DESIGNATED SENDERS PUBLIC-KEY SEARCHABLE ENCRYPTION FOR INTERNET OF THINGS IN HEALTHCARE

SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

SHAKIL MAHMUD

STUDENT ID: 144420

MUHTASIM MUSFIQ ZARAB

STUDENT ID: 144419

UNDER THE SUPERVISION OF

NUSRAT ZERIN ZENIA

Lecturer, Department of CSE (IUT)

بسم الله الرحمن الرحيم

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

Designated Senders Public Key Searchable Encryption for Internet of Things in Healthcare

Author

Shakil Mahmud (Student ID-144420)
Muhtasim Musfiq Zarab (Student ID-144419)

Supervised by
Nusrat Zerin Zenia
Lecturer
Department of Computer Science and Engineering,
Islamic University of Technology.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

GAZIPUR, BANGLADESH

October, 2018

Declaration

We, Shakil Mahmud and Muhtasim Musfiq Zarab, declare that this thesis titled "**Designated Senders Public Key Searchable Encryption for Internet of Things in Healthcare**" and the works presented in it are our own. We confirm that:

- This work has been done for the partial fulfillment of the Bachelor of Science in Computer Science and Engineering degree at this university.
- Any part of this thesis has not been submitted anywhere else for obtaining any degree.
- Where we have consulted the published work of others, we have always clearly attributed the sources.

Submitted by:

Shakil Mahmud (144420)

Muhtasim Musfiq Zarab (144419)

Designated sender public-key searchable encryption for Internet of Things in healthcare

Approved by:

Nusrat Zerin Zenia

Lecturer,

Department of Computer Science and Engineering,

Islamic university of Technology

Date: …….

Prof. Dr. Muhammad Mahbub Alam

Head of the Department,

Department of Computer Science and Engineering,

Islamic university of Technology

Date: ……

Acknowledgement

First and foremost, we offer gratitude to the Almighty Allah (SWT) for giving us the capability to do this work with good health.

We are grateful to our thesis supervisor, Nusrat Zerin Zenia, for the support and guidance throughout our research at Islamic University of Technology (IUT). She created a nice research environment for which we have able to explore many ideas without constraint. We have gained a wealth of knowledge and future endeavor. For all of her efforts as our true mentor, we express our heartfelt gratitude to her.

We would like to thank all the faculty members of the department of CSE, IUT for their inspiration and help.

And last but not least we are thankful to our family, friends and well-wishers for their support and inspiration. Without them it would never been possible for us to make it this far.

ABSTRACT

Internet of things (IoT) applications comprising thousands or millions of intelligent devices or things is fast becoming a norm in our inter-connected world, and the significant amount of data generated from IoT applications is often stored in the cloud. Today Cloud storage has become popular technology for data storage. However, end users will not completely trust the cloud. So, they encrypt and store the data. So, essential information should be encrypted before outsourcing for privacy concerns, but this makes difficult for the users of the cloud to utilize it. However, searching encrypted data in the cloud remains an ongoing challenge. Existing Searchable Encryption protocols include searchable symmetric encryption (SSE) and public-key encryption with keyword search (PEKS). In the area of searchable encryption, the Public key Encryption with keyword search (PEKS) is an attractive technique in secure cloud storage. PEKS assures the data confidentiality without affecting the usage of the data stored in the cloud. Furthermore, compared with the symmetric searchable encryption, PEKS does not require key distribution and management. Using the public key, anyone including the server can compute the ciphertext of any keyword. Thus, one drawback for this system is Keyword Guessing Attack (KGA).

To address this security, we introduce a new type of PEKS that is secure against KGA, designated-senders PEKS. In this type of PEKS, the receiver can designate a group of senders who can encrypt keywords. Thus, since the malicious server cannot encrypt any keyword, the server cannot launch KGA. Furthermore, we construct a designated-senders PEKS scheme using a broadcast encryption. we propose an efficient and secure searchable encryption protocol using the trapdoor permutation function (TPF) rather than using bilinear pairing operation to reduce complexity of key generation. Both the theoretical analysis and the experimental results show that our scheme achieves strong security along with high efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF NOTATIONS

Λ	A security parameter
G1	An additive group of prime order q
P	A generator of G1
negl(λ)	A negligible probability
Ω	The system parameters
PK_C	Public key Key-Center
SKc	Private key key-Center
W	The keyword set
TPF	A trapdoor permutation function
I_w	A set of indexes of files containing the keyword w
C_{Iw}	The ciphertext of index set I_w
U_{Tw}	The storage location of C_{Iw}
T_w	The trapdoor of keyword w
ED	The database of ciphertext
|S|	The size of set S
||	Concatenation operator
$\oplus$	The bit-wise XOR operation

Introduction

1.1 Overview

The Internet of Things (IoT) has the potential to transform the healthcare sector. From pacemakers to blood pressure cuffs, IoT healthcare can help doctors better manage diseases, monitor patients, and improve treatment outcomes – but healthcare data security is a substantial risk that must be addressed.

The patient data collected and stored via connected healthcare devices is extremely sensitive and valuable to hackers who can use the stolen information for blackmail or medical identity theft.

The overall costs for the prevention or management of chronic illnesses can be reduced using number of technologies. These include devices constantly monitor health indicators, devices that auto-administer therapies, or devices that track Realtime health data when a patient self-administers a therapy. Because they have increased access to high-speed Internet and smartphones, many patients have started to use mobile applications (apps) to manage various health needs. These devices and mobile apps are now increasingly used and integrated with telemedicine and telehealth via the medical Internet of Things (mIoT). The Internet of Things (IoT) is a network of physical devices and other items, embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. Its impact on medicine will be perhaps the most important, and personal, effect. By 2020, 40% of IoT-related technology will be health-related, more than any other category, making up a $117 [2] billion market. The convergence of medicine and information technologies, such as medical informatics, will transform healthcare as we know it, curbing costs, reducing inefficiencies, and saving lives.

Cloud computing is set to dominate healthcare, with a new report predicting the global market for healthcare cloud computing will reach $US35 billion by 2022.

Increased patient and practitioner expectations for healthcare technology to offer anytime/anywhere access is driving the adoption of cloud technology across the healthcare space, according to a new study by BCC Research.

The report, Healthcare Cloud Computing: Global Markets to 2022, predicts the global market for healthcare cloud computing to grow at a compound annual growth rate of 11.6% from 2017 through 2022, to reach $35.0 billion by 2022 [3]. This exponential growth is linked to cloud-based solutions allowing faster access from anywhere, and any device, while offering better diagnostic quality and instant online access to patient images and reports.

Initially approached with caution by healthcare organizations, cloud computing is now becoming more widely adopted. In 2017, cloud adoption in healthcare increased globally, as cost savings outweighed potential data protection concerns.

However, patient privacy and confidentiality remain significant concerns, with the report stressing new strategies must be continue to be developed, refined and updated to limit security breaches. Even so, more than 80% of IT organizations in healthcare are using cloud computing globally [4].

A cloud-based healthcare ecosystem offers more effective integration and storage of volumes of patient data and information, while helping to orchestrate fragmented information across multiple systems and sites of care. In fact, according to a Kaufman Hall report, healthcare consolidation hit an all-time high in 2017, with 115 deals totaling $63.2 billion [5].

Transitioning to a cloud-computing environment will be a daunting process, as many in healthcare systems still rely on handwritten notes and paper charts," BCC Research senior IT editor, Michael Sullivan, said. "Investing in cloud for healthcare providers represents a cultural change that will require time to realize, but even still, BCC Research expects a greater portion of total healthcare IT spending to migrate to cloud technologies. There is increasing pressure on healthcare providers to manage big data, secure it better with the right data governance structures and leverage it to gain deeper patient insights at scale. And it's no surprise why more healthcare providers are turning to the cloud ecosystem as a solution. Electronic patient records continue to increase in size. And a surge in patient data volumes is putting healthcare organizations under pressure to find more effective solutions to managing and storing sensitive patient data.

In the US, the 2017 TechTarget purchasing intentions survey, conducted in conjunction with the College of Health Information Management Executives, showed that increased healthcare data storage needs were driving health IT technology purchases at 30 per cent of respondents' organizations. That number was at 19 per cent just a year earlier [6].

Last year, a survey conducted by KLAS Research found 70 per cent of healthcare organizations in the US have moved at least some applications or IT infrastructure off-premises and over half of those healthcare organizations have moved their electronic medical record (EMR) applications to a hosting or cloud environment [4]. Among the surveyed healthcare provider organizations, cloud infrastructure providers were the most commonly considered.

In Australia, a white paper by Medical Director revealed the benefits of cloud-based health solutions, with, 64% of respondents admitting they consider flexibility to be the main benefit of using a cloud-based HER/PMS system.

Many in the medical industry are unsure about the security of moving sensitive patient information around, especially now given the rise of cyber-attacks, data breaches and tougher cyber security laws. In fact, 76% consider the security of information being stored or sent their greatest area of concern in regards to managing patient information.

Although 80% of respondents recognized the value of real-time collaboration, and data sharing as having the potential to optimize the industry, lack of understanding and anxieties still exist around security, presenting significant hurdles to its industry-wide uptake [7]. Meanwhile only a third of respondents were satisfied with the current flow of information between their practice and other healthcare providers.

But that increased flow of information also brings risks that health IT professionals need to address. Those risks include possible harm to the patient's safety and health, loss of PHI and unauthorized access to devices.

The collaboration among healthcare professionals to ensure healthcare IoT security is an indication that the risks are not hypothetical. When it comes to healthcare IoT security issues, the list can seem overwhelming. One problem is devices entering hospitals the One example of

this is BYOD. When this happens, it can be difficult to figure out the lifecycle management of that device and identify the operating system. Although a variety of channels, with some of these avenues being unknown. Furthermore, "because [devices] come in through a different process, they wouldn't necessarily have any common controls surrounding them, meaning having passwords, encryption.

In the coming decades, the healthcare industry is predicted to grow at an unprecedented rate, and so is the data associated with it. It is expected to produce petabytes, exabytes or even zettabytes of data through the information collected from EHRs, laboratories, medical equipment and from the patient themselves. It will become more challenging for IT industries to analyze such enormous amount of data and turn all this into actionable medical sights. Embedding this Big Data to the healthcare sector and housing it on Cloud offers an effective solution to these issues.

Up till now the collection of data is limited to the major available resources in the healthcare sector. However, with the advent of smartphone apps and wearables, data is now everywhere. And this allows practitioners to know patients' health conditions in a more precise manner. Apps that act like pedometers to measure your steps, the calorie counter for your diet, the app for monitoring and recording heart rate, blood pressure and blood sugar levels, and wearable devices like Fitbit, Jawbone etc. are all sources of data nowadays. In the near future, the patient will share this data with the doctor who can utilize it as a diagnostic toolbox to provide better treatment in less time. This raises the trend of a partnership between medical and data analyst like Pittsburgh Health Data Alliance — where they are committed to collect record from various health-related instruments.

A lot of data is produced on a routine basis by hospitals, laboratories, retail, and non-retail medical operations and promotional activities. But most of it gets wasted because respective persons are not able to figure out what to do with that data. This is where Cloud-based Big Data comes into the picture. The big data analytics tools and repositories remove the hard thinking and generate reliable and calculative insights out of huge volumes of data within a matter of seconds. This means in the future we will need more doctors who are trained to work with big

data. The big data revolution is bringing up sophisticated methods of consolidating information from tons of sources. The focus is on providing the most relevant and updated information to doctors and medical practitioners in real time while they are consulting their patients.

Privacy is essential in the healthcare industry. Healthcare practitioners of all kinds are bound by professional and ethical standards—not to mention the law, in most jurisdictions—to keep patient information confidential and secure. This includes not only anything learned through conversations with patients, but also data kept on file—medical records, credit card information, social security numbers and more. If you're sending, receiving or holding sensitive information, you are responsible for keeping it safe and secure. If your security is compromised, there may be serious implications for you as a healthcare provider, your patients and their families, including embarrassment, personal identity loss, and even legal ramifications.

Patients' medical data is highly personal and needs to be protected and secured against loss and theft. Recently the largest healthcare data theft took place where cyber thieves have stolen medical records of 80 million patients from Anthem. With the improved computing technology and enablement of big data in the healthcare sector, privacy and security issues can be dealt with prima facie. The centralization of medical data is a subject of concern but big data certainly plays a key role in the development of new treatments and drugs as long as privacy and security can be maintained.

If an outsider gets to know the weakness of a person it can be danger for the patient if the outsider wants to cause any harm. The outside may know if the other person has any serious kind of illness which he may not want others to know. In some cases, the sickness can isolate a person from society due to superstition. Violation of privacy can make a patient's life more miserable. Here if privacy is maintained properly, it'll be easy for a patient to open up to his/her doctor. Which is important. Privacy can ensure socially beneficial activities like health research. Individuals are likely to participate in and support research if they believe their privacy is being protected.

As the high risk of medical identity theft continues to threaten consumers' financial and health records, patients should be more proactive in guarding against unauthorized use of their personal

information. With the recent string of data breaches in the health care industry, patients are more vulnerable to identity theft that could cause potentially harmful medical errors and denial of medical care.

If an attacker is able to break into a work device, encryption renders files useless by masking them into an unusable string of indecipherable characters. From a security standpoint, encryption is essential to keep your patients' protected health information (PHI) safe.

The Health Insurance Portability and Accountability Act, contains various protections mandating confidential handling of patients' protected health information.

- To set up or confirm appointments
- To discuss medical conditions with specialists or other practitioners
- To send or receive lab results
- To send patient information to medical billing or insurance organizations

HIPAA regulations state that data encryption is "addressable" and that it is essentially up to an individual organization to decide if encryption is needed.

If implementing data encryption is a reasonable measure, then HIPAA endorses it. The downside is that organizations that should execute encryption and do not could be held accountable for security breaches that could have been prevented through implementation of encryption.

Encrypting is similar to hashing because data is run through a mathematic algorithm; however, we use an encryption key that has a paired decrypting key. This way the data is safely stored and the only way to see the data is by using the decryption key to unlock it.

Encrypting data should be a baseline measure. However, using multiple encryption keys will help organizations keep their data more secure. That way if one key is compromised, not all of the data is compromised.

We can protect our data by encrypting. But what we will do if we want to search in that encrypted data and find a specific information.

In the cloud based-health care system, patients are outsourcing their sensitive personal information in the cloud which is owned by a third-party. Analyzing the information, doctors are also uploading the prescription for patients. Here the question comes, can the patients or doctors trust the third party with their sensitive information. As they cannot trust, they need the privacy for their information. Keeping the data encrypted in the cloud can ensure their privacy. But the next question that comes in our mind that if the doctor needs to search a particular information or a patient need to search for particular doctor's prescription how the search operation will be operated in the cloud without giving the authority to decrypt to the server. It is not a wise idea to download all the data and decrypt in the device and get the information. The good solution can be searching on the encrypted data. And this is known as searchable encryption which is first time introduced by Boneh et al in 2004 [1].

1.2 Motivation

The use of cloud to store data is increasing day by day, so does the need to search efficiently and securely in the cloud. Though cloud storages are giving users unlimited data storage and computational power at a cheap cost, it raises privacy issues to users. If the data is outsourced to a third-party cloud storage, it is exposed not only to third party intruders but also to careless or even potentially malicious Cloud Service Providers (CSPs). General encryption can protect the data from the intruders in cloud. At the same time, it prevents users to search on encrypted data. If users need a particular piece of information where general encryption has been used, users have two option to perform the search operation and get the data. These options are-

1. User can download all the data and decrypt the to search over the data in his own device.

2. User can give the remote server the secret key to encrypt the data and let the server decrypt and search.

In the first scenario, when the amount of data is huge, solution is not good at all. In case of multiple users, the process become more complicated. Users will experience a large amount of communication overhead.

In case of second option, users need to trust the remote server as they are giving their secret key to the remote server to decrypt and search the data. Having the secret key, admin of the remote server can do harmful acts like knowing secret information of the users, changing information.

The notion of Searchable Encryption (SE) introduce a promising solution to keep the outsourced data protected from an unauthorized accessed by CSPs or other intruders. Encrypted data is linked with encrypted keywords, sometimes called search token in such a way that a remote server, which is given encrypted keywords, can check whether a record has keywords that satisfies the search term. Such schemes allow the CSP to perform encrypted search on encrypted data without leaking the content of the query and the data.

Now if we look into the medical sector of present world, a lot of patients are sending their information in the cloud using IoT devices. Each and every minute, update about patient's condition have been sent to cloud storage. Again, checking their condition doctors prescribe them. Prescriptions are also uploaded in the cloud. Healthcare industry growing so fast and as a result, this sector is becoming a sweet-spot for hackers to steal large amounts patient records. A hacker can hack the cloud and do different acts for which a patient may have to pay big price.

1.3Thesis Contribution

In this paper we study the problem of how to resist inside keyword guessing attack in PEKS and try to solve it and implemented on healthcare system. Based on the observation that in a KGA attack the server is able to encrypt each keyword candidate and test its ciphertext with the given trapdoor, we suggest a method to prevent the server from doing so. Roughly, we make the following contributions in the paper.

1. We introduce the notion of Designated Senders Public-key Searchable Encryption with Keyword Search, in which the data sender is authenticated by the receiver before encrypting the data, so that the server cannot encrypt a keyword itself and thus cannot launch the inside keyword guessing attack successfully.

2. We present the security model of PAEKS, and propose a concrete construction. Security of the scheme is proved in the given security model based on simple and static assumptions, for

example Decisional Bilinear Diffie–Hellman assumption and a simple variant of Decision Linear assumption, with the help of random oracles.

3. We compare our scheme with some other related PEKS schemes in terms of both computation and communication efficiency. We also do some experiments to demonstrate the efficiency of our scheme. Experimental results show that its efficiency is comparable with that of Boneh et al.'s scheme.

1.4 Paper Organization

In the next section i.e. in Section 2 We review the different existing scheme and show its security weakness. In Section 3, we describe some preliminaries related to PEKS. In Section 4 we define the basics of searchable encryption. We then propose our new method of PEKS in Section 5. In Section 6 we compare our scheme with Boneh et al.'s scheme and show the experiment results. Finally, we conclude the paper in Section 7. Section 8 and 9 respectively acknowledgment and references.

Related Literature

In this section, we present the related existing SE protocols. First, we will present few existing SE protocol by paper and then categorized them according to their design goals.

Suppose someone wants to send a message to Alice. In order to send a message M to Alice with keywords W1…. Wk, an employee sends[24]:

$W_{Apub}(M)$ || PEKS (A_{pub} , S_{pub}, W_1)||…….. || PEKS(A_{pub},S_{pub},W_k)

to the sever. where Apub is Alice's public key and Spub is Server's public key. Alice gives the server a trapdoor Tw via the public channel. Server can use Tw to select the desired emails. In the original scheme, A certain keyword corresponds to a constant trapdoor. An attacker who intercepts and captures the communications can statistics the frequency of occurrences of these trapdoors, can choose the highest frequency trapdoor to attack [6]. Once the attack is successful, an attacker may know Alice's privacy interests. Here in this model the first thing which is done is removing the secure channel for sending trapdoor. This means that the trapdoors can be sent via a public channel (between receiver and server). Another thing is that an outside attacker can't determine which PEKS cipher text encrypts which keyword even if the attacker gets all the trapdoors for the keywords [24].

Trapdoor indistinguishability: Trapdoor does not reveal any information without the server's private key. Here the trapdoor is updated every time and attacker can't distinguish two trapdoors came from the same keyword [2].

Simple encryption prevents searching on encrypted data. 3 different types of keyword search systems over encrypted data. One is Public storage system where data is stored in remote database and not reveal anything to the administrator. User retrieve information with particular keyword. Another is Vendor System where the solution is offered by PIR protocols. 3rd one is The Store- and forward system where user can search info that is encrypted under user's public key.

But the problem of PEKS is that it assumes secure channel between receiver & server. The solution for this problem is given in Beak et al. [11] 's first proposed a secure channel-free public key encryption with keyword search (SCF-PEKS). But there is also a problem in the solution. Here given a trapdoor attacker can determine the corresponding keyword. Later "trapdoor indistinguishability" is used to solve it.

The problem with PEKS is that it is vulnerable to keyword guessing attack (KGA). Thus, using public key anyone/ server can compute the ciphertext of any keyword. Malicious server prepares cipher text of possible keywords, test If keywords of trapdoor (sent by receiver) the same as the keywords of the prepared cipher text. Thus, server can guess the keyword of the trapdoor.

Thus Designated – senders PEKS is proposed [25]. Here receiver can designate a group of senders who can encrypt keywords so that Server can't encrypt the keyword. Thus, no inside keyword guessing attack happens. This is constructed using Broadcast Encryption.
In the proposed Designated Sender PEKS [25], receiver can designate a set of senders. {S}, A trusted key center is needed, generates and distribute secret keys to users as senders, Key generation of receiver - Public and Secret key corresponding to {S} is generated, Senders {S} can generate cipher text only. Here a secret session key is shared between the senders in $\mathbb{S}$ and the receiver via the receiver's public key including the broadcast encryption.

To protect PEKS from KGA, other types of PEKS have been proposed [3,7,8,12,13]. One is PEKS scheme with registered keywords (PERKS) where receiver should perform keyword registration algorithm [3] in the pretreatment process and send the pre-tag to the sender through a secure channel.

- Receiver computes pre-tag from any available keyword
- Sender is given the pre-tag
- Without pre-tag, no one can generate the cipher text.

But it also suffers with some problems like for lots of keywords, sender has to be given lots of pre-tags, secure channel for communication pre-tags (because server may wiretap) is needed and it is not considered for multiple senders

Another solution is PEKS with fuzzy keyword search [13] where in addition to the original trapdoor, a fuzzy trapdoor is generated by the receiver. A fuzzy test function allows the server to fuzzily check the keywords of a cipher text and the trapdoor are the same. Since multiple keywords share a fuzzy trapdoor. Thus, the server cannot guess the exact keyword of the trapdoor. The problem is solved using two trapdoors. It is then showed that it is secure against inside KGA. But the problems of this proposed method are low efficiency and poor security, server still can know small number of keywords including correct one and server responds multiple matched ciphertexts to the receiver, the receiver must locally find the exact ciphertext using the exact trapdoor.

Another solution method is Dual server PEKS [7] where server is separated to a front server and a back server. Front server is given a ciphertext and trapdoor thus it generates an internal test state. The back server is given the test state and it outputs the test result. Each server does not know the test result from a pair of cipher text-trapdoor. Thus, it is shown that it is secure against inside KGA. But it has also some limitations. By colluding both servers, an attacker can launch KGA. Low efficiency and poor security is another issue.

Another solution approach is SCF (Secure-Channel Free)-PEKS [8] were it considers the trapdoor secrecy without assuming the secure channel between a server and a receiver. Here a server (tester) generates his public key and secret key. Thus, only a designated server with the secret key can test. Any outsider that is not the designated server cannot launch KGA. But the problem is that the designated server can launch KGA.

An inside adversary may recover the keyword from a given trapdoor by exhaustively guessing the keywords offline. Thus, PEKS scheme is vulnerable to file injection attack. But PEKS is secure based on the bilinear Diffie-Hellman assumption. The Keyword space is assumed to have low min entropy and thus Keyword guessing attack can happen.

Advantages of this model in [25] is that it Doesn't require key distribution and management. Here Data sender not only encrypts a keyword but also authenticate it. Problem of this model is that it fails to achieve the ciphertext indistinguishability and vulnerable against the chosen keyword attack.

There is Offline keyword guessing attack vulnerability on existing PEKS [2,9]. It is because:

- The keyword space is assumed to be at least super-polynomial large but is small in real applications.
- Keywords are often chosen from a low-entropy keywords space.
- Adversary tries each possible keyword, encrypts it, and tests the ciphertext with the given trapdoor.

KGA works for two reasons:

1.Adversary could get the trapdoor. Thus, the solution is setting up a secure channel

2. Other is that it can do test freely. Thus, the solution is resisting unauthorized adversary from doing test. For these two models are proposed. One is Designated [7,8] tester PEKS and the other is PEKS with authorization [9,10]. But the problem for both of these is neither method can prevent an inside adversary to launch an attack

PEKS variants are of the following types:

1. Multi-user access control in PEKS [9,10,22]
2. Fuzzy keyword search in PEKS [14,15,16]
3. Flexible keyword search in PEKS [17,18,19]
4. Trapdoor privacy in PEKS [2,9,20,21]

The only scheme that counterattacks the inside KGA makes use of two servers and assume that those two don't collude is Keyword search encryption scheme resistant against keyword-guessing attack by the untrusted server [2,9]

There are several ways to leak information about the keyword:

1. Search result
2. No of records satisfy the search query

3. Trapdoor may leak info [20]
4. Keywords may be revealed due to leakage search pattern [21] etc.

The contribution of this paper is that here in this PAEKS scheme sender authenticates the data (server can't encrypt a keyword itself)

SSE includes complex & expensive key management & distribution and PEKS suffer from inefficiency and inside KGA. Most protocols are insecure against file-injection attacks carried out by malicious servers.

A solution for this problem is given in [26] where the protocol uses trapdoor permutation function (TPF), it incurs lower computation cost at the expense of a slightly higher storage cost. The protocol uses neither bilinear pairing nor map-to-point hash operation thus complexity is reduced. The search time of the protocol is related to the database update times. This proposal achieves inside KGA, forward privacy and file injection attack resilience.

To maximize efficiency and minimize search time and avoid having complex operation multiple solution approach is proposed. In [32] search time is linear to database size. Thus, it is inefficient in big data environment. In [28] index-based SSE is used to achieve sublinear search time. Here complexity is related to keyword space. Another model is proposed in [29] where highly scalable SSE is used to support large database. In most PEKS complex operation (bilinear pairing and map-to-point hash) affect efficiency. But in [30] no bilinear pairing operation is used. But it is not practical protocol. Another one is [31] where it doesn't require bilinear pairing but uses homomorphic smooth projective hash function. Thus, it is more efficient. For security, in [1] it is shown that the model protects user search pattern and access pattern.

ID-PKC (Public-key infrastructure) is a significant platform that could provide public key encryption and digital signature services[27]. Certificate Authority (CA) is the kernel part of PKI system, which is responsible for issuing and managing all users' digital certificates. In [11] the need for certificate management is reduced.

A trusted KGC is indispensable part which to generate user's private key. The problem here is that KGC could decrypt any ciphertext and forge any message's signature which is defined as key escrow problem.

In this paper a solution is proposed where a KGC generates a partial private key, user chooses a random number as its secret value and user's private key is generated by the above two parts and it can't be obtained by the KGC.

There are some practical techniques for searches on encrypted data which is proposed in the paper [9]. They first proposed the searchable encryption based on stream cipher and symmetric encryption where a document is divided into 'words', and then they are encrypted respectively. Here sequential scan the whole ciphertext is done.

But problems of this paper demonstrated in [33] showed that it can't resist the attack of a malicious KGC and offline keywords guessing attack

The contribution of this paper is that it proposes a new CLPEKS scheme. It can resist KGA under two types of adversaries 1. Adversary who can replace any users public key 2. adversary can access to master key. The computation cost and communication cost of our CLPEKS scheme is lower than other schemes [35] [34].

We can see that these algorithms can be categorized by efficiency of algorithms, applications and security.

Efficiency: The search time is one of the key factors in determining the efficiency of SE protocol. Song's [29] search time is linear to the database size; thus, the protocol is inefficient in a big data environment. To mitigate such a limitation, Curtmola et al. [15] presented an index-based SSE construction to achieve sublinear search time. The complexity of the index-based protocol is related to the keyword space. In 2013, Cash et al. [9] presented a highly-scalable SSE, designed to support very large database. From existing literature, a practical SE protocol should minimize the search time, which is not a surprising observation. However, in most PEKS protocol, complex operations (e.g. bilinear pairing, map-to-point hash) affect efficiency. To reduce the computational complexity, Di [16] proposed a PEKS protocol

without bilinear pairing operation. However, the protocol is not practical. Recently, using homomorphic smooth projective hash functions, Chen [13] designed a protocol that does not require the bilinear pairing operation. Thus, their protocol is more efficient. In other words, the design of a SE protocol should avoid having complex operations.

Application: SE protocol should adapt to a variety of application scenarios. Boneh [5] first designed a PEKS protocol for the email system based on public key cryptosystem. To facilitate data sharing, multi-owner SE protocols [32,36] were proposed. Such protocols generally allow multiple data owners to contribute data to multiple receivers. In practice, a typical scenario is for each data receiver to belong to a different access authority (e.g. different service providers). Thus, SE protocols should have flexible access control strategies [11]. Zheng [27] defined an attribute-based keyword searchable encryption protocol, which allows an authorized user to search over the outsourced data. Liang [24] also designed a protocol that offers searchable attribute-based functionality and flexible keyword update service. Thus, the design of SE protocol should consider the application environment.

Security: Security is the most important property in the design of any cryptographic protocols. For example, Goh [19] provided the security definition for SE protocols, and reiterated the importance of security in SE protocols. To prevent privacy leakage, Abdalla et al. [1] suggested that user search pattern and access pattern should be protected. Islam et al. [21] and Cash et al. [8] demonstrated that information leakage can be abused by a passive attacker to reveal the user's sensitive information. Recently, Zhang et al. [37] demonstrated that an adaptive attack can reveal the content of a past query by inserting as few as 10 new files in the dynamic database. Thus, SE protocol should be designed to prevent or reduce information leakage. Meanwhile, most PEKS protocols are vulnerable to inside KGA [7,22], which is usually launched by a malicious cloud server. In 2013, Peng [35] presented the public-key encryption with Fuzzy keyword Search (PEFKS), designed to withstand inside KGA using two trapdoors. A number of such protocols have been designed for the cloud. However, the dynamic nature of the cloud environment may in itself be a risk. For example, due to the dynamic nature of database on the cloud, the updating operations of the database may result in (accidental) information leakage. This is an area of active research. For example, to protect the security of dynamic databases, a number of SE protocols with forward privacy have been presented in the literature [6,34].

Preliminaries

3.1 Mathematical Notations

Sets and rings: The set of integers is denoted Z, the set of non-negative integers N. If a and b are two integers such that $a \leq b$, we denote by [a; b] the set $\{x \in Z | a \leq x \leq b\}$ of integers between a and b (both included). Also, Zn is the ring Z/nZ of integers modulo the integer n. For a prime integer p, Fp = Zp is the field with p elements. Ᵽ() is the Euler totient function. For any set S, P(S) is the set of all finite subsets of S.

Bilinear groups: Some of the cryptographic primitives will use an additional structure on groups called a pairing (a.k.a. bilinear groups). Let G_1, G_2 and GT be three cyclic groups of the same order N and with respective generators g_1, g_2 and g_T. The quadrule (G_1; G_2; G_T; e) is called a bilinear group if e: G1 × G2 → GT satisfies the following properties:

- For all (a, b) $\in (ZN)^2$, $e(g^a_1 ; g^b_2) = e(g_1; g_2)^{ab}$ (bilinearity);
- The element $e(g_1; g_2)$ generates G_T (non-degeneracy);
- e(.;.) is "efficiently" computable.

Diffie-Hellman: Diffie–Hellman key exchange (DH) is a method of securely exchanging cryptographic keys over a public channel. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Inside Keyword Guessing Attack: Suppose the cloud server provider (CSP) is an honest but curious adversary. We assume that the keyword space has a polynomial size. Upon receiving a trapdoor Twi CSP tries to recover the keyword wi corresponding to Twi . The adversary first encrypts a possible keyword wi* to generate a ciphertext Cwi*. Then It runs the test algorithm of the existing schemes based on the traditional framework of PEKS. The inputs of the algorithm are Cwi* and Twi. If the test algorithm returns 1, this result illustrates that the adversary's guess is correct. Otherwise, the adversary continues to repeat the process until it finds the correct result.

File Injection Attack: Local File inclusion (LFI), or simply File Inclusion, refers to an inclusion attack through which an attacker can trick the web application in including files on the web server by exploiting functionality that dynamically includes local files or scripts. The consequence of a successful LFI attack includes Directory Traversal and Information Disclosure as well as Remote Code Execution.

The attack consists of two phases, including the GenFile phase and the Guess phase. The GenFile phase is in charge of generating the files to be injected. The Guess phase is responsible for guessing the keyword wi corresponding to the trapdoor Twi.

The attack works as follows:

GenFile Phase:

- Identifies the keyword space W with {0, |W|−1} written in binary.
- Generates the injected file Fi that contains the keywords whose ith bit is 1, where i = 1, 2, . . . , log |W|.

Guess phase:

- takes as input the trapdoor Tw which wishes to be guessed and runs the test algorithm of SPE schemes to obtain the returned files.
- determines the keyword w corresponding to Tw based on the returned result.

3.2 Cryptographic Preliminaries

Security parameter: In order to properly formalize security notions, we need to bound the computing power of an attacker. Indeed, one can always break cryptosystems using a large enough computer and spending a high amount of time. However, in cryptography, we restrict ourselves to the defense against reasonable attackers. To do so, we use the notion of security parameter, denoted $\lambda \in \mathbb{N}$. The security parameter is passed as an input to the attacker, under its unary representation 1^{λ} and we only consider attackers whose running time is polynomial in λ, and whose success probability is non-negligible in λ.

Adversaries: An adversary is a probabilistic Turing machine, which, in this manuscript, run in polynomial time, which may carry a state when they need to be called several times. In most cases, we implicitly give as input to the adversary, both the unary representation of the security

parameter, and the state. As adversaries' inputs are always polynomial in the security parameter, the polynomial time adversary runs in time polynomial in the security parameter

Games: Security notions are often defined using security games (or experiments). Simple games are defined by having an adversary accessing a set of oracles, sometimes with some restrictions on calls to these oracles, and the output of the game is defined as the output of the adversary. More generally, games are defined using the code-based games formalism introduced in [10]. Such a game G is a set of oracle procedures – including an initialization Init procedure and a finalization Final procedure – that is executed with an adversary A, i.e. A has access to the procedures, with some possible restrictions. For instance, the Init oracle is always the first one to be called and Final the last one, once A halted, taking A's output as input. The output of Final is called the output of the game and is denoted $G^A(1^\lambda)$. When Final is omitted, it just forwards the adversary's output.

In those games, at startup, the Boolean variables are initialized to false and the integer variables to 0.

3.3 Cryptographic Primitives

Pseudorandom Function: In cryptography, a pseudorandom function family, abbreviated PRF, is a collection of efficiently-computable functions which emulate a random oracle in the following way: no efficient algorithm can distinguish (with significant advantage) between a function chosen randomly from the PRF family and a random oracle (a function whose outputs are fixed completely at random). Pseudorandom functions are vital tools in the construction of cryptographic primitives, especially secure encryption schemes.

Pseudorandom functions are not to be confused with pseudorandom generators (PRGs). The guarantee of a PRG is that a single output appears random if the input was chosen at random. On the other hand, the guarantee of a PRF is that all its outputs appear random, regardless of how the corresponding inputs were chosen, as long as the function was drawn at random from the PRF family.

A family of functions,

f_s: $\{0, 1\}^{\lambda(|s|)} \rightarrow \{0, 1\}^{\lambda(|s|)}$, where $s \in \{0, 1\}^*$, and λ: $\mathbb{N} \rightarrow \mathbb{N}$,

is pseudorandom if the following conditions are satisfied:

- Given any s and x such that $|x| = \lambda(|s|)$, there always exists a polynomial-time algorithm to compute $f_s(x)$.
- Let F_n be the distribution of functions f_s where s is uniformly distributed over $\{0, 1\}^n$, and let RF_n denote the uniform distribution over the set of all functions from $\{0, 1\}^n$ to $\{0, 1\}^n$. Then we require F_n and RF_n are computationally indistinguishable, where n is the security parameter. That is, for any adversary that can query the oracle of a function sampled from either F_n or RF_n, the advantage that she can tell apart which kind of oracle is given to her is negligible.

Pseudorandom Permutation: In cryptography, a pseudorandom permutation (PRP) is a function that cannot be distinguished from a random permutation (that is, a permutation selected at random with uniform probability, from the family of all permutations on the function's domain) with practical effort. An unpredictable permutation (UP) F_k is a permutation whose values cannot be predicted by a fast randomized algorithm. Unpredictable permutations may be used as a cryptographic primitive, a building block for cryptographic systems with more complex properties.

Let F be a mapping $\{0,1\}^n \times \{0,1\}^s \rightarrow \{0,1\}^n$. F is a PRP if

- For any $K \in \{0,1\}^s$, F is a bijection from $\{0,1\}^n$ to $\{0,1\}^n$.
- For any $K \in \{0,1\}^s$, there is an "efficient" algorithm to evaluate $F_K(x)$.
- For all probabilistic polynomial-time distinguishers D: $|Pr(D^{F_K}(1^n) = 1) - Pr(D^{f_n}(1^n) = 1)| < \varepsilon(s)$, where $K \leftarrow \{0,1\}^n$ is chosen uniformly at random and f_n is chosen uniformly at random from the set of permutations on n-bit strings.[1]

A pseudorandom permutation family is a collection of pseudorandom permutations, where a specific permutation may be chosen using a key.

Unpredictable Permutation: An adversary for an unpredictable permutation is defined to be an algorithm that is given access to an oracle for both forward and inverse permutation operations. The adversary is given a challenge input k and is asked to predict the value of F_k. It is allowed to make a series of queries to the oracle to help it make this prediction, but is not allowed to query the value of k itself.

A randomized algorithm for generating permutations generates an unpredictable permutation if its outputs are permutations on a set of items (described by length-n binary strings) that cannot be predicted with accuracy significantly better than random by an adversary that makes a polynomial (in n) number of queries to the oracle prior to the challenge round, whose running time is polynomial in n, and whose error probability is less than 1/2 for all instances. That is, it cannot be predicted in the complexity class PP, relativized by the oracle for the permutation.

Trapdoor function: Trapdoor Function: A trapdoor function is a function that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information, called the "trapdoor". The idea of trapdoor function. A trapdoor function f with its trapdoor t can be generated by an algorithm Gen. f can be efficiently computed, i.e., in probabilistic polynomial time. However, the computation of the inverse of f is generally hard, unless the trapdoor t is given.

Basic of Searchable Encryption

A searchable encryption is the process to search on the encrypted data in the cloud without decoding the data. There are two types of searchable encryption just like encryption. Basically, these types are based on the number of keys for encryption. Two types of searchable encryptions are Symmetric Searchable Encryption (SSE) and Public key Encryption with Keyword Search (PEKS). PEKS is just an implementation of asymmetric cryptography where pairs of keys are used to encrypt and decrypt. Two keys are called public key and private key respective. Public is used for encryption and private key used to create trapdoor to search on the cloud.

Keeping the private key private; the public key can be openly distributed without compromising security. Though in public key searchable encryption, two keys need to generate, it has some benefits over SSE scheme. Such as-

1. Public key algorithms, unlike symmetric key algorithms, do not require a secure channel for the initial exchange of one or more secret keys between the parties.
2. Because of the computational complexity of asymmetric encryption, it is usually used only for small blocks of data, typically the transfer of a symmetric encryption key (e.g. a session key- used to encrypt the rest of the potentially long message sequence.

In a public key signature system, a person can combine a message with a private key to create a short digital signature on the message. Anyone with the corresponding public key can combine a message, a putative digital signature on it, and the known public key to verify whether the signature was valid, i.e. made by the owner of the corresponding private key. Changing the message, even replacing a single letter, will cause verification to fail. In a secure signature system, it is computationally infeasible for anyone who does not know the private key to deduce it from the public key or any number of signatures, or to find a valid signature on any message for which a signature has not hitherto been seen. Thus, the authenticity of a message can be demonstrated by the signature, provided the owner of the private key keeps the private key secret. Public key algorithms-

- underpin various Internet standards, such as Transport Layer Security (TLS), S/MIME, PGP, and GPG.
- provide key distribution and secrecy (e.g., Diffie–Hellman key exchange), some provide digital signatures (e.g., Digital Signature Algorithm), and some provide both (e.g., RSA).
- finds application in Information security (IS) (concerned with all aspects of protecting electronic information assets against security threats.)
- assuring the confidentiality, authenticity and non-reputability of electronic communications and data storage.

In PEKS, suppose Alice wants to send medical records M with some keywords W = {w 1, w 2, ⋯, w n} to Bob through an untrusted cloud server. Alice uses Bob's public key to generate encrypted data Enc (M) and the encrypted keyword index ind {W}. Then, Alice sends both Enc (M) and ind {W} to cloud. If Bob wants to search the medical records comprising keyword w, he first uses his own private key to generate the trapdoor T w of keyword w, and sends it to server. Once receiving T w, server will start to test keyword index ind {w} matching the trapdoor T w, and returns the corresponding encrypted records Enc (M) to Bob.

- Receiver generates public and secret key
- Sender encrypts a keyword using public key
- Sender attach PEKS cipher text of the keywords to the encrypted content of the data.
- Receiver derives a token (trapdoor) to search a keyword from secret key.
- Server tests if the keyword of the cipher text is equal that of the trapdoor

PEKS primarily comprises of (KeyGen, PEKS, Trapdoor, Test)

- 1. KeyGen(s): Takes a security parameter, s, and generates a public/private key pair Apub;Apriv.
- 2. PEKS(Apub;W): for a public key Apub and a word W, produces a searchable encryption of W.
- 3. Trapdoor(Apriv;W): given Alice's private key and a word W produces a trapdoor TW.

- 4. Test(Apub; S; TW): given Alice's public key, a searchable encryption S = PEKS(Apub;W0), and a trapdoor TW = Trapdoor(Apriv;W), outputs `yes' if W = W0 and `no' otherwise.

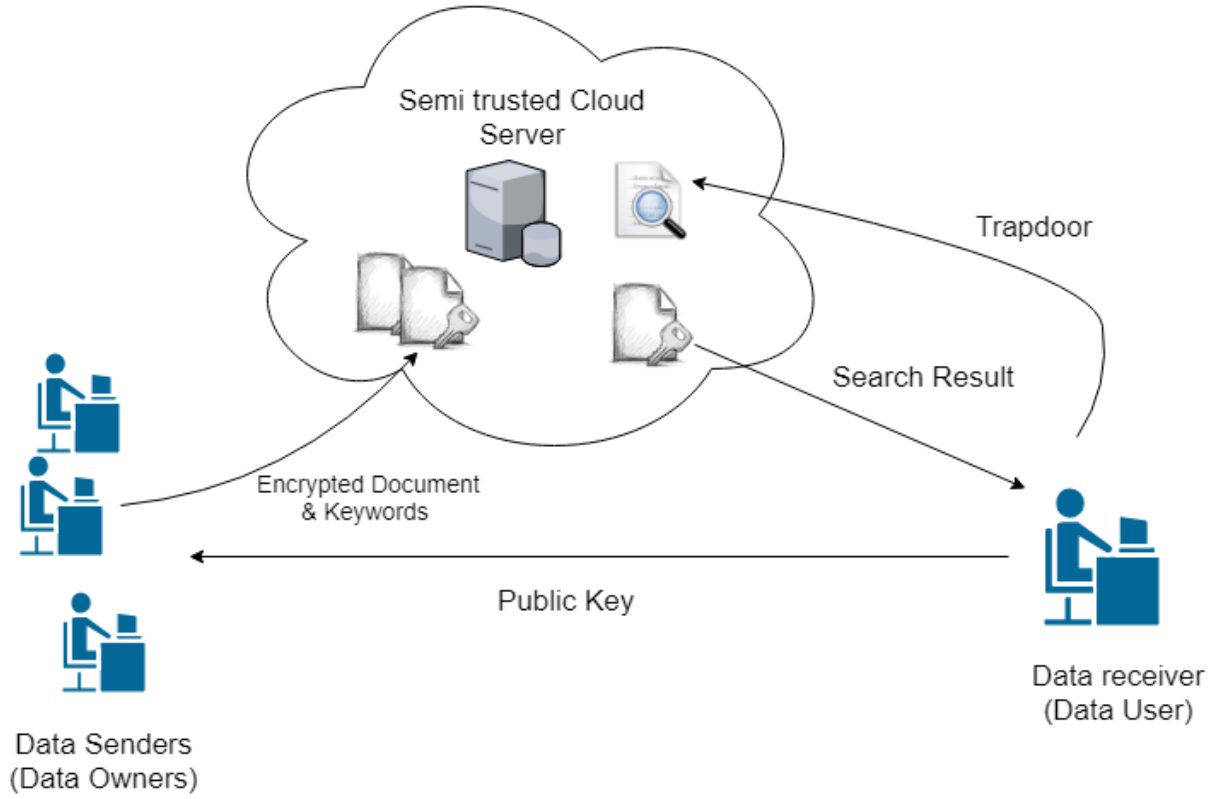

Figure 4.1: Public Key Encryption with Keyword Search

Proposed Method

In this section, present our proposed Designated senders public key searchable encryption for health care system.

5.1 System Model

System model of our method is described in fig. 4.1, which comprise of four entity. These entities are data sender, data receiver, cloud server and key center. From the fig. 4.1, patients will send data and

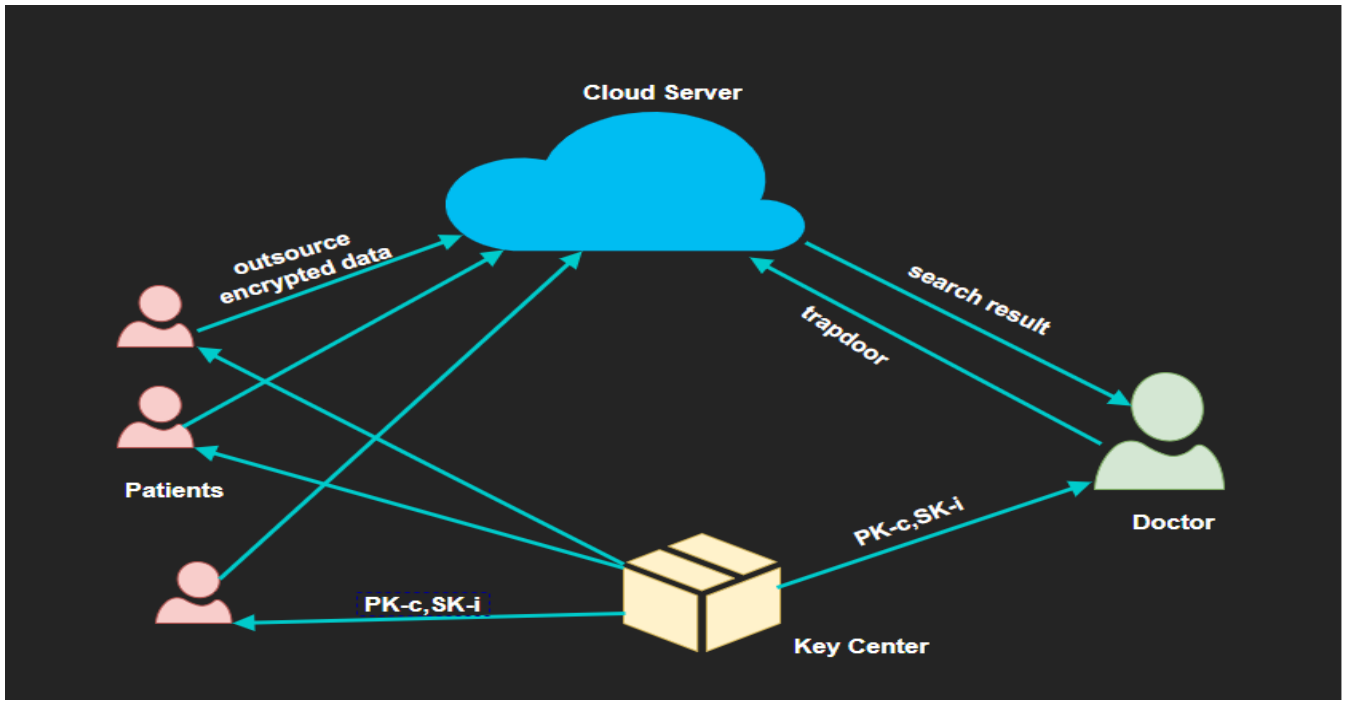

Figure 5.1 System Model

Initialize the process and doctors are the receiver of the data. Each entity has different responsibilities, as described below:

- Key Center will create keys for both data sender and data receiver to make data sender designed one.

- Data sender encrypts the files and uploads to the cloud
- Data receiver generates trapdoor and search data by sending trapdoor
- Cloud server is responsible for storing encrypted data and run test algorithm in the encrypted data.

5.2 Formal Definition

Our proposed algorithm consists of following algorithms:

- Setup (λ): Takes input security parameter λ and outputs system global parameter Ω and pk_c,sk_c.
- $KeyGen_S$ (pk_c, skc): Takes pk_c, skc as input and produce sk_i.
- $KeyGen_R$ (pk_c): pk_c is given as input and $pk_{R,S}$
- Enc: with pk_c and $pk_{R,S}$, keywords are encrypted and uploaded in the cloud.
- Trapdoor: to generated trapdoor pk_c, $pk_{R,S}$ and $sk_{R,S}$ is taken as input.
- Test: test algorithms run with input pk_c, $pk_{R,S}$, C and T.

5.3 Algorithm

Setup: Given security parameter ℓ and the number of users n.

1) DS chooses an additive group G1 with a large prime order q and a generator P.

2) Key Center selects a TPF f: X → Y, where X and Y are two sets of strings with length l. Then, executes Gen(1λ) to generate pk_c,sk_c. (pk_c,sk_c). $R \leftarrow- \mathbb{Z}$

$KeyGen_S$: Given pkC, skC.

1)Compute $di = g_i^{x'_C}$, and output $ski = (i, d_i)$ for all $i \in \{1, \ldots, n\}$.

$KeyGen_R$: Given pkC, $\mathbb{S} \subseteq \{1, \ldots, n\}$.

- Select $x_R \leftarrow- \mathbb{Z}^*_p$, and compute $yR = g^{xR}$.
- Select $t \leftarrow- \mathbb{Z}^*_p$, and compute the session key $K = e(g1, gn)t$. Compute the broadcast ciphertext

  $(E1 = g^t, E2 = (v \cdot \Pi_{j\in\mathbb{S}}\, gn{+}1{-}j)^t)$.
- Output $pk_{R,\mathbb{S}} = (\mathbb{S}, y_R, E_1, E_2)$ and $sk_{R,\mathbb{S}} = (\mathbb{S}, x_R, K)$.

Enc: Given pk_C, $pk_{R,\mathbb{S}} = (\mathbb{S}, y_R, E_1, E_2)$, $sk_i = (i, d_i)$, and the keyword w.

- Decrypt the broadcast ciphertext ($E1$,$E2$) to obtain the session key by K = e(gi,$E2$) / e(di ·$\Pi_{j\in\mathbb{S},j\neq i}$ $g_{n+1-j+i}$,E_1).
- As in the original PEKS, select $r \leftarrow - \mathbb{Z}^*_p$, and compute $C_1 = g^r$ and $C_2 = e(y_R, H(w\|K))^r$.
- Output the ciphertext C = (C_1, C_2).

Trapdoor: Given pk_C, $pk_{R,\mathbb{S}}$, $sk_{R,\mathbb{S}}$ = ($\mathbb{S}$, x_R,K), and the keyword w.

- As in the original PEKS, compute $T = H(w\|K)^{xR}$.
- Output the trapdoor T.

Test: Given pk_C, $pk_{R,\mathbb{S}}$, C = (C_1, C_2) and T.

- As the original PEKS, check the equation $e(C_1, T) = C_2$.
- Output 1 if the equation holds. Otherwise, output 0.

5.4 Security Requirements Analysis

Inside KGA Resilience: According to the description of our proposed protocol in Section 4, the encryption process requires as input the parameters PK_C , SK_C and the index set Iwi , where the secret key SKs of DO is used to generate the ciphertext of the set I_{wi}. Thus, a malicious cloud server is not able to generate ciphertexts of the keywords and test the given trapdoor without the secret key SK_i. SK_i is key to achieving inside KGA resilience. Thus, we say our protocol is insider KGA resilience.

FIA Resilience: To prevent the adversary from recovering the content of query based on the indexes of the returned result, our protocol randomly stores the newly inserted file and encrypts the index of inserted file by $CI_{(j+m)wi} = h_2(K, st_{(j+m)wi}) \oplus ind_m$. However, since the adversary may inject only one file at a time, it can still know the index of injected file corresponding to the encrypted index. Eventually, the adversary recovers the content of query by the returned encrypted indexes. For example, although Bost's protocol [6] also encrypts the index of file, it still cannot resist such an attack. To address this drawback, we add a padding index ind^* if $|I_{wi}| = 1$ in our protocol. The adversary is, therefore, unable to determine the mapping between the encrypted indexes and the plaintext index. Therefore, our protocol can resist the file injected attack.

Performance Evaluation

We will now present a comparative summary of the computation cost and security properties between our protocol and those presented in [5,34], based on bilinear pairing operation, the map-to-point, exponentiation, keyed hash function, scalar point multiplication, etc.—see Table 2. The notations used in the comparative summary are as follows:

1. Tp: Time cost for a bilinear pairing.
2. Tmtp: Time cost for a map-to-point hash function.
3. Tsm: Time cost for a scalar point multiplication operation in G1.
4. Texp: Time cost for an exponentiation operation in G2.
5. Tmul: Time cost for a multiplication operation in G2.
6. Te: Time cost for a encryption process of TPF, fsk(x) = y.
7. Th: Time cost for a keyed hash function.

As shown in Table 1, in our protocol, the computation cost of encryption and authorization phase, trapdoor generation phase and search phase are 1Te + 2Th, 3Tsm + 1Th and 1Td, respectively. Our protocol is the only protocol to achieve inside KGA resilience and FIA resilience. To achieve a baseline security level for comparison, the 1024-bit RSA algorithm is used as the trapdoor function, and the keyed hash function is instantiated using SHA-1. G1 with order q is generated by a generator on an elliptic curve E(Fp), where q and p are the 160-bits and 512-bits prime numbers, respectively. To evaluate the efficiency of the three protocols, we perform our experiments on a personal computer with a single Intel core i5 CPU 2.50 GHz, 8.00 GB of RAM, a 250 GB, Lenovo T410 running Windows 7. We run the related operations based on MIRACL library each for 100,000 times, and the average runtime is presented in Table 1. As shown in Fig. 4, the time cost in the protocols of Boneh**[1]** and Libing[34] is similar at the starting point. However, our protocol is more efficient at the encryption phase for large dataset, due to the fact that we do not require any complex operations, such as pairing and map-to-point. The protocol in [34] is the most inefficient due to the need for a pairing operation and a map-to-point hash function operation per encryption phase.

Also from Fig-5, we can see a similar case. Our scheme gives a good output for the larger data. As the initially computer needs to start up every things it takes a little more time for out scheme but later on it takes less time in our scheme.

Table-1 Runtime of main operations							
Operation	T_p	T_{mtp}	T_{sm}	T_{exp}	T_{mul}	T_e	T_h
Time (ms)	8.31	8.881	1.085	1.512	0.901	1.215	.75.

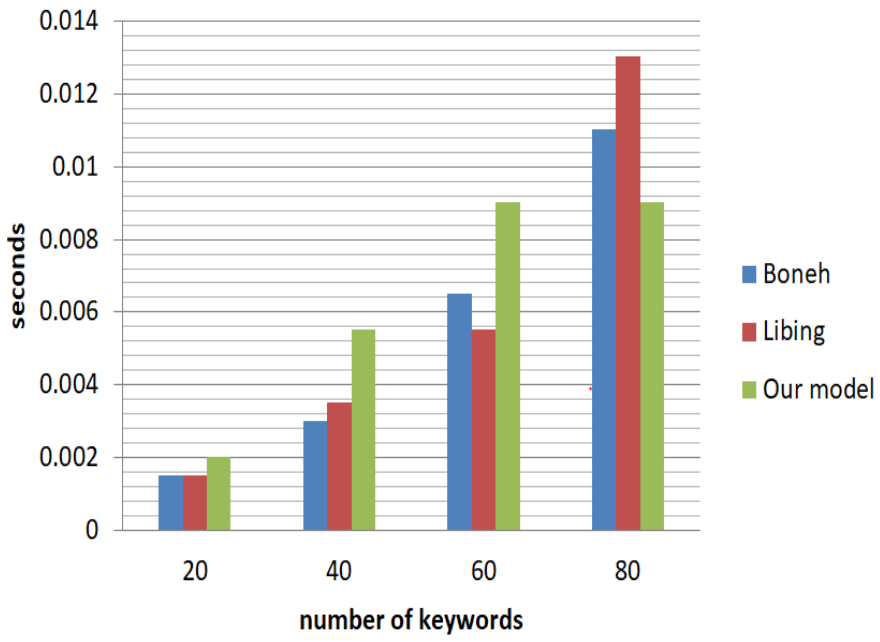

Figure-6.1 Computation Cost at Encryption

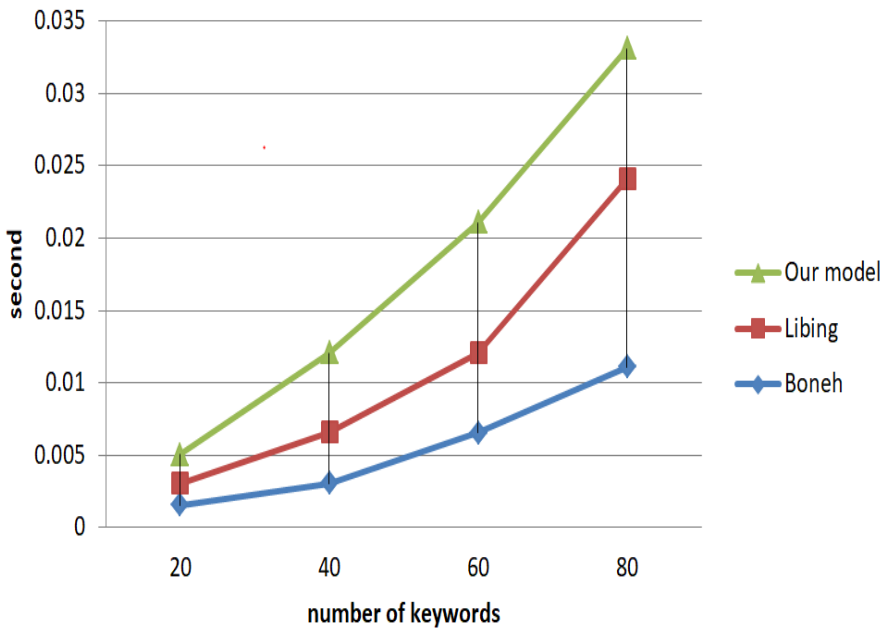

Figure-6.2 Computation Cost at Searching

Limitations

We have some limitations in our work. These limitations are:

- When set of patients are changed, doctor need to regenerate his/her key pair again.
- Every time doctor needs to send his public key to the new set of users.
- As we are still using bilinear function, computation overhead is more in the encryption part

Conclusion and Future work:

The system we proposed perform effective for large dataset. But it gives as less effective output for less dataset as we are using bilinear function. We will try remove bilinear function in the encryption and key generation function to increase the effectiveness of our system.

Use of IoT devices and cloud is likely to be more popular and possibly resulting in other trends such as Cloud of Battlefield Things and Cloud of Military Things. So, it is important for any organization, public or private, seeking to deploy IoT with Cloud and related architecture to be assured of the security of the system and privacy outsourced in the cloud.

In this paper, we sought to contribute to one of many Cloud of Things security and privacy challenges. Specifically, we defined a searchable encryption protocol, its security model, and security requirements. We then proved the security of the protocol, as well as demonstrating the utility of the protocol in comparison to four other related protocols in the literature.

References

[1] J.Baek, R.Safiavi-Naini, and W.Susilo, “Public key encryption with keyword search revisited,” in Proc. Of the 8th International Conference on Computational Science and Its Applications (ICCSA’08), Perugia, Italy, LNCS, vol. 5072. Springer-Verlag, June–July 2008, pp. 1249–1259.

[2] H. Rhee, J. Park, W. Susulo, and D. Lee, “Trapdoor security in a searchable public-key encryption scheme with a designated tester,” Journal of Systems and Software, vol. 83, no. 5, pp. 766–771, May 2010.

[3] Q. Tang and L. Chen, “Public-key encryption with registered keyword search,” in *Proc. 6th European Public Key Infrastructure Workshop (EuroPKI 2009)*, ser. LNCS 6391. Springer–Verlag, 2009, pp. 163–178.

[4] W.C. Yau , S.H. Heng , B.M. Goi , Off-line keyword guessing attacks on recent public key encryption with keyword search schemes., in: Autonomic and Trusted Computing, International Conference, ATC 2008, Oslo, Norway, June 23–25, 2008, Proceedings, 2008, pp. 100–105 .

[5] I. Jeong, J. Kwon, D. Hong, and D. Lee, “Constructing peks schemes secure against keyword guessing attacks is possible?” Computer Communications, vol. 32, no. 2, pp. 394–396, February 2009.

[6] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, “Dual-server publickey encryption with keyword search for secure cloud storage,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp.789–798, 2016.

[7] L. Wu, B. Chen, K.-K. R. Choo, D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things", *J. Parallel Distrib. Comput.*, vol. 111, pp. 152-161, Jan. 2018.

[8] L. Fang, W. Susilo, C. Ge, and J. Wang, “Public key encryption with keyword search secure against keyword guessing attacks without random oracle,” *Information Sciences*, vol. 238, pp. 221–241, 2013.

[9] Q. Tang , Public key encryption schemes supporting equality test with authorisation of different granularity, IJACT 2 (2012) 304–321 .

[10] S. Ma , Q. Huang , M. Zhang , B. Yang , Efficient public key encryption with equality test supporting flexible authorization, IEEE Trans. Inf. Forensics Secur. 10 (2015) 458–470 .

[11] R. Ostrovsky, “Software protection and simulation on oblivious rams,” Ph.D. dissertation, University of Californiaat Berkeley Computer Science Division, November 1993.

[12] X. Wang, Y. Mu, and R. Chen, “Privacy-preserving data search and sharing protocol for social networks through wireless applications,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 7, 2017.

[13] P. Xu, H. Jin, Q. Wu, and W. Wang, “Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack,” *IEEE Transactions on computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

[14] N. Cao , C. Wang , M. Li , K. Ren , W. Lou , Privacy-preserving multi-keyword ranked search over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2014) 222–233 .

[15] M. Chuah , W. Hu , Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data, in: 2011 31st International Conference on Distributed Computing Systems Workshops, IEEE, 2011, pp. 273–281 .

[16] H. Li , D. Liu , Y.-S. Dai , T.H. Luan , X. Shen , Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage, IEEE Trans. Emerging Topics Comput. 3 (2015) 127–138

[17] Z. Lv , C. Hong , M. Zhang , D. Feng , Expressive and secure searchable encryption in the public key setting (full version), IACR Cryptology ePrint Arch. 2014 (2014) 614 .

[18] P. Golle , J. Staddon , B. Waters , Secure conjunctive keyword search over encrypted data., in: Applied Cryptography and Network Security, Second International Conference, ACNS 2004, Yellow Mountain, China, June 8–11, 2004, Proceedings, 2004, pp. 31–45 .

[19] D. Jin Park , K. Kim , P.J. Lee , Public key encryption with conjunctive field keyword search, in: International Conference on Information Security Appli- cations, 2004, pp. 73–86 .
[20] A. Arriaga , Q. Tang , Trapdoor privacy in asymmetric searchable encryption schemes, IACR Cryptology ePrint Arch. 2013 (2013) 330 .
[21] C. Liu , L. Zhu , M. Wang , Y.A. Tan , Search pattern leakage in searchable encryption: attacks and new construction, Inf. Sci. (Ny) 265 (5) (2014) 176–188 .
[22] W. Sun , S. Yu , W. Lou , Y.T. Hou , H. Li , Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, IEEE Trans. Parallel Distrib. Syst. 27 (2016) 1187–1198 .
[23] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2004, pp. 506–522.
[24] T. Saito, T. Nakanishi, “Designated-Senders Public-Key Searchable Encryption Secure against Keyword Guessing Attacks”, 2017 Fifth International Symposium on Computing Networking
[25] Q. Huang a ,An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks, Information Sciences 403–404 (2017) 1–14
[26] L. Wu, B. Chen, K.-K. R. Choo, D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things", *J. Parallel Distrib. Comput.*, vol. 111, pp. 152-161, Jan. 2018
[27] Mimi M, Debiao H. Khan Muhammad Khurram, Chen Jianhua. Certificateless searchable public key encryption scheme for mobile healthcare system. *Comput Electr Eng*. 2018;**65**:413-424.
[28] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, J. Comput. Secur. 19 (5) (2011) 895–934.
[29] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, M. Steiner, Highlyscalable searchable symmetric encryption with support for boolean queries, in: Advances in Cryptology–CRYPTO 2013, Springer, 2013, pp. 353–373.
[30] G. Di Crescenzo, V. Saraswat, Public key encryption with searchable keywords based on Jacobi symbols, in: International Conference on Cryptology in India, Springer, 2007, pp. 282–296.

[31] R. Chen, Y. Mu, G. Yang, F. Guo, X. Wang, Dual-server public-key encryption with keyword search for secure cloud storage, IEEE Trans. Inf. Forensics Secur. 11 (4) (2016) 789–798.

[32] D.X. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: 2000 IEEE Symposium on Security and Privacy, 2000. S&P 2000. Proceedings, IEEE, 2000, pp. 44–55.

[33] Wu T-Y , Meng F , Chen C-M , Liu S , Pan J-S . On the security of a certificateless searchable public key encryption scheme. In: International conference on genetic and evolutionary computing. Springer; 2016. p. 113–19 .

[34] Baek J , Safavi-Naini R , Susilo W . Public key encryption with keyword search revisited. In: International conference on computational science and its applications. Springer; 2008. p. 1249–59 .

[35] Peng Y , Cui J , Ying Z . Certificateless public key encryption with keyword search. China Commun 2014;11(11):100–13 .

[36] H.S. Rhee , W. Susilo , H.J. Kim , Secure searchable public key encryption scheme against keyword guessing attacks, Ieice Electron. Express 6 (5) (2009) 237–243