



Addis Ababa University

College of Natural Sciences

*Design of Searchable Encryption with Refreshing Keyword Search using
Pairing-Based Cryptography*

Kuma Bekele Ejeta

A Thesis Submitted to the Department of Computer Science in Partial
Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

October 2024

Addis Ababa University
College of Natural Sciences

Kuma Bekele Ejeta

Advisor: Minale Ashagrie (PhD)

This is to certify that the thesis prepared by *Kuma Bekele Ejeta*, titled: *Searchable Encryption with a Refreshing Keyword Search using Pairing-Based Cryptography* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards to originality and quality.

Signed by the Examining Committee:

<u>Name</u>	<u>Signature</u>	<u>Date</u>
<u>Advisor: Minale Ashagrie(PhD)</u>	_____	_____
<u>Examiner: Ayalew Belay (PhD)</u>	_____	_____
<u>Examiner: Dagmawi Lemma (PhD)</u>	_____	_____

ABSTRACT

To maintain data security and privacy, the Public Key Encryptions with Keyword Search (PEKS) scheme has been implemented. They offer search capabilities for encrypted data. However, because the Key Generating Center (KGC) knows the target users' private key, the existing PEKS schemes are vulnerable to key-escrow issues. The Certificate-Less Public Key Encryptions with Keyword Search (CL-PEKS) scheme was created to address the key escrow problem in PEKS schemes. However, refreshing keyword searches are not considered by the CL-PEKS schemes that are currently existing. As a result, the target server can launch keyword-guessing attacks and store search trapdoors for system keywords. By appending date information to the encrypted data and keyword, we proposed the certificate-less based Searchable Encryption with a Refreshing Keyword Search (SERKS) scheme.

We designed the system model and algorithms for the proposed certificate-less based SERKS using pairing-based cryptography. We also developed its prototype in the case of a web-based e-mail system by using Java Pairing-Based Cryptography (JPBC) library. The security hardness of the proposed scheme is based on the hardness of the Bilinear Diffie-Hellman (BDH) problem assumption. We assessed the suggested scheme's performance with respect to time complexity in terms of both communication and computational costs. The experimental results demonstrate that the suggested SERKS scheme has a lower computational cost than the two related schemes during the key generation and testing phases when compared to the earlier related work. Additionally, it has lower communication costs.

Keywords: Searchable Encryption, Refreshing Keyword Search, Pairing-Based Cryptography, Search Trapdoor, Keyword Guessing Attack

DEDICATED TO:

My parents

ACKNOWLEDGMENTS

First and foremost, I would thank God for being able to finish this thesis. Then, I would like to express my sincere gratitude to my advisor Minale Ashagrie (PhD) for his approachable guidance and for letting me do this thesis. Without his helpful criticism and unwavering support throughout the entire study, this thesis would not have been feasible. I want to express my gratitude to my friend Ebisa Desalegn for his unwavering guidance, inspiration, and support during the writing of this thesis. Lastly, I want to express my gratitude to my parents and friends, who have supported me throughout the thesis completion process with insightful advice and recommendations.

Table of Contents

List of Tables	iii
List of Figures	iv
List of Acronyms	v
Common Symbols and Basic Notation	vi
CHAPTER ONE: INTRODUCTION	1
1.1 Background	1
1.2 Motivation.....	3
1.3 Statement of the Problem.....	4
1.4 Objectives	6
1.5 Methods.....	7
1.6 Scope and Limitations.....	8
1.7 Application of Results.....	8
1.8 Organization of the Rest of the Thesis.....	9
CHAPTER TWO: LITERATURE REVIEW	10
2.1 Overview	10
2.2 Symmetric Key Encryption Keyword Searchable (SKEKS)	10
2.3 Public key Encryption with Keyword Search (PEKS).....	12
2.4 Elliptic Curve Cryptography (ECC).....	15
2.5 Pairing Based Cryptography	17
2.6 Semantic Security (SS)	20
2.7 Attacks in Searchable Encryption	22
2.8 Security Proof in Pairing-based Cryptography	22
CHAPTER THREE: RELATED WORK	25
3.1 PKI-Based Searchable Encryption.....	25
3.2 Certificate-less-Based Searchable Encryption	26

3.2.1 Generic Searchable Encryption.....	26
3.2.2 Certificate-less Searchable Encryption in the case of Email System.....	26
3.2.3 Certificate-less Searchable Encryption in the case of IOT	27
3.2.4 Certificate-less Searchable Encryption in the case of Healthcare System	28
3.3 Summary	30
CHAPTER FOUR: SEARCHABLE ENCRYPTION with a REFRESHING KEYWORD SEARCH.....	31
4.1 System Model	31
4.2 The SERKS Algorithm	34
4.3 Computational Consistency	37
4.4 The SERKS Flow Chart.....	37
4.5 Keyword Guessing Attack Resistance	38
4.6 Summary	39
CHAPTER FIVE: PROTOTYPE and PERFORMANCE EVALUATION	40
5.1 Prototype	40
5.2 Performance Evaluation	42
5.2.1 Computational Cost (Execution Time)	42
5.2.2 Communication Cost.....	44
5.2.3 Security Properties	45
CHAPTER SIX: CONCLUSION AND FUTURE WORK.....	46
6.1 Conclusion	46
6.2 Contribution	46
6.3 Future Work.....	47
REFERENCES	48
Annexes	53
Annexe A: Type A selected points on elliptic curve properties.....	53
Annexe B: <i>Source Code</i>	53

List of Tables

Table 5.1: Notations and execution times (ms).....	43
Table 5.2: Comparison of computational cost	44
Table 5.3: Comparison of communication cost	45
Table 5.4: Comparison of security properties	45

List of Figures

Figure 4.1: The SERKS System Model	32
Figure 4.2: Flow Chart of the SERKS Algorithm	38
Figure 5.1: User interface of the prototype for e-mail system.....	41
Figure 5.2: Ciphertext of the prototype	42

List of Acronyms

API	Application Programming Interface
BDH	Bilinear Diffie-Hellman
CA	Certificate Authority
ECC	Elliptic curve cryptography
ECCDHP	Elliptic Curve Computational Diffie–Hellman problem
ECDDHP	Elliptic Curve Decisional Diffie–Hellman problem
ECDLP	Elliptic curve discrete logarithm problem
CKA	Chosen-Keyword Attack
CL-PEKS	Certificate-less Public Encryption with Keyword Search
GMP	GNU Multiple Precision Arithmetic
IDE	Integrated Development Environment
IND-CKA	Indistinguishability of Ciphertext against an Adaptive Chosen Keyword Attack
KGC	Key Generating Center
KP-ABE	Key Policy-based Attribute-Based Keyword Search
PEKS	Public Key Encryption with Keyword Search
PBC	Pairing-Based Cryptography
PKG	Public Key Generation
PKI	Public Key Infrastructure
SE	Searchable Encryption
SERKS	Searchable Encryption with Refreshing Keyword Search
SKEKS	Symmetric Key Encryption with Keyword Search

Common Symbols and Basic Notation

$\in$	Belong to
ε	Negligible function
G_1	Additive cyclic group on elliptic curve
G_T	Multiplicative cyclic groups on elliptic curve
q	Order of group G_1 and G_T
P	Generator of G_1
PK	Public Key
SK	Private Key
$e: G_1 \times G_1 \rightarrow G_T$	Bilinear mapping
$e(,)$	Pairing Operation.
F_p	Rational function on point P in elliptic curve
F_{pk}	Field extension of pk
$H()$	Hash function.
C	Ciphertext
p	Field size
Z	Integer
Zq^*	Cyclic group (generated by a single element) of a prime number
T_w	The trapdoor of keyword w
T	Timestamp
w	The keyword set

CHAPTER ONE: INTRODUCTION

1.1 Background

A cryptographic scheme refers to a specific set of cryptographic algorithms and protocols designed to achieve certain security objectives, such as confidentiality, integrity, authenticity, or non-repudiation. The basic aim of cryptography is to construct schemes or protocols that can still accomplish certain tasks even in the presence of an adversary. A basic task in cryptography is to enable users to communicate securely over an insecure channel in a way that guarantees their transmissions' privacy and authenticity. To ensure the security of sensitive data, password encryption mechanisms are used [1]. But processing and searching encrypted data has become a complex problem [1]. Two methods were developed to solve these problems. The first method requires downloading all the ciphertext data, decrypting it locally, and then finding the desired result in the plaintext data. These methods require downloading many files, waste network bandwidth, and require high computational efficiency while considering the time and space efficiency of decryption [2]. The other method is to have the server decrypt the data, execute the query server-side, and then just send the results back to the intended user [2]. In this method, the target user sends a secret key to the cloud service provider to decipher the query. This allows the server to learn the queries to reduce encryption. Instead, it makes sense to support as complete a search function as possible on the server side without decrypting the data. This is called searchable encryption [2].

Based on the keys used for encryption and decryption, searchable encryption can be divided into symmetric encryption and asymmetric encryption or public key encryption. In symmetric searchable encryption, the same key is used for both encryption and decryption. This depends on the shared key agreement between communicating parties. In asymmetric searchable encryption, a different key is used for encryption and decryption. It is also known as public-key encryption and does not require a shared key. Song *et al.* [3] provides a practical symmetric searchable encryption scheme that becomes the forefront of searchable encryption development. The scheme is based on a sequential scan of the file without an index. Following Song *et al.* [3], Camara *et al.* [4] and Wang *et al.* [5] proposed searchable encryption schemes based on

symmetric cryptography. In symmetric searchable encryption, both parties need to exchange keys before decrypting. To solve the limitations of symmetric searchable encryption, Boneh *et al.* [6] first proposed public key searchable encryption (PEKS). The PEKS does not require a prior key agreement between the data owner and the target user. Data owners use the public key to the target users to generate encrypted texts containing encrypted documents and encrypted keywords. Then upload encrypted text to the cloud service provider. When the target user needs to search for a keyword in the ciphertext, it will use its key to generate a keyword search trap and send it to the cloud service provider. The cloud service provider performs a check operation and returns ciphertext containing the target keyword.

The Certificate-less Searchable Encryption with Keyword Search (CL-SEKS) scheme is a public-key cryptosystem based on identity encryption [7]. The private key in certificateless public key encryption is no longer generated separately by the public key generator (PKG) (the device or program used to generate the golden key) but is jointly generated by the PKG and the target user [8]. Whereas Al-Riyami *et al.* [8], proposed identity-based credential-free public key cryptosystem encryption scheme overcomes the key escrow problem in identity-based search encryption, where a fully trusted private key generator knows the private keys of all users. This is because it supports PKG to generate partial private keys for the target user and provides the ability to generate secret keys known only to the target user. In addition, this solution maintains the benefits of certificate-less and overcomes the certificate management problem based on PKI encryption.

Pairing-based cryptography (PBC) is a branch of cryptography that uses a pairing (mapping) function and elastic mathematical tool to design and develop new cryptographic protocols, which are often based on new security assumptions [9]. The pairing (mapping) function of the point on the elliptic curve is the foundation upon which PBC is built [9]. It enables the efficient computing of a wide range of distinct cryptographic algorithm implementations [10]. It presents the application of various elliptic curve varieties that are both cryptographically secure and effectively calculable [11]. It depends on the algebraic structure of elliptic curves over finite fields and is an extension of elliptic curve cryptography (ECC), which is the most effective public-key cryptography that employs shorter key lengths [12].

In many areas of cryptography, PBC helps to solve problems that are impractical, challenging, or inefficient when using traditional public-key cryptography or symmetric encryption [13]. It offers many advanced cryptographic primitives such as enhanced privacy ID (allows hardware devices to authenticate to a remote system), identity-based cryptography (support a to encrypt a message without needing public key to have been certified and distributed in advance), attribute-based encryption (individual can encrypt a message in such a manner that anyone possessing a particular set of attributes, which can be defined at the time of encryption, could decrypt the message) [9]. PBC is also utilized in searchable encryption [13]. This refers to several techniques that enable searching an encrypted database without having to decrypt the database. It can be used for various authentication schemes, privacy-preserving auctions, privacy-friendly aggregation for the smart grid, network communication resilient to traffic analysis, anonymous credentials, and many more.

The certificate-less based searchable encryption with refreshing keyword search is public-key cryptosystem that rely on the identity-based public-key cryptosystem and refreshing keyword search. The idea behind refreshing keyword search is to generate a search keyword that is only valid in a specified time interval, the time interval in the trapdoor, the server cannot use the keyword to search in ciphertexts.

Thus, the purpose of this research is to design certificate-less based searchable encryption with refreshing keyword search that allow to refresh search keyword using pairing-based cryptography. This needs to bound the duration of time the search keyword remains active and to refresh keywords every time searching is conducted, by attaching period information to encrypted files and keywords.

1.2 Motivation

The basic motivation for conducting this thesis work is the absence of specifying the time in which the trapdoor is active for conducting tests to know which keyword is used in the given trapdoor. In a keyword-guessing attacks, the attacker challenges each possible keyword, encrypts it, and tests the ciphertext with the given trapdoor. If the test succeeds, the attacker can identify

which keyword is used in the given trapdoor. Since people usually select keywords that are used regularly and easy to memorize, the server can discover the underlying keyword [14].

Suppose a user **Alice** sends an encrypted data attached with the ciphertext of a keyword to another user **Bob**. When user **Bob** wants to extract the information, he generates his trapdoor and sends it to the target server. After receiving the trapdoor, the target server tests the search function on the keyword and returns a search result to the data user **Bob**. In this way, the adversary may know the theme of the data and thus the user's privacy is leaked. Knowing this, insider can do a keyword guessing attack and it may be able to access the ciphertext through a lengthy trial for unlimited time.

Integrating certificate-free searchable encryption with refreshing keyword search will improve the security of searchable encryption with keyword search. This can be achieved by making sure that every time the user Bob generates his trapdoor and extracts information from the target server, the trapdoor information expires after some time defined by the user Bob.

1.3 Statement of the Problem

Al-Riyami *et al.* [8] proposed certificate-less public cryptography based on identity-based public key cryptography. It overcomes the key escrow problem that exists in identity-based searchable encryption and does not involve a third party holding the private key. This ensures that no other entity, including the PKG, has access to the user's private key, strengthening the privacy and security of the encryption system. This is because it supports PKG to generate partial private keys and provide secret keys to target users. This secret key is known only by the target user. Additionally, the scheme reserves the certificate-less benefit, it alleviates certificate management issues based on PKI encryption. The certificate management issues in PKI encryption are certificate issuing, renewal, revocation, replacement, storing, distribution and monitoring.

In spite of the certificate-less searchable cryptography alleviating the certificate management issues in PKI encryption, there are remaining enormous issues and challenges. These issues are when the data owner performs encryption, it encrypts both keywords and data using the target user's public key and identity [15]. Then, it sends the cipher text to the target server. Then, the

target user produces a search trapdoor using its complete private key and sends it to the target server to perform a keyword search. Whenever the data user sends the trapdoors, the server can store a trapdoor and it can use it to search in ciphertexts [16]. Using the trapdoor stored in the system, an adversary can launch a keyword guessing attack by checking all encrypted files and keywords [17]. Although certificate-less encryption has been proposed for keyword searching [15, 16, 17], the search keyword has never been updated. Therefore, their scheme is vulnerable to keyword guessing attacks, which allows adversaries to recover keywords from trapdoors [7].

Following Boneh *et al.* [6] Baek *et al.* [16] proposed a keyword guessing attack concept and a secure changeless PEKS scheme. The concept of keyword guessing attacks recognizes that the keyword space used in practice is limited. This is because people usually choose keywords that are easy to remember. The concept of no secure channel is to eliminate the secure channel between the data user and the target server. In addition, they didn't prove that the proposed PEKS schemes are resist keyword word guessing attacks. Later, Wu *et al.* [18] showed that Certificate-less Designated Server Based Public Key Encryption with Keyword Search (CL-dPEKS) schemes are vulnerable to keyword guessing attacks on ciphertext and trapdoor by the outside adversary. Wu *et al.* [18] also addressed the need to conduct research works to design a new security model and secure scheme to overcome the known attacks in certificate-less encryption with keyword searchable.

To alleviate the issue of the target server storing trapdoor, Baek *et al.* [16] and Bosch *et al.* [2] recommended that *"refreshing keyword will improve the vulnerability of keyword guessing attack on searchable encryption"*. Baek *et al.* [16] recommended future works on finding an efficient and suitable way to refresh regularly used keywords. The work of refreshing keyword search is to produce a trapdoor that is only valid in a specified time interval [2, 10]. Considering the target system, the data owner or sender defines the time interval for trapdoor generation by the data receiver or user. Every time the data receiver produces its trapdoor and searches for information on target system, the trapdoor information is made to expire after the time well-defined by the data owner or sender. Both Bosch *et al.* [2] and Baek *et al.* [16] recommended that the refreshing keyword search will enhance the security of searchable encryption with keyword search, such as PKI-based, identity-based, attribute-based and certificate-less

searchable encryption. However, there is no research work done for certificate-less searchable encryption in which the time for keyword search generation for the keyword is limited.

With this respect, in this thesis work, we will propose a certificate-less searchable encryption schema with refreshing keyword search which preserves certificate-less advantages, eliminates key escrow problems and refreshes the keywords when the specified threshold time is reached.

1.4 Objectives

General Objective

The general objective of this thesis is to design and develop searchable encryption scheme that allow to refresh search keyword and based on certificate-less public key encryption using pairing-based cryptography.

Specific Objectives

To achieve the stated general objective, the following specific objectives will be performed.

- To review related literature done around searchable cryptographic encryption and refreshing keyword searches,
- To design the system model for the proposed scheme,
- To design the algorithm for the proposed scheme,
- To develop a prototype for validating the proposed scheme,
- To evaluate the time complexity of the proposed SERKS scheme.

1.5 Methods

The following methods will be employed to conduct this thesis work.

Literature review

A review of literature in various fields of cryptography such as encryption, search, security, refreshing search, and data outsourcing will be made for broadening understanding, formulation of the problem, and adoption of some of the concepts and methods.

Algorithm Design

The design of pairing-based cryptographic algorithms often involves a deep understanding to determine the specific cryptographic primitives that can be constructed using pairing-based cryptography, such as identity-based encryption, attribute-based encryption, signature schemes, and key exchange protocols. It includes establishing the overall setup of the pairing-based cryptographic system, including the trusted authority (e.g. key generator), the public parameters, and key generation process. It also requires careful consideration of the trade-offs between security, efficiency, and functionality to create practical and secure solutions.

Selection of Elliptic Curves

Pairing-based cryptography relies on the use of bilinear pairings, which are efficiently computable maps between points on an elliptic curve and a target group. The selection of the appropriate elliptic curve for pairing-based cryptography depends on the specific need of the application, the desired security level, and the performance characteristics of the available options. Careful analysis and evaluation of the trade-offs are necessary to choose the most suitable elliptic curve for a given cryptographic system.

Prototype Development

To demonstrate the proposed SERKS scheme, a prototype will be developed in the case of the selected system. Java Pairing-Based Cryptography (JPBC) library and Java programming

language will be used to develop the prototype. NetBeans IDE 8.0.2 tool will be used for programming Java code with the JPBC library.

Performance Evaluation

The security of the designed algorithm will be based on the Bilinear Diffie-Hellman (BDH) hardness assumption and evaluate the time complexity of the designed algorithms. The efficiency of the proposed algorithm will be evaluated by conducting computational cost analysis, which is the execution time of the proposed algorithm and the number of required rounds to retrieve the matching documents, and communication cost analysis, which is the size of public keys, ciphertext and trapdoor.

1.6 Scope and Limitations

The scope of this study is to design SERKS scheme by integrate a certificate-less based encryption that allow to refresh search keyword using pairing-based cryptography. In so doing, we will design an algorithm by considering text data to perform encryption and keyword search functions. The security of the designed algorithm will be based on the hardness Bilinear Diffie-Hellman (BDH) problem assumption. In this research work, we don't consider videos or images as well as the hardness of lattice-based assumptions.

1.7 Application of Results

The SERKS provides a way to improve the security of the search keyword of searchable encryption. Some of the applications are listed below:

- **Public Key Infrastructure (PKI) Based Public Key Encryption with Keyword Search:** is a type of public key encryption in which the legitimacy of target data using a public key is identifiable by the target data owner before encrypting data using his/her public key and outsourcing to the cloud service provider. The refreshing keyword search can enhance the security of Public Key Infrastructure (PKI) based PEKS by limiting the lifetime of trapdoors on the target servers when target users perform a keyword search.

The target server performs the test within the specified time interval, so it cannot use the search trapdoor in the past or future outside the time interval.

- **Attribute-Based Public Key Encryption with Keyword Search:** is a type of PEKS in which a selected attribute of the target data user is used as a public key. It regards identity as a series of attribute sets; attributes are the information elements of the user. The target data owner encrypts data using the selected attribute of the target data user and outsources it to the cloud service provider. The refreshing keyword search can enhance security attribute-based PEKS by limiting the lifetime of trapdoors on cloud servers when target users perform a keyword search. The cloud server provider performs the test algorithm only in a specific time interval, so it cannot use the search trapdoor in the past or future outside the time interval.
- **Identity-Based Public Key Encryption with Keyword Search:** This is a type of PEKS in which the keyword acts as the identity. The refreshing keyword search can enhance security identity-based PEKS by limiting the lifetime of trapdoors on cloud servers when target users perform a keyword search. The target server performs the test algorithm only in a specific time interval, so it cannot use the search keywords in the past or future outside the time interval.

Therefore, cyber security practitioners, researchers, government offices and even individuals will be the users that directly or indirectly benefit from this research work.

1.8 Organization of the Rest of the Thesis

The remaining chapters of this thesis are organized as follows. In Chapter Two, we review the basic design of searchable encryption algorithms, security models, and tools used in existing works done on searchable encryption. Chapter Three discusses related works done regarding searchable encryption based on public key encryption. Chapter Four discuss the proposed system model, algorithm design, algorithm computational consistency and workflow. In Chapter Five, the prototype of the proposed scheme is implemented, and its performance is evaluated in terms of computational cost, communication costs and security properties. Finally, Chapter Six gives the conclusion, contribution and future work of the thesis.

CHAPTER TWO: LITERATURE REVIEW

In this Chapter, we review the basic design of searchable encryption algorithms, security models, and tools used in early work on searchable encryption. It provides an over-all background regarding the area of study as well as the tools used to be used.

2.1 Overview

Cryptography refers to the concept and implementation of engineering cryptographic systems, i.e., encryption and decryption engines, digital signature and authentication hardware and software systems, key generation, distribution, and management systems, and random number generators [19]. It is used everywhere and by billions of people worldwide daily to protect data at rest and data in motion. It refers to the design of mechanisms based on mathematical algorithms that offer important information security services.

The searchable encryption (SE) is a cryptographic mechanism that allows to perform search on encrypted data. From the perspective of cryptography, searchable encryption mainly classified into two types, the first one is symmetric encryption keyword searchable (SKEKS), and the second is public-key encryption with keyword search (PEKS).

2.2 Symmetric Key Encryption Keyword Searchable (SKEKS)

In symmetric key searchable encryption, the same key is used for both encryption and decryption. This depends on the shared key agreement between communicating parties. An important turning point in the evolution of searchable encryption was reached when Song et al. [3] offered a workable searchable encryption technique. The plan is based on scanning the document sequentially without an index. Although the search process is straightforward, the cloud necessitates full-text scanning, and keyword searches are not indexed. Statistics may also be used by the server to gather more user data.

A workable dynamic searchable encryption approach that facilitates update operations—that is, adding files to an existing index and removing files from it—was put forth by Kamara *et al.* [4]. It is possible to add new terms to an index's dictionary and update the index's posting lists in a dynamic searchable encryption system. We keep two indexes to add new documents rapidly: a small auxiliary index that points to new documents and is kept in the proxy, and a large main

index that is kept in the index server. In the index server, the auxiliary index is combined with the primary index when it surpasses a certain threshold. When integrating the auxiliary index into the main index in the SKEKS scheme, they must also pad.

A workable dynamic searchable encryption approach that facilitates update operations—that is, adding files to an existing index and removing files from it—was put forth by Kamara *et al.* [4]. It is possible to add new terms to an index's dictionary and update the index's posting lists in a dynamic searchable encryption system. We keep two indexes to add new documents rapidly: a small auxiliary index that points to new documents and is kept in the proxy, and a large main index that is kept in the index server. In the index server, the auxiliary index is combined with the primary index when it surpasses a certain threshold. When integrating the auxiliary index into the main index in the SKEKS scheme, they must also pad

IDCrypt is a workable symmetric searchable encryption technique with proved security strength that was proposed by Wang *et al.* [5]. To increase searchable encryption security and search efficiency, they suggested IDCrypt. The SKEKS schemes on IDCrypt can be readily implemented in real application and are compatible with commercially available index-based search engines. To improve search efficiency and the security strength of searchable encryption, Wang *et al.* [5] implemented an index server in their IDCrypt architecture that can be set up using commercial index-based search engines to offer search operations on encrypted documents. Advanced searchable encryption features like dynamic searchable encryption and multi-keyword ranking searchable encryption are also supported. They created the Two-Layer Encryption Scheme (TLES) to safely and effectively transfer secret keys across many users, combining Identity-Based Encryption (IBE)

A SKEKS scheme proposed by Wang *et al.* [5] is a collection of four polynomial-time algorithms. These are key generation, building index, generating trapdoors and performing search operations. A secret key and a public key are produced by the probabilistic key generation algorithm, which accepts a security parameter. An index is produced by the index-building process using a secret key and a collection of documents as input. A secret key and a keyword are inputs to the trapdoor generating process, which then produces a trapdoor. The user runs it to conduct a search. Lastly, the search operation algorithm returns a search result keyword or the number of documents after receiving an index and a trapdoor for the keyword as inputs.

2.3 Public key Encryption with Keyword Search (PEKS)

In 2004, Boneh *et al.* [6] introduced public encryption with keyword search (PEKS). Public-key encryption methods provide the foundation of PEKS. The three primary components of the PEKS are the cloud service provider, data transmitter, and data receiver. A data sender sends their documents and index to the remote service provider after encrypting them using the public key. The search function can be carried out by a data receiver who has obtained a matching private key. By using the private key to search for the phrase, the user creates the trapdoor and transmits it to the server. Without knowing the plaintext message of the encrypted data and the keyword, the cloud service provider receives the trapdoor and runs a test to determine whether a given ciphertext contains the search term. The query results are then sent back by the service provider to.

The searchable encryption proposed by Boneh *et al.* [6] mainly contains four probabilistic polynomial-time algorithms:

Key Generation(k): a key generation algorithm run by the data receiver. It takes a security parameter λ as input and outputs public key PK and private key SK.

PEKS (PKR, w): an encryption algorithm run by the data sender. The algorithm takes the receiver's public key PKR and a keyword w as input, producing as an output a PEKS cipher text S which is searchable encryption of w . Then it writes likewise

$$S = \text{PEKS}(PKR, w).$$

Trapdoor (SKR, w): a keyword trapdoor generation run by the data receiver. It takes a receiver's private key SKR and a keyword w as input and outputs a trapdoor Tw for the keyword w .

Test (Tw, S): the algorithm takes a trapdoor Tw for a keyword w and a PEKS cipher text.

$S = \text{PEKS}(PK, w')$ as input, the algorithm returns a result "yes" if $w = w'$; otherwise, it returns "no".

There are four main variants in the PEKS scheme. These are PKI-based encryption with keyword search, identity-based encryption with keyword search, attribute-based encryption with keyword search and certificate-less encryption with keyword search.

a. Public Key Infrastructure-based PEKS (PKI- PEKS)

The PEKS was proposed to address the distribution and administration of secret keys in symmetric searchable encryption [6]. The majority of PEKS schemes are based on certificate management and public key infrastructure (PKI). After verifying the validity of the recipient's public key, the sender can use that key to encrypt the data and keywords. The PKI, the third authority, generates the public key.

Bilinear pairing was used by Boneh *et al.* [6] to propose the PKI-based PEKS. In this technique, the receiver's public key can be used to enable the target user to produce searchable information. But the only person who can make a trapdoor for keywords to query is the owner of the private key.

Baek *et al.* [16] re-visited the PKI-based keyword search system proposed by Boneh *et al.* [6] and proposed a new PKI-based PEKS scheme that alleviated the need for a secure channel. Their implementation requires a server public key and private key, and only the server chosen by the sender can perform search. They demonstrated security in the random oracle model under the BDH hardness assumption. Moreover, the authors introduced a method of multiple keywords PEKS outline which indicates that encoded conjunctive keywords effectively with limited amounts of the keyword. Since no secure channel is wanted, an impersonation attack is not feasible in a PKI system. Their scheme also supports conjunctive and disjunctive search, respectively. However, Tsu-Yang Wu *et al.* [18] demonstrated that Baek *et al.* [16] scheme is vulnerable to keyword-guessing attacks. The structures based on PKI keyword search systems have a serious issue of certificate management including certificate verification, certificate distribution, and certificate revocation.

b. Identity-based PEKS (IB- PEKS)

Shamir [20] initially proposed identity-based encryption (IBE) in 1984. It simplifies the maintenance of public keys and certificates in conventional PKI-based public-key encryption. A user's public key in an identity-based encryption technique is frequently an arbitrary string, such as their ID, phone number, email address, or IP address. Many PEKS-IBKS have been proposed since Shamir's breakthrough. PKI-based encryption's certificate management issue is resolved by the scheme. However, because a fully trusted private key generator can know every user's private key, the present PEKS-IBKS methods have key escrow concerns [15].

c. Attribute-based encryption PEKS (AB-PEKS)

Sahai and Waters [21] introduced attribute-based encryption (ABE), commonly referred to as fuzzy identity-based encryption, in 2005. It is about identity as a collection of sets of attributes. These characteristics constitute the user's informational components. By implementing an access control policy on the encrypted document that is outsourced, the Attribute-Based Keyword Search (AB-PEKS) technique enables the sender to grant its search function to other data consumers or recipients.

When using attribute-based encryption, the sender just needs to create an access policy that several users can follow before encrypting a document just once to produce ciphertext. A data owner can offer receivers access to its search capabilities by implementing an access control policy on the ciphertexts that are outsourced. The properties are connected to the user's private key and the ciphertext document. The user can decrypt cipher-texts when their characteristics align with the relevant access policies.

The two types of attribute-based encryption are cipher-text-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE), depending on whether the private key or cipher-text is linked to the access control policy.

An idea based on ABE was proposed by Han *et al.* [22], who also examined converting ABE into PEKS-ABE. Based on KP-ABE, they created a concrete PEKS-ABE scheme that accommodates many users.

A cipher-text-policy attributed-based encryption technique was presented by Wang *et al.* [23] to facilitate keyword search capabilities. Only authorized user receivers who comply with the policy can retrieve the keyword and decrypt the ciphertext, giving the data owner control over their data through access policy.

Because of its effective data sharing and finding capabilities, attribute-based encryption is frequently used in PEKS [23]. However, because the PKG knows the private keys of every user, attribute-based encryption with the PEKS scheme also concurrently introduces the key escrow issue.

d. Certificate-less encryption with keyword search

The first certificate-less encryption, a kind of public-key cryptosystem based on the identity-based public key, was proposed by Al-Riyami et al. [8] in 2003. In certificate-less public key encryption, the PKG and the user jointly produce the private key rather than the PKG doing so independently. The issue of key escrow in identity-based encryption was mitigated by certificate-less encryption since the PKG is unaware of the secret key, which is only known by the target user.

The proposed certificate-less public key cryptography algorithms in [8] is summarized as follows:

Setup: the algorithm takes security parameter k as an input and public parameter $prms$, and master-key ms as the output.

Extract Partial Private Key: the KGC algorithm takes public parameter $prms$, master key ms and identity of the target user ID_i , $ID_i \in \{0, 1\}^*$ as an input and outputs a private partial key.

Set Secret Value: the algorithm takes public parameter $prms$ and target user identity ID_i as inputs and produces a secret value.

Set Private Key: the algorithm takes public parameter $prms$, private partial key and secret value as input, and output private key.

Set Public Key: the algorithm takes public parameter $prms$ and secret value as input and outputs the public key.

Encrypt: The algorithm takes public parameter $prms$, a message $M \in M$, the public key and the identity of the target user as inputs and a cipher-text C as output.

Decrypt: the algorithm takes public parameter $prms$ as, cipher-text C and private key as input and message M as the output.

2.4 Elliptic Curve Cryptography (ECC)

In 1985, Neal Koblitz and Victor Mille made the discovery of elliptic curve cryptography (ECC) [12]. Among asymmetric cryptographic primitives, ECC is a form of Public Key Cryptography (PKC) that offers strong security with reduced key sizes [24]. It is a public key cryptography system in which the sender's private key is used to encrypt the message, and the sender's public

key and the recipient's private key are used to decrypt it [25]. At the same security level as other commonly used systems like RSA (Rivest, Rivest, and Adleman), it provides more effective implementations and smaller key sizes.

The first generation and most useful asymmetric-key encryption is RSA [26]. The integer factorization problem is the source of the RSA algorithm's security [26]. An ECC-based cryptosystem is superior to one based on RSA or its variations, and because it uses fewer resources than an RSA-based cryptosystem, it is most appropriate for devices with limited memory [27]. Discrete logarithmic issues on the elliptic curve are also challenging to solve. ECC is becoming more and more common than the RSA algorithm because of the aforementioned benefits.

The base point, which is a point on the curve, and the coefficients of the elliptic curve make up the elliptic curve over the prime field [25]. The chosen curve demonstrated that the communication parties were aware of it. Scalar multiplication, which includes point addition and point doubling, is the fundamental operation in ECC. To cut down on the expensive inversion process, all fundamental operations are carried out in projective coordinates.

A. Elliptic Curve Cryptography Point Addition

Assuming that P and Q are the two points on the curve, adding them yields $P+Q$, or $R=P+Q$. Drawing a line connecting points P and Q is the fundamental method of adding points. This line crosses $-R$, a point on the curve. The real result R is the inverse of this point.

B. Elliptic Curve Cryptography Point Doubling

If P is a point on the curve, then $R=2P$ is the outcome of doubling. A tangent drawn on the point P provides a graphical representation of the computation. The intersection of this line yields the point $-R$. The actual point R is the inverse of this point.

The security hardness of the ECC is defined by the following hard problems [28].

- 1) *Elliptic curve discrete logarithm problem (ECDLP)*: Given $G, Q \in G_q$, where G is the group generator and Q is an element of G_q , it is computationally hard to find an integer k , such that $Q = kG$, where $k \in \mathbb{Z}_q^*$.
- 2) *Elliptic curve computational Diffie–Hellman problem (ECCDHP)*: For any $c, d \in \mathbb{Z}_q^*$, it is computationally hard to compute cdG from the given instance $(G, cG, dG) \in G_q$.

- 3) *Elliptic curve decisional Diffie–Hellman problem (ECDDHP)*: For any $c, d, z \in \mathbb{Z}_q^*$, it is computationally hard to distinguish the two instances (cG, dG, cdG) and (cG, dG, zG) of G_q , i.e., whether $cdG = zG$ or not.

The capacity to compute a point multiplication and the inability to compute the multiplicand given the original and product points are prerequisites for the security of elliptic curve cryptography [12].

2.5 Pairing Based Cryptography

The foundation of Pairing-Based Cryptography (PBC) is the pairing (mapping) function of the elliptic curve point [9]. Various kinds of cryptographic methods that are computationally efficient can be implemented with it [10]. It provides the use of several elliptic curve types that are both cryptographically secure and efficiently calculable [11]. It is an extension of elliptic curve cryptography (ECC), which is based on the algebraic structure of elliptic curves over finite fields and is the most effective public-key cryptosystem with reduced key lengths [12]. Some of the commonly used elliptic curves in pairing-based cryptography include:

Barreto-Naehrig (BN) Curves: These are a group of pairing-friendly curves with an embedding degree of $k = 12$, providing a good balance between security and efficiency.

Barreto-Lynn-Scott (BLS) Curves: These are another group of pairing-friendly curves, with an embedding degree of $k = 12$ or $k = 24$, offering even higher security levels.

Super-singular Curves: Certain super-singular elliptic curves, such as the MNT curves, can also be used in pairing-based cryptography, with an embedding degree of $k = 6$. The MNT curves are named after their creators Miyaji, Nakabayashi and Takano.

A. Bilinear Pairing (e)

A function " e " that takes two points on an elliptic curve and returns an element in a finite field is known as a bilinear pairing [13]. G_1 is the group of points of an elliptic curve over $(\mathbb{F}_p)$, and G_T is a subgroup of $\mathbb{F}_p^*$ bilinear pairing (e). Let G_1 be an additive group and G_T be a multiplicative cyclic group (a group generated by a single element) having the same order q for some large prime q . The formula for G_T is $e: G_1 \times G_1 \rightarrow G_T$ map [29]. The following properties are met by this mapping.

- 1) *Bi-linearity*: $\forall a, b \in \mathbb{Z}_q^*$ and $\forall P, Q \in G_1$, $e(aP, bQ) = e(bQ, aP) = e(P, Q)^{ab}$, where

Zq^* is a cyclic group of a prime number.

- 2) *Non-degenerate*: there exists $P, Q \in G_1$ so that $e(P, Q) \neq 1 \in G_T$.
- 3) *Computable*: $\forall P, Q \in G_1$, there is an efficient algorithm to compute $e(P, Q)$.

Bilinear pairing operations can be calculated using the Miller algorithm or the Miller loop, the Weil pairing, the Tate pairing, or other pairing types (e.g., Ate, Eta, O-Ate) that compute pairings on pairing-friendly elliptic curves [20]. The most popular methods for creating bilinear mappings $e: G_1 \times G_1 \rightarrow G_T$ and bilinear pairings between two groups are the Tate and Weil pairings. In practice, the Tate pairing is chosen since it is typically faster than the Weil pairing [20].

One common mathematical structure that can be applied to cryptography is the bilinear pairing [30]. Because of their equational characteristics, cryptographic primitives and protocols can be created that would otherwise be impossible or ineffective [25]. Cryptographic protocols including identity-based encryption, aggregate signatures, and one-round three-party key agreement have all been designed using it.

B. Bilinear Diffie-Hellman problem assumption

The Bilinear Diffie-Hellman problem (BDHP) is to calculate $e(P, P)^{abc}$ from the given P, aP, bP, cP , where a, b, c are random elements of integer number(Zq^*) and P is the generator of the cyclic group, a group that is generated by a single element [29]. Let e be a bilinear pairing on (G_1, G_T) , where G_1 is an additive group and G_T is a multiplicative cyclic group. Additionally, the following is a description of it [18]:

Assume that we have a generator P of G_1 and some known points $P, aP, bP, cP \in G_1$ as input wherein $a, b, c \in Zq^*$ are random numbers, then required to calculate or compute $e(P, P)^{abc} \in G_T$. When any polynomial-time approach has a negligible advantage (ϵ) in calculating Bilinear-Diffie-Hellman problems, as demonstrated by equation 1, it can be said that Bilinear Diffie-Hellman problems are intractable.

$$(Pr[(aP, bP, cP) = e(P, P)^{abc}]) \leq \epsilon \quad (1)$$

Where pr represents the probability

The difficulty of the BDHP implies the hardness of the Diffie-Hellman Problem (DHP) in both G_1 and G_T . First, if the DHP in G_1 can be efficiently solved, then one could solve an instance of

the BDHP by computing abP and then $e(abP, cP) = e(P, P)^{abc}$. Also, if the DHP in G_T can be efficiently solved, then the BDHP instance could be solved by computing:

$$P = e(P, P),$$

$$P^{ab} = e(aP, bP),$$

$$P^c = e(P, cP) \text{ and then } P^{abc}.$$

In this case, the BDHP is thought to be as difficult as the Diffie-Hellman Problem (DHP) in G_1 and G_T , assuming that nothing else is known about its intractability.

C. Pairing-Based Cryptography (PBC) library

The PBC library is a C programming library available free and distributed under the GNU Lesser General Public License, is constructed based on the GNU Multiple Precision Arithmetic (GMP) library to execute the mathematical computations that form the foundation of pairing-based cryptosystems [11]. The PBC library has been strategically developed to serve as the fundamental framework for the execution of pairing-based cryptosystems, with a clear focus on achieving efficiency and adaptability. Key objectives include prioritizing speed and portability within its design. This library offers a range of essential functionalities, including the generation of elliptic curves, operations involving elliptic curves, and computations related to pairings.

The PBC library is available for different platforms and programming languages. Among them, pypbc library is available for python and JPBC is for Java programming language. The Java Pairing-Based Cryptography Library (JPBC) offers a Java-based adaptation of the Pairing-Based Cryptography Library (PBC) originally developed by Ben Lynn [31]. This adaptation facilitates the direct execution of the mathematical operations that serve as the foundation for pairing-based cryptosystems within the Java environment. The following curves are those that JPBC supports [31]:

- 1) Type A: are super-singular curves that have the form $y^2 = x^3 + ax$, for any $a \in F_q$, have embedding degree 2. Embedding degree k represents the smallest positive integer such that the elliptic curve points can be embedded into the multiplicative group of the finite field

- 2) Type A1: The same curve as for Type A pairings can be used to discover a suitable pairing where the cryptosystems require the curve order to be a certain integer, such as $N = p \cdot q$, where p and q are huge primes, making N difficult to factorize.
- 3) Type D: are to construct curves with embedding degree 6.
- 4) Type E: these curves have embedding degree 1 and All computations necessary for the pairing can be done F_q alone.
- 5) Type F: Barreto and Naehrig found a parametrization that produced embedding degree 12 curves by taking cyclotomic polynomials into consideration
- 6) Type G: The embedding degree of these curves is 10.

Additionally, JPBC comes with a Wrapper that makes it possible to move pairing computation duties to the PBC library, improving speed all around. Additionally, JPBC uses Multilinear Maps, which improve primary memory utilization by supporting multithreading and utilizing memory-mapped files.

2.6 Semantic Security (SS)

To characterize the security of the PEKS scheme that was indistinguishably secure against an adaptive chosen keyword attack (IND-CKA), Boneh *et al.* [6] established the idea of "indistinguishability of PEKS against chosen keyword attack." To define IND-CKA security between an adversary and a challenger, they provided the following game.

Setup: To create the public key PK and private key SK , the challenger executes the key creation algorithm. It retains the private key SK while sending the opponent PK .

Phase 1: Using terms of its choosing, the adversary can adaptively request trapdoors from the challenger.

Challenge: To be challenged, Adversary A gives the challenger the keywords W_0, W_1 . The adversary always wins the game if W_0 and W_1 are not requested in Phase 1; otherwise, it is a simple attack. The challenger gives the adversary the ciphertext $C = \text{PEKS}(PK, W_b)$ after selecting a bit $b \in \{0, 1\}$ at random.

Phase 2: Apart from M_0 and M_1 , the opponent may continue to seek additional trapdoors for any keyword, same as in Phase 1.

Guess: If b equals b_0 , the adversary wins the game and outputs its guess of b_0 .

Let's also discuss two experiments, $\text{EXP}(0)$ and $\text{EXP}(1)$. Both a challenger and an adversary are required for each experiment $\text{EXP}(b)$. The challenger is aware that the adversary is attempting to predict which experiment it is in [32]. Only when the adversary prepares two messages, M_0 and M_1 , of identical length and transmits them to the challenger are communications permitted. The challenger produces a random key k and encrypts the message M_b and sends $E(k, M_b)$ to the adversary. Finally, the adversary guesses whether she/he was given the encryption of M_0 or the encryption of M_1 by outputting a bit $b' \in \{0,1\}$.

If an opponent cannot determine with a better probability than half whether the provided ciphertext is encryption of message M_0 or M_1 [33], the cryptographic encryption strategy is semantically safe. Adversary A 's semantic security advantage (Adv) over encryption E is as follows:

$$\text{AdvSS}[A, E] = |\Pr[M_0] - \Pr[M_1]| \in [0, 1]$$

Where AdvSS represents the advantage of semantic security

‘ A ’ represents an adversary

‘ E ’ represents encryption

‘ $\Pr$ ’ represents the probability

‘ M ’ represents the message

When $M_b = [A(\text{EXP}(b)) = 1]$. Here the event M_b happens when the adversary is an experiment b and guesses M_1 . The encryption E is semantically secure if for all "efficient" $\text{AdvSS}[A, E]$ is negligible [32].

Furthermore, ciphertext security can be effectively defined using the concept of semantic security. The attacker should only be able to view one encrypted message, and the ciphertext should only reveal the length of the original message.

2.7 Attacks in Searchable Encryption

Keyword privacy is one of the primary issues with searchable encryption. The keyword guessing attack (KGA) is a primary searchable encryption attack in which revealed keywords reveal related information and the user's interests [34]. Anyone can create a searchable ciphertext in KGA by selecting a keyword from the keyword space. The reason for this is that anyone with the receiver's public key can create the searchable ciphertext using any keywords. By determining whether the searchable ciphertext he or she has constructed matches the trapdoor, the attacker constantly tries to get the keyword from it. The attack is successful and the right keyword is found if they match; if not, the attacker keeps guessing. By performing this attack, an attacker may correctly guess the keyword encoded in each keyword trapdoor [14]. It can be categorized into Inside keyword-guessing attacks and Outside keyword guessing attacks [35].

The inside keyword guessing attacks (IKGA) refer to attacks carried out by malicious insiders who have some level of authorized access or privileges within a system. The attacker may be able to leverage their access and knowledge of the system to more effectively guess and test keywords against encrypted data. The insider may have additional information or capabilities that give them an advantage over an outside attacker. Adversaries such as untrusted servers might launch inside keyword-guessing attacks (IKGA) to guess keywords from the trapdoors [35].

The outside keyword guessing attacks (OKGA) are carried out by external actors who do not have any authorized access or privileges within the target system. Outside attackers typically must rely on more generic methods, such as using dictionaries or brute-force techniques, to systematically guess keywords. They do not have the same level of insider information or access that could aid an internal attacker. An external adversary such as eavesdropper can be the OKGA [35].

2.8 Security Proof in Pairing-based Cryptography

The following approaches are used to provide security proof in the context of pairing-based cryptography [10].

a. Security Model

Security models are used to estimate potential attacks in practice and theoretically show if a method is secure [36]. The capabilities of the attacker, the security objectives, and the

presumptions made on the underlying cryptographic primitives are all specified in a clearly defined security model. Since they are used to assess and characterize the attack capabilities of various adversaries, security models are crucial to cryptography [36].

Additionally, security models are employed to mimic the adversary's behavior and guarantee their objectives. By offering legitimate evidence of its corresponding security game, a certificate-less technique is said to be secure [37]. If the specified requirements are met, the adversaries attempt to submit questions to win a security game.

Two kinds of adversaries exist in certificate-less public-key cryptography [37]. Specifically, adversary type I, represented by A_1 , and adversary type II, represented by A_2 . A malicious but passive key generation center (KGC) is simulated by adversary A_1 posing as an outside attacker, and a dishonest data user by adversary A_2 posing as a key-generation-center attacker. The public key of the data user can be changed by the adversary A_1 . Adversary A_2 may be the system's master key, meaning that he or she is able to create any partial private key for a data user. The adversary A_2 can likewise ask for public keys and conduct requests for the extraction of private or secret keys, but he or she is unable to replace the data user's preferred public key.

b. Complexity Assumptions

The difficulty of mathematical issues like the Computational Diffie-Hellman (CDH) problem or the Decisional Bilinear Diffie-Hellman (DBDH) problem is frequently the basis for pairing-based cryptography [10]. The presumed intractability of these underlying complexity assumptions subsequently determines the security of pairing-based methods. In the framework of bilinear groups, where cyclic groups of prime order have a non-degenerate bilinear map, the DBDH issue is defined. A generator is used to define the CDH issue in a prime-order cyclic group.

c. Security Reduction

A security reduction, which demonstrates that if an efficient attacker can break the pairing-based scheme, then they can also efficiently solve the underlying complexity assumption. This reduction shows that breaking the pairing-based scheme is at least as hard as solving the assumed hard problem [10].

d. Formal Proofs

The goal of these formal proofs is to offer a strict assurance of the pairing-based scheme's security features, like unforgeability, confidentiality, and integrity. Usually, methods from computational number theory and complexity theory are used to formalize the security reduction into a mathematical proof.

e. Random Oracle Model

Bellare and Rogaway [38] created a mechanism called a random oracle model to provide rigorous security proofs of cryptographic protocols. It is the perfect oracle for simulating a cryptographic hash function that uses their output domains to provide a truly evenly chosen random response to each distinct query. Apart from recording its responses $H(M)$ and repeating the corresponding response if M is queried again, the oracle model makes an independent random pick for every inquiry.

A random oracle can also be described as a mathematical function mapping possible queries to random responses from their output domains. They are methods that are used in cryptographic proofs. They are usually employed when there are no defined implementable functions that provide the mathematical property needed by the proof. This is to mean, a cryptosystem where proofs that have been used, are secure in random oracles model else secure in standard models. In an implementation, random oracles are usually employed for modeling the cryptographic hash function in the protocols since random oracles do not cover practical implementations. Hence, a robust random assumption is crucial to the hash function's outputs. The proof demonstrates that an algorithm is secure by describing that attackers must require impossible performance from oracles to solve certain mathematical problems assumed not feasible to break.

Generally, we may not identify how to construct an efficient algorithm that is provably secure, depending on the standard cryptographic assumption, the standard that is well-accepted in the crypto community [38]. Hence, to have an effective algorithm, it is necessary to use these random oracles model and to prove the security hardness of cryptographic construction in this model.

CHAPTER THREE: RELATED WORK

In this Chapter, we discuss related works done on searchable encryption. The two basic encryption approaches for achieving searchable encryption. These are Public Key Infrastructure (PKI)-based searchable encryption and certificate-less based approaches. The PKI-based searchable encryption relies on a trusted certificate authority to issue digital certificates that bind public keys to identities. The certificate-less-based searchable encryption eliminates the need for a traditional PKI infrastructure and instead uses a two-tier key generation process involving a key generation center (KGC) and a data user. The different related work for both encryption approaches for different applications are presented in the below sections.

3.1 PKI-Based Searchable Encryption

The PKI-based searchable encryption uses PKI infrastructure, which requires a trusted certificate authority (CA) to issue and manage certificates. Related work that used this approach is discussed in this section.

Boneh *et al.* [6] proposed the public key encryption with keyword search (PEKS) scheme which enables one to search encrypted keywords without compromising the security of the original data. They used bilinear pairing on elliptic curves to build the scheme in the cloud mail server. A cloud mail server, a data sender, a data receiver, and a key generating center (KGC) are the four entities that make up the suggested system paradigm. Both the public key and the private key for a data sender and a data recipient are generated by the KGC. She or he uses the keys to create a trapdoor that the cloud mail server or mail gateway should look for. The provided trapdoors are fed into the test procedure by the cloud mail server. Their research work only supports a single keyword search and requires a secure channel for encryption key exchange.

The public key encryption with keyword search (PEKS) system was proposed by Baek *et al.* [16] in response to the work of Boneh *et al.* [6]. It addresses the issue of the secure channel and allows for numerous keyword searches. The PEKS ciphertexts for numerous keywords are efficiently generated by the multiple keyword searches. It is known as PKI-based public key encryption with keyword search (PEKS) and the encryption keys are generated by a third party called Public Key Infrastructure (PKI). They also introduced the idea of renewing keyword

search (RKS), which involves adding time information to commonly used terms and setting them to expire after a predetermined amount of time.

The proposed work by Baek *et al.* [16] has the problem of certificate management such as certificate creation, revocation, validation, assurance, renewal and enrollment due to the public keys being generated by the public key infrastructure (PKI).

3.2 Certificate-less-Based Searchable Encryption

The searchable encryption based on certificate-less encryption eliminates the need for a PKI infrastructure and instead uses a two-tier key generation process involving a key generation center (KGC) and a data user. The KGC generates a partial private key, and the data user generates a secret key using its own information and the partial private key. Related work that used this approach considering different applications are discussed in this section.

3.2.1 Generic Searchable Encryption

Zheng *et al.* [39] combine a keyword search on encrypted data with certificate-less cryptography. They demonstrated its security using the elliptic curve cryptography's computational hardness assumption, the Decisional Linear Assumption (DLA). To carry out the mathematical operations that underlie pairing-based cryptosystems directly in Java, they built the CLKS scheme using the Java Pairing-Based Cryptography Library (JPBC). They used Type A pairing to instantiate the bilinear map, providing a security level comparable to 1024-bit Discrete Logarithm security. The work of Zheng *et al.* [39] has the drawback that the keyword trapdoor is not updated and is susceptible to keyword-guessing attacks.

3.2.2 Certificate-less Searchable Encryption in the case of Email System

A crucial component of searchable encryption for data protection and operability, certificate-less public key encryption with keyword search was first proposed by Yanguo *et al.* [15]. For the first time, the writers solved the key escrow issue. They created a secure channel-free CL-PEKS scheme and designed a certificate-less PEKS (CL-PEKS) scheme for an email system. Under the Bilinear Diffie-Hellman assumption, they demonstrated its security. The technique was later discovered to be susceptible to assaults employing a rogue key-generation-center and keyword guessing attack [40], and the work's weakness is that it does not handle multiple keyword searches. The vulnerability of Yanguo *et al.* [15]'s certificate-less searchable public-key

encryption system to keyword-guessing attacks was demonstrated by Wu *et al.* [40]. Furthermore, the keyword trapdoors are never updated.

3.2.3 Certificate-less Searchable Encryption in the case of IOT

For the Internet of Things, Ma *et al.* [41] created the secure channel-free certificate-less searchable public-key encryption with multiple keywords (SCF-MCLPEKS) system. Four entities make up the proposed SCF-MCLPEKS scheme: a key generation center (KGC), a receiver, a data owner, and a cloud server. System keys and partial private keys for the server and receiver are generated by the KGC. The data owner encrypts the data and its index of keywords using the public keys of the server and receiver. The encrypted data and encrypted keyword indexes on the cloud server can then be outsourced by the data owner. The KGC provides the recipient, who is a data user, with his or her partial private key. After that, it creates a trapdoor using the desired keywords and transmits it to the cloud server. KGC provides the cloud server with its partial private key. It oversees data processing, which includes data computation, data storage, and receiver search.

They proved the scheme's security against both kinds of adversaries in the random oracle model, where type II adversary-two can learn the master key and type I adversary-one (A1) can choose a random public key to replace the user's public key. The suggested scheme's performance was assessed in terms of computational and communication costs. Wu *et al.* [42] later noted that keyword guessing attacks were a problem with the suggested SCF-MCLPEKS scheme. The work by Ma *et al.* [41] has two drawbacks: the trapdoors for a keyword are never renewed, and it is susceptible to keyword guessing attacks.

For the Industrial Internet of Things (IIoT), Zhang *et al.* [43] suggested a verified certificate-less searchable encryption technique. A cloud server and data user, base station, and service provider (SP) serving as KGC in the IIoT paradigm make up the suggested VCLPKES model. Wireless sensors, actuators, and smart devices are used by the Industrial Internet of Things (IIoT) to gather and share data. Numerous wireless sensor nodes and a range of base stations make up a wireless sensor network. The base station forwards data from the Industrial Internet of Things (IIoT) to the cloud server, serving as a bridge between the sensor nodes and the server. Using the base station's identity verification and non-repudiation, the communication between the base station and the cloud server should ensure the confidentiality and integrity of the uploaded data.

Integrity ensures that the data submitted is not altered by the adversary, while confidentiality ensures that the data is not intercepted. The legitimacy of the uploaded data should be ensured if the Base Station's identity is confirmed. To show that their system is insecure, they provided a workable attack after analyzing the security of a certificate-less searchable encryption scheme in the Industrial Internet of Things (IIoT) context of Ma *et al.* [41]. As demonstrated by the attack, Type I adversary A1 can manufacture the trapdoor value for every keyword. They demonstrated that, under the random oracle model assuming intractability of BDH, the VCLPKSE technique meets the ciphertext indistinguishability, trapdoor indistinguishability, and unforgeability for data owner. Lastly, they assessed the scheme's effectiveness in terms of communication and computation expenses.

The secure channel free certificate-less searchable public-key encryption with multiple keywords (SCF-MCLPEKS) scheme for the Internet of Things (IoT) was shown by Wu *et al.* [42] to violate trapdoor indistinguishability by Ma *et al.* [41]. While capturing a trapdoor, the attacker can test how well the keyword and trapdoor match.

The security flaws in the CLPAEKS scheme, which was suggested by Ma *et al.* [41], and the CLPEKS scheme, which was suggested by Zhang *et al.* [43], were resolved by Wu *et al.* [44]. They demonstrate that the Ma *et al.* [41] scheme's reduction proof is flawed. The value of the trapdoor is not subjected to a hashing operation in the trapdoor algorithm developed in [41]. As a result, the attacker (A1) is unable to do a hash query on the challenge keyword's trapdoor. The reduction procedure demonstrates that the challenger algorithm cannot use the attacker (A1) as a subroutine to solve the Bilinear Diffie-Hellman (BDH) problem.

Secondly, they analyzed the security of Zhang *et al.* [43] is not secure against keyword guessing attacks. They proved that the malicious adversary could recover keyword-specific information from any query message taken by the protocol.

3.2.4 Certificate-less Searchable Encryption in the case of Healthcare System

For the mobile healthcare system (MHS), Ma *et al.* [45] suggested certificate-less public key encryption with keyword search (CL-PEKS) in 2018. Numerous factors, including remote data collection, remote monitoring, tracking, diagnostic treatment, and medical professional communication, were considered in the proposed mobile healthcare system (MHS). It allows providing healthcare services anytime anywhere or pervasively (Pervasive Healthcare). They

designed a scheme that alleviates the issue of key escrow and key management. Four entities—a patient, a physician, a KGC, and a cloud server—are involved in the proposed certificate-less public key encryption with a keyword search for the mobile healthcare system (MHS). The user's partial private key, public parameters, and master key are all generated by the KGC. Using the user's public key, the patient creates an encrypted keyword index and affixes it to the encrypted personal health data. It uploads them to the cloud server after that. The doctor creates the trapdoor after obtaining his partial private key from KGC. He or she then forwards it to the server of the cloud mobile healthcare system. The cloud server handles the tasks of searching and storing data.

Under the intractability assumption problem of the Bilinear Diffie-Hellman (BDH), the security of the suggested CL-PEKS for the mobile healthcare system (MHS) is demonstrated in the random oracle model. Lastly, a computational and communication cost analysis of the suggested scheme's performance is shown.

Su *et al.* [46] combined attribute-based keyword search with certificate-less encryption. Three categories of organizations comprise the proposed system model: the data consumer (medical institution), the medical cloud service provider (MCSP), and the data owner. To gather EHRs and upload them to the medical cloud service provider (MCSP), the data owner dons smart medical equipment. Data storage and search services are offered by the MCSP. The medical facility that looks up and downloads particular EHRs from the MCSP for use in medicine is the data user.

The proposed scheme alleviates key escrow challenges and allow search permission control. The attribute-based keyword search is used to achieve search permission control, in which, the data owners can set their EHRs can be searched by medical institutions with specific attributes. To do authorizations, only specified by the data owners can search and access electronic health records (EHRs) in the cloud server. Their scheme supports conjunctive keyword search to enhance search accuracy or to make search results more accurate and adopts a hidden access structure to protect the privacy attributes of the users and EHRs. Furthermore, the plan facilitates dynamic updating of EHRs, giving data owners the flexibility to add and remove EHRs from the cloud server. They developed the index for encrypted EHRs using a Dynamical Adjustable-capacity Cuckoo Filter (DACF), which enables quick EHR location and effective EHR update. Keyword

indexes that facilitate quick lookups and adaptable EHR updating processes are kept in the DACF. Finally, they demonstrated performance evaluation regarding storage and computation cost.

3.3 Summary

In conclusion, we discovered that the certificate-less encryption with keyword search (PEKS) has been designed for different application in different scenarios such as cloud-based mail systems, electronic health systems (EHS), and the Industrial Internet of Things (IIoT). Due to the prevalence of certificate-less-based cryptography, it overcomes the problem of certificate management and key escrow. Whereas the proposed refreshing keyword search by Baek *et al.* [16] has the challenge of certificate management, due to the key generations are based on Public Key Infrastructure (PKI). So, the related works show that there is a gap in limiting trapdoor generation time or refreshing the trapdoor of the search keyword for certificate-less-based encryption to resist the known attacks. Thus, this research goals is to design and develop the searchable encryption that allow to refresh search keyword and alleviate the certificate management issues using pairing-based cryptography.

CHAPTER FOUR: SEARCHABLE ENCRYPTION with a REFRESHING KEYWORD SEARCH

In this Chapter, we describe the proposed system model, the algorithm and computational consistence and workflow of the searchable encryption with a refreshing keyword search based on certificate-less encryption using pairing-based cryptography.

4.1 System Model

The model integrates identity-based cryptography, elliptic curve cryptography, pairing-based cryptography and refreshing keyword search. We used identity-based cryptography because it simplifies the process of key management by using easily recognizable information, such as an email address or phone number for key generations and eliminates the need for certificate management. The elliptic curve cryptography offers strong security with relatively shorter key lengths compared to other public-key algorithms like RSA, and leading to faster operations and lower resource requirements. The pairing-based cryptography utilizes bilinear pairings on elliptic curves and enables a search function on the encrypted data using keyword search. While offering efficient solutions for various cryptographic tasks, understanding the underlying assumptions and implementing secure key management practices are crucial for deploying pairing-based cryptographic systems effectively. The purpose of the renewing keyword search is to provide a search term that is only relevant for a predetermined window of time.

The user identity, point on elliptic curve generator, refreshing keyword, and key generation center (KGC) are the five parts of the suggested SERKS scheme for generic applications. Figure 4.1 displays the system model and its constituent parts of the suggested certificate-less searchable encryption with a novel keyword search technique.

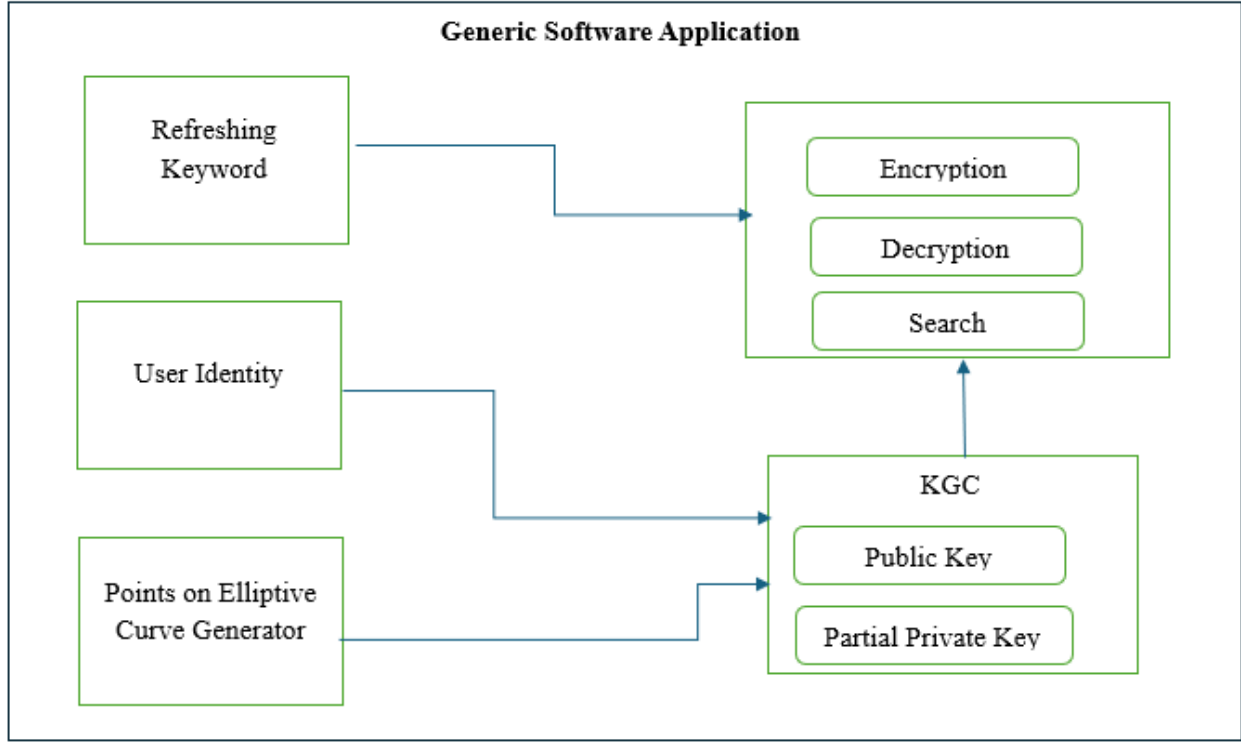


Figure 4.1: *The SERKS System Model*

The basic function and their relationships to the components of the model are described as follows:

a. Refreshing Keyword Search

To accomplish the updating keyword search feature, time information is attached to the trapdoor and is set to expire when the user-specified threshold time is reached. The refreshing keyword search algorithm is shown below.

```

Input: current_Date // Read date on cloud e-mail system server,
       active_Time = null // initialize active time for trapdoor
Output: active_Time
Begin
    If active_Time = null
        active_Time = current_Date
    End If
    If (active_Time - current_Date) > Threshold time //generate trapdoor
        active_Time = current_Date
    End If
End

```

Algorithm 4.1: *Refreshing Keyword Search Algorithm*

b. User Identity

We employed an identity-based primitive, wherein a user's distinct identity information is used to construct their public key. A user's public key in an identity-based encryption method is frequently an arbitrary string, such as an IP address, phone number, or email address. To generate the keys, we utilized an email address as a user identity input. The ability to produce public keys using identities removes the requirement for public key certificates and certificate authorities because identity is a natural link to a user. Thus, our suggested SERKS cryptosystem has the advantage of not requiring a certificate.

c. The Point on Elliptic Curve Generator

The Elliptic Curve Cryptography (ECC) was selected due to it offers a security level equivalent to RSA (Rivest–Shamir–Adleman) and it provides a smaller key size. The ECC generates the random keys through the properties of the elliptic curve equation, of the form $y^2 = x^3 + ax$ over the field F_q (where q is 512 bits), for any $a \in F_q$ (finite field). The generated or selected random key is used as input to perform cryptographic pairing.

d. Key Generating Center (KGC)

The KGC is responsible for generating master keys, public keys and partial private keys for the user. It uses an e-mail address as its identity to generate the key and point on an elliptic curve where identity-based encryption is integrated with elliptic curve cryptography.

e. Encryption and Decryption

The encryption/decryption is performed using the encryption/decryption algorithms, mathematical operations and generated keys. Five fundamental operations form the foundation of the suggested algorithm's construction. These include hash-to-point operations, scalar multiplication, point addition, bilinear pairing, and general hash functions. A mathematical function known as the bilinear pairing operation preserves specific bilinear characteristics while mapping two points on an elliptic curve to an element in a finite field. A point on an elliptic curve is multiplied by a scalar value to produce another point on the same curve, which is known as scalar multiplication. Bilinear pairing calculations and other elliptic curve-based cryptographic primitives heavily rely on the point addition procedure. The general hash functions

operation is used to map input strings to points on the elliptic curve group and accomplished using specialized hash-to-curve functions. The hash-to-point operation refers to the process of mapping an arbitrary input, typically a string or a binary representation, to a point on an elliptic curve. This input needs to be converted into a valid elliptic curve point that can be used in the subsequent pairing computations.

4.2 The SERKS Algorithm

In this section, we describe the proposed algorithm that is implemented for the proposed system model shown in Figure 4.1. The algorithm is used by the Key Generating Center (KGC) component where the KGC component uses User Identity and Point on Elliptic Curve Generator component to generate the keys. Finally, the E-mail System component performs encryption and decryption using generated keys by KGC and uses the Time Stamp component to perform a refreshing keyword search.

The design of the proposed algorithm is based on five basic operations. These operations are bilinear pairing, general hash functions, scalar multiplication, point addition and hash-to-point operations.

With a novel refreshing keyword search, the suggested certificate-less based searchable encryption with refreshing keyword search, the following polynomial-time probabilistic algorithms make up SERKS:

Configuration (k): Assume that G is a group of order q and that q is a huge prime number. Let P represent a generator of G . Select four distinct hash functions in cryptography. $H1, 2$, and 3 : Between $\{0, 1\}^*$ and G and $H4: G_T \rightarrow \{0, 1\}^n \in Zq^*$, where n is a binary string of fixed length, KGC carries out the subsequent actions:

- i. Pick a bilinear pairing by selecting two cyclic groups G_1, G_T with the same order q .

$$e: G_1 \times G_1 \rightarrow G_T$$
- ii. Pick a generator $P \in G_1$ and choose a random number $s \in Zq^*$
- iii. Generate $P_{pub} = sP \in G_1$. Let s be the master key,
- iv. Perform public parameter $prms = (k, G_1, G_T, q, e, P, P_{pub}, H1, H2, H3, H4)$ and publish. Then keep master key s as secret. Moreover, the refresh keyword space $w = \{0, 1\}^* \in G_1$ and refreshed encrypted keyword space is $C_w = \{0, 1\}^n \in G_T$.

To extract the partial private key (D_{ID}), KGC uses the following algorithm:

- i. Takes as input the user's digital identity $ID \in \{0, 1\}^*$.
- ii. Identify the hashed identification ID by computing $Q_{ID} = H1(ID)$.
- iii. Determine the partial private key of the data user, $D_{ID} = s * Q_{ID}$

Determine the secret value (X_{ID}) by having the data sender and data recipient enter their identities ($ID \in \{0, 1\}^*$) and then randomly select their secret value ($X_{ID} \in Zq^*$).

Establish private key (SK_{ID}): The data user establishes its secret key using both partial private (D_{ID}) and secret values (X_{ID}).

Establish the public key (PK_{ID}): The data user establishes the public key as follows:

- i. Enter the user's secret value, X_{ID} , and the public parameter.
- ii. Determine $PK_{ID} = X_{ID}P$. i.e. the public key the product of generator P and secret value (X_{ID}).

Encryption: Assume that $W = \{w_i \mid 1 \leq i \leq n\}$ is a collection of refreshing keywords. The keyword refresh by appending time information to the encrypted document. For example, the keyword $w = w||T$ where T is the time stamp generated using Algorithm (4.1) and represents a refreshing keyword denoted with a time stamp. When owner of the data sends this search keyword to the target server, it executes as follows to encrypt keyword $w_i \in W$ after obtaining public parameter $prms$, data user's public key PK_{ID} and data user's or receiver's identity ID as an input:

- i. Calculate $Q_{ID} = H1(ID)$
- ii. Pick a number $ri \in Zq^*$ randomly.
- iii. Execute $U_i = riP$.
- iv. Calculate $T_i = e(ri(H2(w||T)), PK_{ID}) e(riQ_{ID}, P_{Pub}) e(ri(H3(w||T)), P)$

Where T is the time stamp generated by using Algorithm (4.1)

' e ' represents a bilinear pairing

' ri ' random number $ri \in Zq^*$ selected by the data owner

'H2' represents the hash function used to hash the keywords 'w' with a time stamp

'H3' represents the hash function used to hash the encrypted document with a time stamp

' P_{Pub} ' achieved from the setup algorithm

' P ' represents the generator achieved from the setup algorithm

' PK_{ID} ' represents the public key of the data owner

' Q_{ID} ' represents the hashed identity ID

Perform: $Vi = H4(T_i)$ (2)

The output ciphertext is given by: $C = \{C1, C2 \dots\}$

Where $Ci = (Ui, Vi)$ (3)

' Vi ' represents the hashed (by H4) of both encrypted data and keyword

' Ui ' is used as a security parameter

Finally, the encrypted document Ci is uploaded to the cloud service provider

Generate Trapdoor: the algorithm obtains public parameter $prms$, a refreshing keyword and, the data user's Secret Key SK_{ID} as an input. The trapdoor is achieved by the data user as:

$$Tw = X_{ID}(H2(w||T)) + D_{ID} \quad (4)$$

Where T is the time stamp generated using Algorithm (4.1)

' Tw ' represent the search trapdoor (i.e the refreshed keyword)

' X_{ID} ' represents the secret value chosen by the user

' w ' represents the search keyword

' D_{ID} ' represents the partial private key, which is generated by the KGC

' $H2$ ' represents the hash function, which is used to hash the search keyword with the generated timestamp.

The search is performed by sending the generated trapdoors (Tw) to the cloud service provider.

Performing Test: to perform this task, the target server obtains public parameter $prms$, refreshed keyword w 's, trapdoor Tw 's and refreshed keyword encrypted document Cw as input and

$$\text{Executes: } Vi = H4 (e(Tw + (H3 (w||T)), Ui)) \quad (5)$$

If the equation (5) is correct or if the equivalent holds, keyword w is included in Cw and the output is “1”. Otherwise, outputs “0” which indicates the keyword w is not included in Cw .

4.3 Computational Consistency

Computational consistency refers to the requirement that the various computational operations involved in the proposed algorithm must be consistent with each other and produce the expected results. It ensures the correct and secure functioning of the algorithms.

With the assumption that $w = wi$, where $i \in \{1, 2 \dots n\}$ and T represents the time stamp produced by Algorithm (4.1), we demonstrate that the certificate-less searchable encryption with a refreshing keyword search method satisfies the computational consistency as stated below:

$$\begin{aligned} & H4 (e(Tw + (H3 (wi ||T)), Ui)) \\ &= H4 (e(X_{ID}(H2 (w||T)) + D_{ID} + (H3 (w||T)), riP)). \\ &= H4 (e(X_{ID}(H2 (w||T)), riP) e(D_{ID}, riP) e(H3 (w||T), riP)) \\ &= H4 (e(ri(H2 (w||T)), X_{ID}P) e(sQ_{ID}, riP) e(H3 (w||T), riP)) \\ &= H4 (e(ri (H2 (wi ||T)), PK_{ID}) (riQ_{IDR}, P_{pub}) ((H3 (wi ||T)), P)) \\ &= Vi \end{aligned} \quad (6)$$

The computational consistency proof using equation (6) ensures soundness and can achieve the desired security guarantees and seamlessly integrate into larger application contexts. It shows correct outputs and overall system success.

4.4 The SERKS Flow Chart

The flow chart of the proposed algorithm shows a sequence of steps that exist in the proposed scheme. It starts by performing the setup of the parameters such as curve selection, computing generator, generating masker key and public key. Once the setup is completed, the KGC generates a partial private key for the user (sender and receiver) by using their email address as their identity. The sender or receiver sets their secret value. Then, the data user generates the private key by using its secret key and partial private. By using the keys, the sender encrypts both the message and the search keywords. Finally, it sends both encrypted messages or

documents and search keywords to the cloud server. When the user wants to search on the encrypted document, it creates a trapdoor equivalent to the search keyword by attaching time information in which the search keyword refreshes. Then, the cloud server checks whether the search keyword and the trapdoor are equivalent. If it matches the search keyword, the cloud server outputs 1 and returns the search result to the users. Otherwise, it outputs 0, and the trapdoor does not match the search. Then it ends the search.

The flow chart of the proposed algorithm is shown in Figure 4.2.

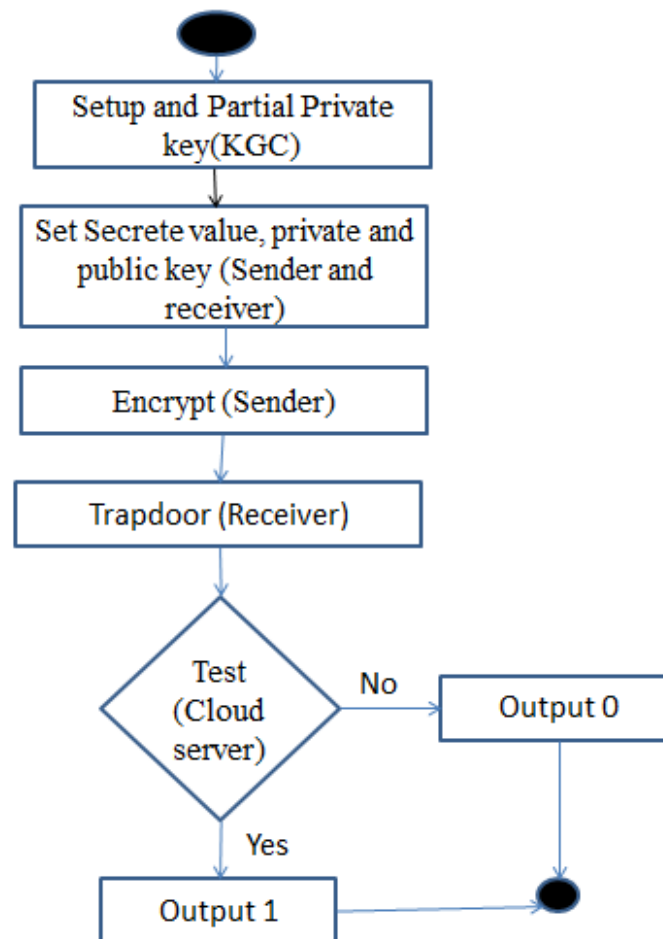


Figure 4.2:Flow Chart of the SERKS Algorithm

4.5 Keyword Guessing Attack Resistance

As the keyword guessing attack is the main attack in searchable encryption the proposed scheme allows to enhance the security of the scheme. Everybody can select a keyword from the keyword space and produce a searchable ciphertext in the keyword-guessing attack. The reason for this is that anyone with the receiver's public key can create the searchable ciphertext using any

keywords. By determining whether the searchable ciphertext he or she has constructed matches the trapdoor, the attacker constantly tries to get the keyword from it. The attack is successful and the right keyword is found if they match; if not, the attacker keeps guessing. However, in the proposed scheme the search keyword is generated by attaching time information in which the keyword refreshes.

4.6 Summary

In this chapter, we presented the proposed system model, the algorithm and computational consistence and workflow of the searchable encryption with a refreshing keyword search based on certificate-less encryption using pairing-based cryptography. Since it is based on certificate-less public encryption it alleviates the need of cryptographic certificate, certificate management and certificate authority. It refreshes the keyword by attaching time to the encrypted keyword and the encrypted document.

The proposed SERKS scheme supports secure multi-party computation, where multiple participants jointly compute a result without revealing their inputs. It enhances the security of the searchable encryption and minimizes the keyword guessing attack. The security hardness of the proposed scheme is based on the intractability of the Bilinear Diffie-Hellman (BDH) problem assumption. Due to, the proposed certificate-less refreshing keyword search scheme is also based on the identity-based encryption and allow partial public key by user, it excludes the key escrow problem.

CHAPTER FIVE: PROTOTYPE and PERFORMANCE EVALUATION

In this Chapter, we discuss the prototype development and the performance evaluation of the designed and developed Searchable Encryption with Refreshing Keyword Search (SERKS) scheme using pairing-based cryptography.

5.1 Prototype

We developed the prototype of the designed Searchable Encryption with Refreshing Keyword Search (SERKS) scheme in the case of a web-based e-mail system and its source code provided in Annex B. The web-based e-mail system was selected due to most people commonly used to transmit messages electronically.

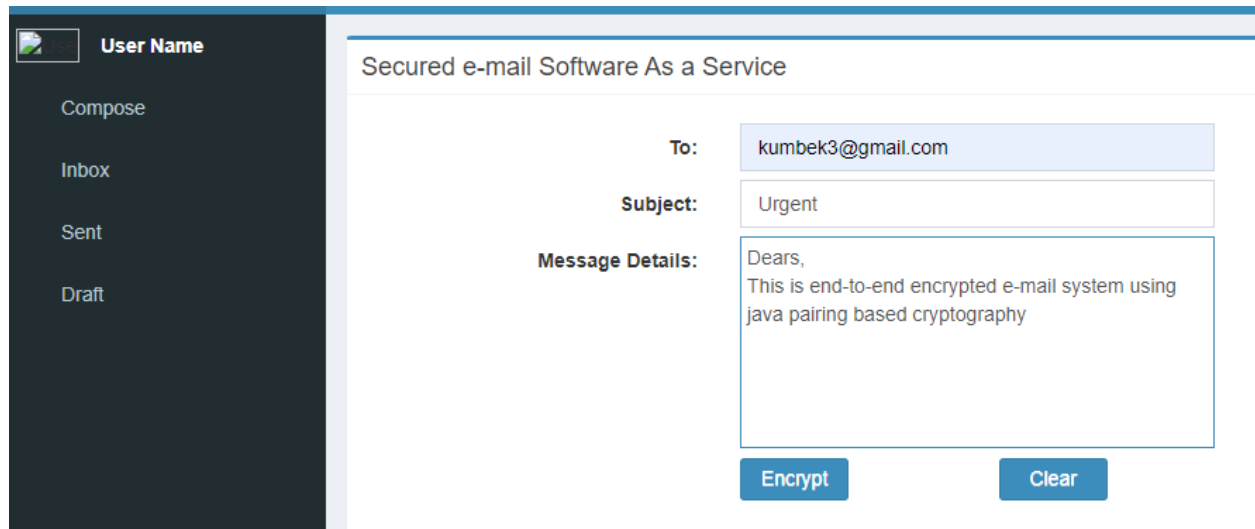
The prototype software is developed under an object-oriented environment, using Java as a programming language and the NetBeans 8.0.2 IDE tool is used to program the system. The Java EE7, which stands for Java Enterprise Edition, which is a Java platform, is used. The Java EE7 platform offers an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications. The user interface of the prototype is developed by using the Java Server Face (JSF) framework, which is part of Java EE.

We used the Java Pairing-Based Cryptography (JPBC) [47] library, to make the mathematical operations underlying pairing-based cryptosystems directly in Java. It provides a port for the pairing-based cryptography (PBC) library. The Pairing-Based Cryptograph (PBC) library is a free library based on the c programming language that performs the mathematical operations underlying pairing-based cryptosystems. It is designed to be the backbone for the development of pairing-based cryptosystems. It also offers different application such as elliptic curve generation, elliptic curve arithmetic and pairing computation.

According to [47], we used the **Type A** elliptic curve, which is super-singular of the form $y^2 = x^3 + ax$ over the field F_q (where q is 512-bits), for any $a \in F_q$ (finite field). Based on this point, pairing functions are computed. The **Type A** elliptic curve points used are presented in Annex A.

Next, using computed pairing the keys are generated and used to perform data encryption.

Figure 5.1 shows the user interface of the prototype in the case of a web-based e-mail system.



The image shows a web-based user interface for an email system. On the left is a dark sidebar with a 'User Name' header and a profile icon. Below it are links for 'Compose', 'Inbox', 'Sent', and 'Draft'. The main area is titled 'Secured e-mail Software As a Service'. It contains a form with three fields: 'To:' with the value 'kumbek3@gmail.com', 'Subject:' with the value 'Urgent', and 'Message Details:' with the text 'Dears, This is end-to-end encrypted e-mail system using java pairing based cryptography'. At the bottom of the form are two buttons: 'Encrypt' and 'Clear'.

Figure 5.1: *User interface of the prototype for the e-mail system*

When the e-mail sender clicks on the ‘Encrypt’ button, it produces the ciphertext as shown in Figure 5.2. We used the subject of the e-mail as a search keyword and the e-mail address as the identity of the users. The sender performs encryption using the public key and identity of the receiver. The public key is generated from the e-mail address which is their unique identity information. Due to the e-mail address which serves as the identity and is a commonly used by the users, the ability to use a unique e-mail identity to generate a public key eliminates the need for public key certificates and certificate authorities. It reduces the cost of certificate management by binding the participant’s identity and public key. The refreshing keyword is generated by attaching the current to the keyword and document (message details). Next, the created ciphertext is sent to the cloud database server.

Encrypted e-mail System using Java Pairing Based Cryptograph

To:	3026167381795745812405256461952594615771
Subject:	8347062863905394750427404387411953755 8257386839928112096169752842450848885
Message Details:	{x=21338023746678953858917495247125911029 89609457047400739765701588901442493101548 17750046520673073859324316579046420490989

Figure 5.2: Ciphertext of the prototype

Finally, the e-mail receiver uses his partial private key and a secret value to generate its trapdoor to perform search functions and decrypt the message sent.

5.2 Performance Evaluation

To evaluate the performance, the study evaluates the time complexity of the proposed SERKS. The efficiency of the proposed algorithm will be evaluated by conducting computational cost analysis, which is the execution time of the proposed algorithm and communication cost analysis, which is the size of public keys, ciphertext and trapdoor. Finally, its security properties are provided and compared with existing works.

We compared the performance of our proposed certificate-less searchable encryption with a refreshing keyword search system using previous studies like Yanguo *et al.* [15] and Ma *et al.* [41].

5.2.1 Computational Cost (Execution Time)

The execution time of the proposed scheme contains five basic components that are considered for overall performance of the proposed scheme. These are bilinear pairing (T_{bp}), scalar multiplication (T_{sm}), point addition (T_{pa}), general hash functions (T_h) and hash-to-point (T_H) execution time.

The bilinear pairing execution time (T_{bp}) measures bilinear pairing computation and scalar multiplication execution time (T_{sm}) measures scalar multiplication operations. The execution time of point addition (T_{pa}) measures the point addition operations are performed on the underlying elliptic curve and the execution time of general hash functions (T_h) measures the hashing of input data and the generation of random values. The hash-to-point (T_H) measures the hash-to-point operation, which refers to the process of mapping an arbitrary input, typically a string or a binary representation, to a point on an elliptic curve. This input needs to be converted into a valid elliptic curve point that can be used in the subsequent pairing computations.

The proposed SERKS computational cost evaluation is based on the notation and execution time provided by Ma *et al.* [41] using the MIRACL library [48]. We used as an evaluation benchmark for the executing times for notations used in the evaluation are defined in Table 5.1.

Table 5.1: Notations and execution times (ms)

Notations	Description	Times(ms)
T_{sm}	execution time for scalar multiplication	2.165
T_{bp}	execution time for a bilinear pairing	5.427
T_H	execution time for a hash-to-point	5.493
T_h	execution time for a general hash function	0.007
T_{pa}	execution time for a point addition	0.013

The computational costs for the Yanguo *et al.* [15], Ma *et al.* [41], and our suggested schemes are displayed in Table 5.2.

Table 5.2: Comparison of computational cost (execution times (ms))

Comparison criteria	Yanguo <i>et al.</i> [15]	Ma <i>et al.</i> [41]	Our proposed scheme
Key Generation	$2T_H + 8T_{sm}$ =28.306	$2T_H + 4T_{sm}$ =19.646	$T_H + 3T_{sm} = 11.988$
Encryption	$3T_H + 2T_h + 5T_{sm} + 3T_{bp} = 43.599$	$3T_H + T_h + 4T_{sm} + 3T_{bp} + T_{pa} = 41.433$	$2T_H + T_h + 4T_{sm} + 3T_{pb} = 35.927$
Trapdoor Generation	$T_H + T_h + 3T_{sm} = 11.995$	$T_H + T_{sm} + T_{pa} = 7.671$	$T_H + T_{sm} + T_{pa} = 7.7$
Test Execution	$T_h + T_{sm} + 2T_{pa} + T_{bp} = 7.625$	$2T_H + T_h + T_{sm} + 2T_{pa} + T_{bp} = 18.611$	$T_H + T_h + T_{sm} + T_{pa} + T_{pb} = 13.105$

Although our suggested SERKS technique offers refreshing keyword search, its computational cost is lower than that of the Yanguo *et al.* [15] and Ma *et al.* [41] schemes, as indicated in Table 5.2. However, the test phase execution time is longer than that of the Yanguo *et al.* [15] scheme.

5.2.2 Communication Cost

The communication cost defines the size of the ciphertext, public keys and trapdoor of the proposed scheme.

We let $|G1|$ represents the bit-size of a point in Group G

$|\mathbb{Z}_q|$ represents the bit size of a number in $\mathbb{Z}_q$

$|PK|$ represents the bit size of PK

$|C|$ represents the bit size of the ciphertext

$|T|$ represents the bit size of the trapdoor respectively. Table 5.3 shows the comparison of communication cost schemes in Yanguo *et al.* [15], Ma *et al.* [41] and the proposed scheme.

Table 5.3: *Comparison of communication cost*

Comparison criteria	Yanguo <i>et al.</i> [15]	Ma <i>et al.</i> [41]	Our proposed scheme
PK Size	$4 G1 $	$2 G1 $	$ G1 $
CT Size	$ G1 + \mathbb{Z}q $	$ G1 + \mathbb{Z}q $	$ G1 + \mathbb{Z}q $
TD Size	$3 G1 $	$ G1 $	$ G1 $

Additionally, it is demonstrated that the suggested system's overall communication cost is lower than that of both methods in Yanguo *et al.* [15] and Ma *et al.* [41].

5.2.3 Security Properties

We provided the security properties of the proposed SERKS scheme by considering comparison criteria such as certificate management problem, key escrow problem and refreshing keyword functionalities. Table 5.4 shows the comparison of the security properties of the proposed scheme.

Table 5.4: *Comparison of security properties*

Comparison criteria	Yanguo <i>et al.</i> [15] scheme	Ma <i>et al.</i> [41] scheme	Our proposed scheme
Certificate management issue	Yes	Yes	Yes
Key escrow issue	Yes	Yes	Yes
Refreshing keyword	No	No	Yes

Yes: represents that the system meets the respective problem.

No: represents that the system does not meet the security requirements

In this Chapter, we developed the prototype and provided the performance evaluation of the proposed SERKS scheme in terms of computational cost, communication cost and security properties. The evaluation result shows that the SERKS scheme performs better than the existing research works.

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we proposed searchable encryption with a refreshing keyword search scheme based on certificate-less public key encryption using pairing-based cryptography. The proposed SERKS scheme alleviate the issue of the target server storing trapdoor for keywords in the system when the target user performs a keyword search by his trapdoor on the target server that occurs in existing searchable encryption. This is achieved by accomplished by appending date information to the keyword and encrypted data. It primarily enhances the security of the data by reducing the chance for the target server to guess the full keyword information that allows to access the ciphertext.

The security hardness of the proposed SERKS scheme is based on the intractability of the Bilinear Diffie-Hellman (BDH) problem assumption. To show the practicability of the proposed SERKS scheme, we developed a prototype in the case of a web-based e-mail system. The implemented prototype allows end-to-end encryption between the communicating parties and only legitimate users can decrypt the messages or documents. It also alleviates the issue of public key replay attacks, due to the digital signature being performed before encryption and encryption.

Additionally, the performance of the suggested SERKS approach is given, and the outcomes are contrasted with some of the current schemes in terms of communication and computing costs.

6.2 Contribution

This work has several contributions to searchable encryption with keyword search. We have contributed:

- Developed an algorithm that refreshes keywords for searchable encryption to enhance the security of searchable encryption and minimize the risk of keyword guessing attacks.
- Designed certificate-less searchable encryption with a refreshing keyword search model based on the intractability of BDH (Bilinear Diffie-Hellman) problem assumption.
- Designed scheme support to secure multi-party computation, where multiple participants jointly compute a result without revealing their inputs.

- Developed a prototype of the proposed SERKS scheme in the case of a web-based e-mail system which allows to achieve end-to-end encryption and searchable.

6.3 Future Work

To further improve SERKS and to make it more efficient, we have identified the following directions:

- To design searchable encryption with refreshing keyword search by considering advanced search functions such as conjunctive and disjunctive search.
- To integrate or deploy pairing-based schemes in real-world applications.
- To design searchable encryption with keywords by using lattice-based pairing.

REFERENCES

- [1] S.Kamara and K.Lauter, "Cryptographic Cloud Storage," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, Tenerife, Canary Islands, Spain, 2010.
- [2] C.Bosch, P.Hartel, W.Jonker and A.Peter, "A survey of Provably Secure Searchable Encryption," *ACM Computing Surveys (CSUR)*, 2014.
- [3] DX.Song, D.Wagner and A.Perrig, "Practical Techniques for Searches on Encrypted Data," in *Proceedings on Security and Privacy*, 2000.
- [4] S.Kamara, C.Papamanthou and T.Roeder, "Dynamic Searchable Symmetric Encryption," in *Proceedings of the 2012 ACM conference on Compute rand communications security*, 2012.
- [5] G.Wang, C.Liu, Y.Dong, P.Han, H.Pan and B.Fang, "IDCrypt: A Multi-User Searchable Symmetric Encryption Scheme for Cloud Applications," in *IEEE International Conference on Communications (ICC)*, 2017.
- [6] D.Boneh and G.Di.Crescenzo, "Public Key Encryption with Keyword Search," in *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 2004.
- [7] Y.Zhou, Y. N.Li, D.An and L.Wang, "Public Key Encryption with Keyword Search in Cloud," *Entropy*, 2020.
- [8] SS.Al-Riyami and KG.Paterson, "Certificateless Public Key Cryptography," in *Proceedings of the 2003 International Conference on the Theory and Application of Cryptology and Information Security*, Berlin/Heidelber Germany, 2003.
- [9] L.Malina, J.Hajny and V.Zeman, "Usability of Pairing-Based Cryptography on Smart Phone," in *International Conference on Telecommunications and Signal Processing (TSP)*, 2015.
- [10] Ben.Lynn, "On the Implementation of Pairing-Based Cryptography, PhD Thesis,

- Department of Computer Science," STANFORD UNIVERSITY, 2007.
- [11] STANFORD-UNIVERSITY, "PBC (pairing-based cryptography) library," STANFORD-UNIVERSITY, Last accessed on April 20, 2024.
 - [12] V.S. Miller, "Use of Elliptic Curves in Cryptography, Advances in Cryptology," in *Conference on the Theory and Application of Cryptographic Techniques*, 1987.
 - [13] D.Moody, R.Peralta, R.Perner, A.Regenscheid, A.Roginsky and L. Chen, "Report on Pairing-based Cryptography," *Journal of Research of the National Institute of Standards and Technology*, 2015.
 - [14] Q.Huang and H.Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Information Sciences*, 2017.
 - [15] P.Yanguo, C.Jiangtao, P.Changgen and Y.Zuobin, "Certificateless Public Key Encryption with Keyword Search," in *IEEE International Conference on Communications (ICC)*, 2014.
 - [16] J.Baek, R.Safavi-Naini and W.Susilo, "Public key encryption with keyword search revisited," in *International Conference on Computational Science and Its Applications*, Perugia, Italy, 2008.
 - [17] J.Li, X.Lin and Y.Zhang, "KSF-OABE: Outsourced Attribute-based Encryption with Keyword Search function for Cloud Storage," in *IEEE Transactions on Services Computing*, 2017.
 - [18] TY.Wu, C.Meng, KH.Wang, CM.Chen and JS.Pan, "Comments on certificateless designated server based public key encryption with keyword search scheme," in *International Conference on Genetic and Evolutionary Computing*, Springer, Singapore, 2017.
 - [19] C. Kaya and Koc, *About cryptographic engineering*, Springer, Boston, 2009.
 - [20] A.Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the 1984 Workshop on the Theory and Application of Cryptographic Techniques*, Paris, France, 1984.

- [21] A.Sahai and B.Waters, "Fuzzy Identity-based Encryption," in *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg,, 2005.
- [22] F.Han, J.Qin, H.Zhao and J.Hu, "A general transformation from KP-ABE to searchable encryption," in *Future Generation Computer System*, 2014.
- [23] C.Wang, W.Li, Y.Li and X.Xu, "A Ciphertext-policy Attribute-based Encryption Scheme Supporting Keyword," in *In Cyberspace Safety and Security*, Berlin/Heidelberg, Germany, 2013.
- [24] SS.Dhanda, B.Singh and P.Jindal, "Demystifying elliptic curve cryptography: Curve selection, implementation and countermeasures to attacks," *Journal of Interdisciplinary Mathematics*, 2020.
- [25] K.Keerthi and B.Surendiran, "Elliptic Curve Cryptography for Secured Text," in *International Conference on Circuit, Power and Computing Technologies*, 2017.
- [26] RL.Rivest, A.Shamir and L.Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*.
- [27] D.Mahto and DK.Yadav, "Performance Analysis of RSA and Elliptic Curve," *International Journal of Network Security*,, vol. 20, no. 4, 2018.
- [28] K.Sowjanya, M.Dasgupta, S.Ray and M.S.Obaidat, "An Efficient Elliptic Curve Cryptography-Based Without Pairing KPABE for Internet of Things," *IEEE Systems Journal*, 2019.
- [29] D.Boneh and M.Franklin, "Identity-Based Encryption from the Weil Pairing," in *Annual International Cryptology Conference*, Springer, 2001.
- [30] A.Pankova and P.Laud, "Symbolic Analysis of Cryptographic Protocols Containing Bilinear Pairings," in *IEEE 25th Computer Security Foundations Symposium*, 2012.
- [31] B.Lynn, "Java Pairing-Based Cryptography Library (JPBC)," Available on, <http://gas.dia.unisa.it/projects/jpbc/index.html>, Last accessed on April 20, 2024.

- [32] meri-stuff, "cryptography-theory-1-meaning-of-secure," *Reterived from <https://meri-stuff.blogspot.com/2013/11/cryptography-theory-1-meaning-of-secure.html>*, 2013.
- [33] K. Sako, "Semantic Security," 2011.
- [34] L.Yang, W.Gang and L.Jiguo, "Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement," *Information Sciences*, 2018.
- [35] T.Xiang, Z.Wang, B.Chen, X.Li, P.Wang and F.Chen, "StopGuess: A framework for public-key authenticated encryption with keyword search," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2024.
- [36] J.Sayid, I.Sayid and J.Kar, "Certificateless Public Key Cryptography: A Research Survey," *International Journal of Security and Its Applications*, vol. 10, no. 7, 2016.
- [37] J.L.Carter and M.N.Wegman, "Universal Classes of Hash Functions," *Journal of Computer and System Sciences*, 1979.
- [38] M.Bellare and P.Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proc. First Annual Conf. Computer and Communications*, 1993.
- [39] Q.Zheng, X.Li and A.Azgin, "CLKS: Certificateless Keyword Search on Encrypted Data," in *International Conference on Network and System Security*, 2015.
- [40] T.Y.Wu, F.Meng, C.M.Chen, S.Liu and J.S.Pan, "On the security of a certificateless searchable public key," in *International Conference on Genetic and Evolutionary Computing*, Berlin/Heidelberg, Germany, 2016.
- [41] M.Ma, D.He, N.Kumar, K.R.Choo and J.Chen, "Certificateless Searchable Public Key Encryption Scheme for Industrial Internet of Things," in *IEEE Transactions on Industrial Informatics*, 2017.
- [42] T.Y.Wu, C.M.Chen, K.H.Wang and J.M.T.Wu, "Security analysis and enhancement of a certificateless searchable public key encryption scheme for Industrial Internet of Things environments," *IEEE Access, ieeexplore.ieee.org*, vol. 7, 2019.

- [43] Y.Zhang, X.Liu, X.Lang and C.Wang, "Verifiable Certificateless Public Key Searchable Encryption Scheme for Industrial Internet of Things," in *IEEE International Conference on Communications*, 2020.
- [44] B.Wu, C.Wang and H.Yao, "Security analysis and secure channel-free certificateless searchable public key authenticated encryption for a cloud-based Internet of Things," *PloS one, journals.plos.org*, vol. 15, no. 4, 2020.
- [45] M.Ma, H.Debiao, M-K.Khan and J.Chen, "Certificateless Searchable Public Key Encryption Scheme for Mobile Healthcare System," in *Computers & Electrical Engineering*, 2018.
- [46] Y.Su, Y.Li, Q.Cao and Z.Wu, "Authorized Certificateless Conjunctive Keyword Search on Encrypted EHRs from WSNs," *Journal of Information Science & Engineering*, 2020.
- [47] A. Caro and V.Iovino, "JPBC: java pairing based cryptography," in *IEEE Symposium on computers and communications*, 2011.
- [48] M.Scott, "MIRACL-A multi-precision integer and rational arithmetic C/C++ library," Retireved from, [http://www. shamus. ie.](http://www.shamus.ie), 2003.
- [49] Q.Huang and H.Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," in *Information Sciences*, 2017.

Annexes

Annexe A: Type A selected points on elliptic curve properties

```
type a

q
878071079966331252243778198475404981580688319941420821102865339926647563088022295707862517942
2662221423155858769582317459277713367317481324925129998224791

h
120160122648911460793888213667405342048029544012513118229196151310472072893597045311028448021
83906537786776

r 730750818665451621361119245571504901405976559617

exp2 159

exp1 107

sign1 1
```

Annexe B: *Source Code*

```
package controller;

import it.unisa.dia.gas.jpbc.Element;

import it.unisa.dia.gas.jpbc.Field;

import it.unisa.dia.gas.jpbc.Pairing;

import it.unisa.dia.gas.plaf.jpbc.pairing.PairingFactory;

import java.io.Serializable;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.faces.view.ViewScoped;

import javax.inject.Inject;

import javax.inject.Named;

import model.Email;

/**
```

```

* @author kuma bekele
*/

@Named("")
@ViewScoped

public class CSERKS implements Serializable {

    @Inject

    Email email;

    Pairing pairing;

    Element masterKeyS;

    Element generatorP;

    Element pPub;

    Element partialPrivateKey_DiD;

    Element secreteVal_eXiD;

    Element privateKey_SK;

    Element publicKey_PK;

    Date date = new Date();

    Element documentWithDate_H3T;

    Element hashedIdentity_QiD;

    Element refreshKeywords_H2T;

    String secreteVal_XiD = "4545";

    Element _Ti; public void generatedParameter() { // to generate Parameters for setup

        pairing = PairingFactory.getPairing("./params/curves/a.properties");

        PairingFactory.getInstance().setUsePBCWhenPossible(true);

        Element g1, g2;

```

```

g1 = pairing.getG1().newRandomElement();

g2 = pairing.getG2().newRandomElement();

generatorP = pairing.getG1().newRandomElement(); //Compute Generator

Field G1 = pairing.getG1();

masterKeyS = G1.newRandomElement(); //Compute Master

pPub = masterKeyS.mul(generatorP); // Compute Public key

System.out.println("pPub=" + pPub);

}

public void generatePartialPrivateKey() { //Generate Partial private key

    byte[] identityAsBytes = email.getSender().getBytes();

    hashedIdentity_QiD = pairing.getG1().newElement().setFromHash(identityAsBytes, 0,
identityAsBytes.length);

    partialPrivateKey_DiD = masterKeyS.mul(hashedIdentity_QiD);

    System.out.println("partialPrivateKey_DiD=" + partialPrivateKey_DiD);

}

public void generateSecreteVal() { //Generate secret value

    byte[] secreteVal_XiDAsByte = secreteVal_XiD.getBytes();

    secreteVal_eXiD = pairing.getG1().newElement().setFromHash(secreteVal_XiDAsByte, 0,
secreteVal_XiDAsByte.length);

}

public void generatedSK() { //Generate Private Key

    privateKey_SK = partialPrivateKey_DiD.mul(secreteVal_eXiD);

}

public void generatePK() { //Generate Public Key

    publicKey_PK = generatorP.mul(secreteVal_eXiD);

```

```

        System.out.println("publicKey_PK=" + publicKey_PK);
    }

    public void refreshKeywords() { //Generate regfreshing Keyword seach

        SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");

        System.out.println(formatter.format(date));

        System.out.println("Curreent date " + formatter.format(date));

        String dateToString = formatter.format(date);

        byte[] dateToStringAsByte = dateToString.getBytes();

        byte[] keywordsAsByte = email.subject.getBytes();

        Element hashedDate = pairing.getG1().newElement().setFromHash(dateToStringAsByte, 0,
dateToStringAsByte.length);

        Element hashedkeywords = pairing.getG1().newElement().setFromHash(keywordsAsByte, 0,
keywordsAsByte.length);

        System.out.println("hashedkeywords " + hashedkeywords);

        refreshKeywords_H2T = hashedkeywords.mul(hashedDate);

        System.out.println("refreshKeywords_H2T " + refreshKeywords_H2T);

        //refreshKeywords_H2T.isEqual(refreshKeywords_H2T); // compare
    }

    public void document() { //Generate encrypt the message or document

        byte[] documentsAsByte = email.message.getBytes();

        Element hashedDocument = pairing.getG1().newElement().setFromHash(documentsAsByte, 0,
documentsAsByte.length);

        SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");

        System.out.println("Curreent date " + formatter.format(date));

        String dateToString = formatter.format(date);

```

```

byte[] dateToStringAsByte = dateToString.getBytes();

Element hashedDate = pairing.getG1().newElement().setFromHash(dateToStringAsByte, 0,
dateToStringAsByte.length);

documentWithDate_H3T = hashedDocument.mul(hashedDate);
System.out.println("documentWithDate_H3T= " + documentWithDate_H3T);

}

public void encrypt() { // attaches the doc't to keyword search

    Element _Ri = pairing.getG1().newRandomElement();

    _Ti = pairing.pairing(_Ri.mul(documentWithDate_H3T), publicKey_PK)

        .mul(pairing.pairing(_Ri.mul(hashedIdentity_QiD), pPub)).

        mul(pairing.pairing(_Ri.mul(documentWithDate_H3T), generatorP));

    System.out.println("_Ti " + _Ti);

}

public void generateTrapdoor() {

    Element _Tw = secreteVal_eXiD.mul(refreshKewords_H2T).add(partialPrivateKey_DiD);

    System.out.println("_Tw " + _Tw);

}

public void encryptedEmail() {

    email.setSender(hashedIdentity_QiD.toString());

    System.out.println("encrypted/refreshed Identity:" + email.getSender());

    email.setSubject(refreshKewords_H2T.toString());

    System.out.println("encrypted/refreshed keyword Subject:" + email.getSubject());

    email.setMessage(_Ti.toString());

    System.out.println("encrypted/refreshed keyword _Ti:" + email.getMessage());

}

```

```
public void tests() {  
    generatedParameter();  
    generatePartialPrivateKey();  
    generateSecreteVal();  
    generatedSK();  
    generatePK();  
    refreshKewords();  
    document();  
    encrypt();  
    generateTrapdoor();  
    encryptedEmail();  
}
```

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university and that all sources of materials used for the thesis have been duly acknowledged. Declared by:

Name: Kuma Bekele

Signature: _____

Date: _____

Confirmed by advisor:

Name: Minale Ashagrie(PhD)

Signature: _____

Date: _____