=1}^{d} S_{l_j}$, the answer will be "Yes" and otherwise it will be "No".

Corollary 2: *Our conjunctive SSE, against an adaptive chosen keyword attack, is semantically secure. It is stated that the function $Adv_A(\lambda)$ is considered negligible for any attacker A who operates within polynomial time, as per the ICC game.*

Security Proof. The analysis of Game is like the one given in [7]

5. Conclusion

Our enhanced PEKS scheme includes features such as multi-keyword search functionality, improved query processing speeds, and enhanced encryption key generation techniques. By leveraging advanced cryptographic principles and innovative algorithms, we have created a highly secure and efficient PEKS scheme that is suitable for a wide range of applications. Our scheme has been extensively tested and evaluated, demonstrating strong resistance to various types of attacks and ensuring the confidentiality and integrity of sensitive data. Through continuous research and development efforts, we are committed to further improving and optimizing our PEKS scheme to meet the evolving security needs of modern information technology environments. This SCF-PEKS scheme allows for improved search capabilities within encrypted data by enabling users to perform keyword searches on ciphertexts without needing to decrypt them. By implementing this innovative cryptographic function, we aim to enhance the overall user experience and make our solution more user-friendly.

Furthermore, our SSE scheme is specifically designed to support conjunctive searches, allowing users to search for multiple keywords simultaneously within encrypted data. This feature enhances the efficiency and effectiveness of searches conducted on encrypted data, providing users with a seamless and convenient experience.

Overall, these complementary cryptographic functions not only enhance the usability and practicality of our solution, but also ensure that users can securely search and retrieve information from encrypted data without compromising its confidentiality.

References

[1] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in Advances in Cryptology – CRYPTO 2001, *Lecture Notes in Computer Science,* vol. 2139, C. Boyd, Ed. Berlin, Heidelberg: Springer-Verlag, pp. 213–229 (2001).

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," in Advances in Cryptology – EUROCRYPT 2004, Lecture Notes in Computer Science, vol. 3027, C. Cachin and J. Camenisch, Eds. Berlin, Heidelberg: Springer-Verlag, pp. 506–522 (2004).

[3] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in Public Key Cryptography – PKC 2004, Lecture Notes in Computer Science, vol. 2947, F. Bao, R. H. Deng, and J. Zhou, Eds. Berlin, Heidelberg: Springer-Verlag, pp. 277–290 (2004).

[4] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searching on Encrypted Data," in Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA: IEEE Computer Society Press, pp. 44–55 (2000).

[5] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in International Conference on Pairing-Based Cryptography, Berlin, Heidelberg: Springer-Verlag (2007).

[6] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report 2003/216 (2003). [Online]. Available: https://eprint.iacr.org/2003/216

[7] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Search over Encrypted Data," in Applied Cryptography and Network Security – ACNS 2004, Lecture Notes in Computer Science, vol. 3089, J. Katz and M. Yung, Eds. Berlin, Heidelberg: Springer-Verlag, pp. 31–45 (2004).

[8] B. Waters, D. Balfanz, G. Durfee, and D. Smetters, "Building an Encrypted and Searchable Audit Log," in Proceedings of the Network and Distributed System Security Symposium (NDSS) (2004).

[9] C. Gu, Y. Zhu, and H. Pan, "Efficient public key encryption with keyword search schemes from pairings," in International Conference on Information Security and Cryptology, Berlin, Heidelberg: Springer-Verlag (2007).

[10] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in International Conference on Computational Science and Its Applications, Berlin, Heidelberg: Springer-Verlag (2008).

[11] K.-M. Chan, S.-M. Yiu, K.-P. Chow, and L.C.K. Hui, "Trapdoor Privacy in Public Key Encryption with Keyword Search: A Review," IEEE Access, vol. 10, pp. 94691–94704 (2022), doi: 10.1109/ACCESS.2022.3195185.

[12] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in International Workshop on Information Security Applications, Berlin, Heidelberg: Springer-Verlag, pp. 73–86 (2004).

[13] L. Wu, B. Chen, S. Zeadally, and D. He, "An efficient and secure searchable public key encryption scheme with privacy protection for cloud storage," *Soft Computing*, vol. 22, pp. 7685–7696 (2018), doi: 10.1007/s00500-017-2795-1.

[14] Y. Zhou, Y. Zhang, Y. Wang, and X. Zhou, "Public key encryption with keyword search in cloud: a survey," *Entropy*, vol. 22, no. 4, p. 421 (2020), doi: 10.3390/e22040421.

[15] 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 Proceedings of the 2006 International Workshop on Secure Data Management (SDM 2006), Seoul, Korea, Sep. 10–11, pp. 75–83 (2006).

[16] 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 Sciences*, vol. 479, pp. 270–276 (2019), doi: 10.1016/j.ins.2018.03.003.

Received September, 2023