Design and Analysis of Schemes for Privacy Preserving Cloud Storage Services

by

PAYAL CHAUDHARI
201121014

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
to

Dhirubhai Ambani Institute of Information and Communication Technology

October, 2018

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Payal Chaudhari

Certificate

This is to certify that the thesis work entitled DESIGN AND ANALYSIS OF SCHEMES FOR PRIVACY PRESERVING CLOUD STORAGE SERVICES has been carried out by PAYAL CHAUDHARI for the degree of Doctor of Philosophy at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Prof. Manik Lal Das
Thesis Supervisor

Acknowledgments

The journey of Ph.D. required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of this dissertation. First of all, I would like to thank the Almighty God for providing me an opportunity and strength to reach the destination and to have an adviser who has always been there to motivate me. I owe my deepest gratitude to my adviser and mentor, Dr. Manik Lal Das, who guided me in my current research area and gave me valuable advice. In spite of his busy schedule and responsibilities, he was always there to support me. Without his continuous optimism concerning, encouragement and support this study would hardly have been completed. His attitude to achieving excellence has helped me to achieve optimal results and my professional growth as well.

I like to thank the whole team of DA-IICT who directly or indirectly supported me throughout this journey. Especially, I like to thank Prof. Anish Mathuria, Prof. Saurish Dasgupta, Prof. Suman Mitra and Prof. Gagan Garg who provided their valuable suggestions to strengthen my research work.

Next, I would like to thank my parents who show me the dream to achieve the Degree of Doctorate and encourage me to gain that goal. I would also like to thank from bottom of my heart to my in-laws who always supported me to fulfill the dream of my parents.

I am especially indebted to my husband, Ketan for his support and sacrifice. I could not have reached this far without his support and encouragement. Despite being a person not involved in the computer science, he examined my work, helped me to correct the errors and most importantly built my confidence.

During this stressful journey of the dissertation, the person who made me smile was my son Shrey. His best wishes have always been a backbone to me whenever facing any challenge in my life. I always feel proud to have such a nice and wise son.

I would not forget to remember my brother Krunal and Sisters Priyanka and Kavita who supported and encouraged me to do the hard work and to give my best.

One more person to whom I want to express my sincere gratitude is Dr. Sarita Agarwal, who helped me a lot. At the same time, I also like to thank our Security Research Group at DAIICT especially Hardik and Nidhi for always standing beside me and helping me in different aspects.

Being an academician it is equally important the role of your current organization. I am also thankful to the management, Principal and staff members of LDRP Institute of Technology and Research for granting my leaves and supporting me in many other ways to accomplish my dissertation.

Contents

Abstract

Public cloud storage services have become the leading choice for individuals and organizations to store their data, as the service provides the benefits of availability and reliability together at a reduced cost. While storing data in public cloud storage server, preserving data security and privacy have become a prime concern. For providing data security on public cloud storage it is required to preserve the confidentiality of data and to enforce the data access policies. Before uploading the data to public cloud storage, data can be encrypted and assured that only authorized users access the data with a valid decryption key. The enforcement of fine-grained access control policies on encrypted data prevents the unauthorized disclosure of sensitive data among multiple users. On one hand the fine-grained access control policy helps to achieve the authorized access control on data, while on other hand, the access policy discloses the target recipient of the ciphertext. The receiver information for a ciphertext helps an adversary to gain the information about the underlying data. Therefore, it is essential to hide the receiver information for preserving the data security.

Attribute Based Encryption (ABE) is a well-known cryptographic primitive that provides both the confidentiality and fine-grained access control together. In ABE, each data user is identified with a set of attribute values. Each data file has an access policy defined by its owner in terms of the attributes. A user can decrypt the document, only if the threshold number of attributes are matched between the access policy and user's attribute list. The customized version of ABE which hides the access policy within ciphertext is known as Anonymous Attribute Based Encryption (AABE). We have worked on AABE for designing and analyzing some schemes for achieving users' anonymity in retrieving results from stored data in

public cloud storage.

The other research challenges for preserving public cloud data storage security include searching over encrypted data, authentication of data, secure data sharing etc. We have formulated five new AABE schemes which enhances the data storage security in public cloud. The common objective of all our proposed schemes is to maintain data confidentiality and preserve the receive anonymity.

The proposed first three schemes facilitate searching over ABE data with hidden access policy. The first scheme "Data Owner based Searchable Encryption (DOSE)" provides searching using data owner's identity. The scheme enables a cloud server to perform the search operation with look-up approach and do not require any mathematical operations on cloud server side. The other two schemes provide keyword based search over attribute based encrypted data with hidden access policy. One of those schemes is "Receiver Anonymous Searchable Encryption (RASE)", which provides an efficient keyword based searching over attribute based encrypted data with receiver anonymity. It enables a user to selectively retrieve a subset of data from the vast amount of encrypted data stored on the cloud. The search operation of RASE scheme is performance efficient when compared with the existing schemes because for any ABE schemes, the number of pairing operations has a high impact on the operational time complexity. Irrespective of the number of attributes, the search operation in RASE requires a limited and constant number of pairing operations. The RASE scheme is built using the multi-linear pairing. The security of RASE has been proved secure against chosen keyword attack.

The RASE scheme is applicable in a scenario, where the data owner has to include only one value per attribute in the access policy. The scheme does not allow the data owner to place multiple values per attribute to be included in the access policy. In our next scheme for searchable ABE, we have addressed this issue and proposed a "Privacy preserving Searchable Encryption (PSE)" scheme, that enables the data owner to place multiple values of an attribute in the access policy. For the PSE scheme we have also customized the system model and make it secure against the file injection attacks. The PSE scheme is also proven secure against chosen keyword attack.

The searchable encryption schemes facilitate to retrieve the subset from encrypted document collection. However, after retrieval it is required that the user should be able to decrypt the retrieved documents with minimum computation overhead and verify the authenticity of the data. With this motive, we have proposed "Privacy preserving Attribute based Signcryption (PASC)" scheme. Th PASC scheme achieves the cost-efficient decryption operation when compared with that of existing AABE schemes. In addition to cost-effective decryption operation, it also allows the verification of data owner's attributes and unique identity. Unlike the existing ABSC schemes, the scheme supports data owner traceability with sender privacy. The sender privacy is referred to the point that only an authorized receiver is able to identify the data owner who has signed and uploaded the document. The PASC scheme supports data confidentiality, receiver anonymity, message authentication and fine-grained access control altogether. The scheme has been proven secure against adaptively chosen ciphertext attack.

The cost-effective unsigncryption operation makes the scheme PASC a better choice for users who wants to download the searched documents and decrypt them. However, there are certain real-life cases where a user instead of downloading and decrypting the documents, wants to forward them to other user for the purpose of sharing the data. To address this requirement of sharing the encrypted data, we have designed a scheme that is an "Proxy ReEncryption for Anonymous Attribute Based Encrypted data (PRE-AABE)". In traditional proxy reencryption scheme a semi-trusted proxy such as the cloud server converts a data encrypted for Alice into the data for Bob without learning the plaintext contents. Our proposed scheme on attribute based proxy reencryption allows the alteration of ciphertext access policy which is hidden inside the ciphertext. The reencryption task in our scheme does not allow the cloud server to learn about the access policy or the plaintext contents concealed in the ciphertext. The scheme imposes minimal decryption overhead on user side. The scheme also facilitates a delegator to

put the reencryption control on the ciphertext, so that the further sharing of data can be controlled. The scheme has been proven secure against chosen plaintext attack. We have experimented the proposed schemes using the pbc cryptography library. The experimental setup for the proposed schemes on end user side used Intel - i5 processor with 3 GB RAM and the cloud side operations were run on a Google compute engine. We have shown the performance analysis of all proposed schemes and compared the results with related schemes.

List of Abbreviations

AABE	Anonymous Attribute Based Encryption
ABE	Attribute Based Encryption
ABKS	Attribute Based Keyword Search
ABPRE	Attribute Based Proxy Reencryption
ABSC	Attribute Based Signcryption
AC	Attribute Center
CP-ABE	Ciphertext Policy Attribute Based Encryption
CP-ABPRE	Ciphertext Policy Attribute Based Proxy Reencryption
CS	Cloud Server
IDSC	Identity Based Signcryption
KP-ABE	Key Policy Attribute Based Encryption
LSSS	Linear Secret Sharing Scheme
MSK	Master Secret Key
PK	Public Key
TG	Token Generator

List of Tables

List of Figures

CHAPTER 1

Introduction

Cloud computing is a fast evolving computational paradigm, where a large number of systems are coupled with private or public networks, to offer infrastructure services for application development, deployment and data storage on pay-as-per-use basis. The cloud computing technology provides the benefits of scalability, elasticity, infrastructure cost reduction and ease of access to data that has been stored in the cloud [1]. One of the most useful cloud services is the cloud storage service. The cloud storage service facilitates the users to store data, such as files, movies, videos, photographs and other important documents on remote side servers and allows them to access their data from any operating system and from anywhere, at any time. The cloud storage helps the user to reduce the large expenditure on storage infrastructure. The costly data storage essentials on the client side are now replaced with remote side storage services offered by the cloud service providers on pay-as-per use basis.The public cloud storage services are built using the cloud storage technology, in which the service providers use the Internet to provide storage resources and services to the users. A user of a public cloud storage service goes online and access his public cloud storage account, then upload or retrieve the data. The abstract level view of cloud service usage is shown in figure 1.1. A user is also able to share his data with other users through the medium of public cloud storage services. Thus cloud storage service is a solution to individuals and organizations anticipating to reduce the infrastructure costs.

With the extensive use of public cloud storage services, the security risks become prevalent. The advantages of on-demand high quality data storage services at a reduced hardware cost comes at the cost of security concerns. Because the

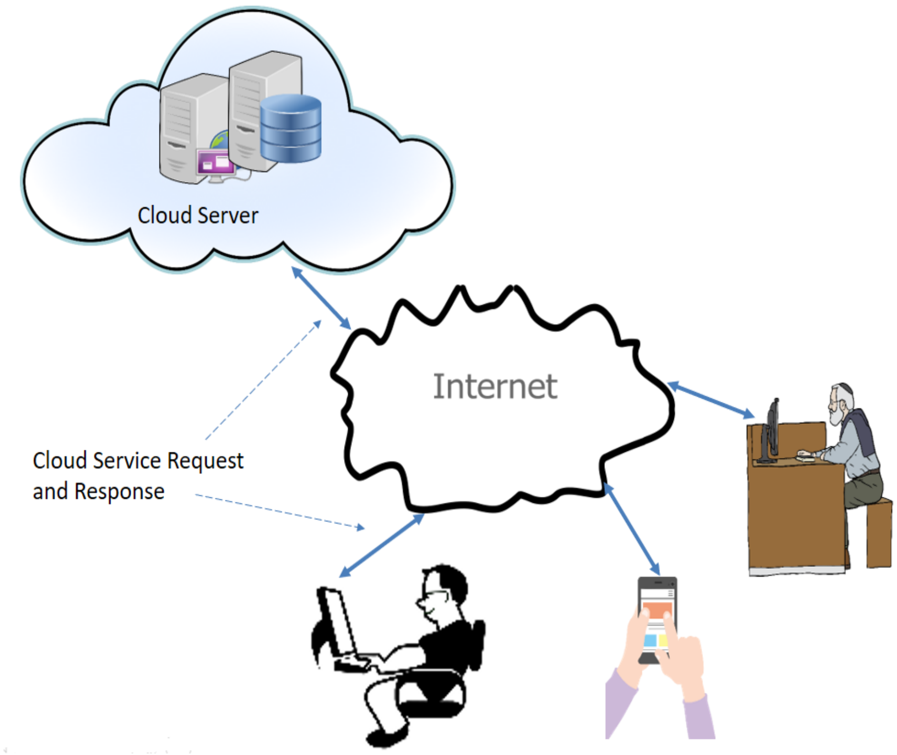


Figure 1.1: Cloud Service Model

public cloud storage service provider, it is still an outsider for the user and out of the user's premises. Therefore, the user can not fully trust on public cloud storage service providers. For example, in [2], an article listing the 17 major security breaches found in well-known cloud storage systems have been listed. In [3], Barron *et al.* have presented a detailed analysis of real time cases where the cloud storage services have been compromised. When outsourcing the confidential data such as financial records, health records, legislative documents etc. on cloud storage, the user always concerns about the security and privacy of his data. Data security is generally defined as the confidentiality, integrity, and availability of data. Put it in other way, it is about all the services and mechanisms which are used to ensure that the data is not accessible to any unauthorized entity. It ensures that the data is perfect and trustworthy and guarantees that the data is available when any authorized person needs its access.

Data privacy is commonly defined as the correct use of data. When users outsource their data to the cloud for storage or computation purpose, then that data should be used only for the agreed objective. The data must not be sold, released or leased to any other parties for commercial reasons without concern of the user or organization. In order to provide such features to service consumers in cloud setup, security and privacy concerns need to be addressed per application requirements, which we discuss in next section.

1.1 Motivation

The research challenges for securing public cloud storage services are :

Data Confidentiality: It is crucial that before adopting cloud storage services, we preserve the data confidentiality while storing it in cloud storage.To reduce the risk of storing the confidential data on public cloud storage, it is a well-known practice for data owners to encrypt the data prior to outsourcing it. For obtaining data confidentiality, the data is encrypted by the data owner with a secret key. The encryption of data must achieve semantic security of data, that is given a ciphertext of a message m and its length, no polynomial time adversary is able to determine any partial information regarding m with a non-negligible advantage $\epsilon(1^l)$ where l is the size of security parameter used for generating the secret key. In order to fulfill this purpose, the cloud server should not learn any meaningful information from the encrypted data that has been stored in it.

Enforcement of access policy: Other than encryption of data, enforcement of access policies is a necessity to prevent the unauthorized disclosure of confidential data. It is highly required that the unauthorized disclosure of sensitive information, such as health records, stored on public cloud servers must be prevented to preserve the data privacy. The task of enforcing access control policy becomes more challenging when the storage is accessed by a number of users and the data has to be shared with a subset of users.

Furthermore, revelation of confidential data needs fine-grained access control [4] to be placed. The fine-grained access control mechanism requires that each user has different access rights for different types of data. Previously, the techniques such as ACL-based [5], Capability-based [6] or Role-based access control [7] were used by the data storage servers to enforce the access control policies. Enforcing the access policies like ACL-based or Capability-based with cryptographic techniques can be one of the solution to provide the access control services on untrusted storage servers. However, these approaches do not perform well in the cloud scenario because both the ACL-based and Capability based approaches suffer from the scalability issue [5]. The ACL-based access control policy requires every data item to maintain the list of authorized users. If ACL technique is mixed with any cryptographic technique, then the ciphertext size and encryption operation complexity increases linearly with the number of users in the system. The same problem arises with the Capability-based access control mechanism. In Capability-based access control system, for every user a capability matrix is prepared. Each entry in the matrix is considered as a pair (o, s), where o is the name of an object and s is a set of access rights.

In role-based access control [7], a user is assigned the access rights based on his role. In this method, the data items do not require to maintain the list of the authorized users. The access permission are attached with a role. Each user is assigned an appropriate role. This method also has a drawback that it has the vulnerability of various attacks, such as user collusion, where malicious users with different roles try to put their decryption keys together and access the data for which they are unauthorized.

Several other existing schemes [8, 9, 10, 11] have addressed the problem of enforcing data access control with symmetric-key or public-key cryptography These schemes work effectively for conventional file systems, but they are less effective when there is need to apply the fine-grained data access policy in large-scale data storage servers such as cloud server.

Attribute-based encryption (ABE) [12, 13] is a well-known public key cryptographic primitive that enforces the fine-grained access policies within the encryption procedure to achieve the confidentiality and access control together. ABE is able to impose the fine-grained access policies in a large-scale system. In ABE the user identification and data access policies both are defined in terms of attributes. Unlike the existing broadcast encryption schemes [14, 15, 16] where the sender has to select the subset of users to whom the data should be sent, ABE facilitates the encryption of data without exact knowledge of the receiver set, because in ABE the data owner has to identify only the characteristics of receiver in terms of attributes. Therefore, in a cloud storage system, where there are multiple data owners and multiple data receivers, ABE is a better fit to provide both the confidentiality and fine-grained access control.

Access Control with Receiver Anonymity: To protect the data against unauthorized access, it is necessary to provide fine-grained access control to the data in the cloud storage [17, 18, 19]. Because there are multiple data items hosted by the cloud storage and multiple users are accessing the storage. The fine-grained access control allows each data item to have its own access control policy. Each data item uploaded on the storage has different access rights decided by the policy enforcer. The user whose credentials satisfy the access policy is able to retrieve the data item. Many ABE schemes [12, 13, 20, 21, 22, 23, 24, 25, 26] have presented the cryptographic schemes to enforce the fine-grained access control on encrypted data. In such schemes, the access control policy for a data item is attached in clear form with the ciphertext of data. The access policy reveals the required attributes for decrypting the ciphertext, which in turn reveals the receiver's identity. It has been observed that if access policy of a data is clearly mentioned along with the encrypted data, then the adversary is able to learn the information related to the encrypted data such as the intended receiver of ciphertext, purpose of ciphertext etc. Therefore, to preserve the semantic security of data, the access policy should be enforced but at the same time the access policy should be kept hidden from the adversaries. As the public cloud servers are outside of user's premises and cannot be trustworthy, it is necessary to reveal as little user information as possible to storage servers to preserve the data confidentiality. In particular, the data owner would like to hide her access policy information inside the ciphertext and the users who are querying the cloud server to get access to the data desires to conceal their access privilege information from cloud servers. Here, we have used word “Receiver Anonymity” to define that even if a valid receiver is able to access the encrypted document, he is not able to learn the access policy and hence not able to learn who else other then him are the valid recipients of the same ciphertext.

Searching over encrypted data with keyword secrecy and receiver anonymity : To preserve the data confidentiality, no information should be revealed to the cloud server about the actual data of user that has been stored on the cloud server. The provision to resolve these privacy concerns is to apply end-to-end encryption on user’s data on user’s machine before uploading it in the cloud. However, these schemes put constraints on cloud services available to users, such as efficient data searching and retrieval from the cloud storage. There can be vast amount of data stored on the public cloud storage and the user is interested to retrieve a subset of data from it. For example, a user wants to retrieve the documents which contain the word “computer”. As the data is encrypted, there must be some cryptographic mechanism which enables the cloud server to search over the encrypted data without decrypting the data and without learning the keyword for which the search process is extended. This mechanism is known as searchable encryption [27]. In a very abstract level, a searchable encryption scheme provides an encrypted index, to hide its contents from the storage server who performs the search operation. To conduct the search operation a user needs to give the valid search token which can be matched against the encrypted index contents. More precisely, when using a searchable encryption scheme, the index is encrypted in such a way that, when given a token for a keyword, the search operation reveals only the identifiers of the encrypted files that contain the keyword; and without a valid token the index contents remain hidden. It is also required that, only a user with valid secret key is able to generate the search token and from the search procedure the adversary learns nothing about the encrypted data or index except that the result of search operation. There are many types of searchable encryption schemes [28, 29], applicable to particular application scenarios. For example, in a setting where a client encrypts the data using his secret key and uploads it on the cloud storage for his personal use, he opts for searchable symmetric encryption [28]. In this case, both the index and search token are generated from user's secret key. In a scenario where Alice encrypts the data using Bob's public key and uploads it on the cloud storage, so that Bob can access the data using his private key, the searchable asymmetric encryption is suitable [29]. In this scenario, the entries in index are computed from the public key of data receiver and the search token is generated using data receiver's private key.

Generally, the cloud storage system hosts a large number of documents, uploaded by multiple owners and accessed by multiple users [30, 31]. Furthermore, each data owner may decide his own fine-grained search access policy that allows only the authorized users to search over the encrypted data. In such multi-owner multi-user scenario, it is more challenging to facilitate searchable encryption with fine-grained access control policy. In [32, 28], symmetric searchable encryption schemes for multi-user setting have been presented. In these schemes, the data owner issues the shared secret key or search token to other users for the purposed of data sharing as shown in figure 1.2. These multi-user symmetric searchable

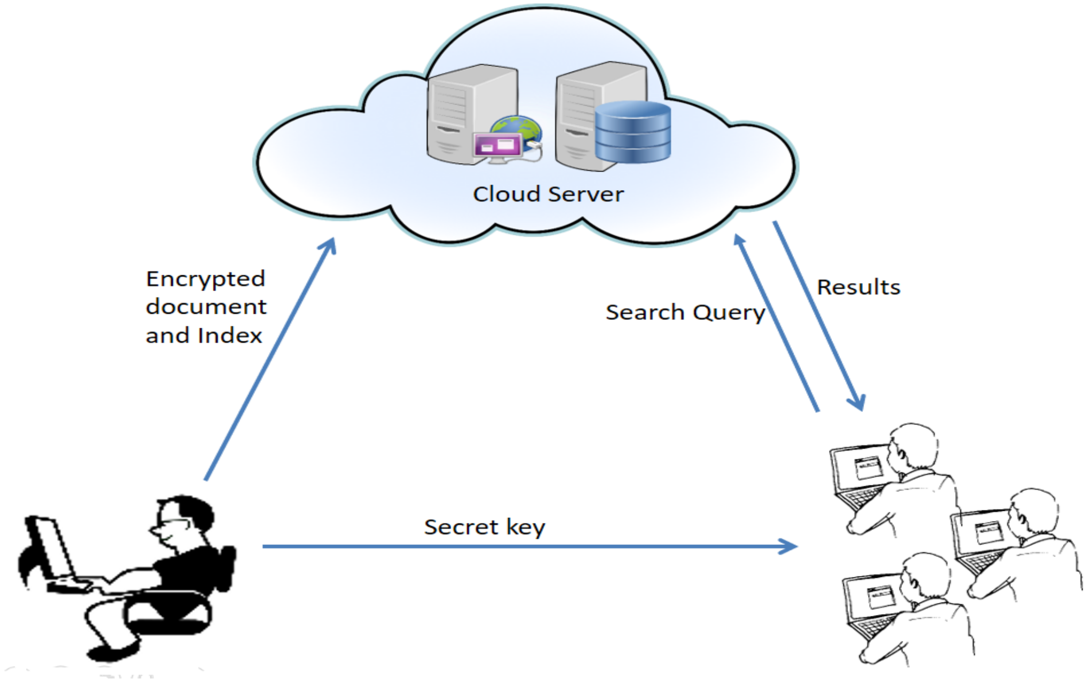


Figure 1.2: The System Model for Multi-user Searchable Symmetric Encryption.

encryption schemes [32, 28] are not preferable in the multi-owners multi-user scenario, because of the increased number of secret keys in the system. Another approach for providing authorized search operation in multi-user scenario is to maintain an authorized user-list on server side [5, 7, 6]. But this approach can be efficient for single-owner-multi-user scenario and cannot perform well in multi-owner-multi-user scenario, because in the later scenario it fails to apply the varying fine-grained owner-enforced access control policy. As there are multiple files contributed by multiple data owners and each data owner has its own access policy, maintaining per file, per data-owner list of users on server side is a bottleneck issue for the server.

The better approach to provide the fine-grained owner-enforced search authorization is of projecting the search facility over ciphertext policy attribute based encryption (CP-ABE) technique [33]. The searchable ABE schemes require a data-owner to encrypt the index of his file with his own access policy. This access policy defines which users can search this index. As shown in figure 1.3, The data user generates the trapdoor using his secret key without communicating with the data owner. The search operation is carried out by the cloud server with the input of trapdoor and index. The task of providing search facility over attribute based encryption becomes more challenging when it is required to preserve the receiver anonymity. The searching operation with fine-grained access control and receiver anonymity requires to facilitate the search operation over attribute based encrypted data without disclosing the access policy or revealing user's attributes.

Data Authentication with sender privacy : Data authentication, while retrieving data from cloud server, is equally important than that of data confidentiality. When accessing the data from a remote side third party storage server a user needs to be sure about the authenticity of data. The purpose of data authentication is to prevent an adversary from doing unauthorized changes in the data. The authenticity of data ensures the receiver that the data has come from a source which has been claimed and has not been changed. As the cloud storage is used by multiple data owners, the data user has to verify the identity of data owner

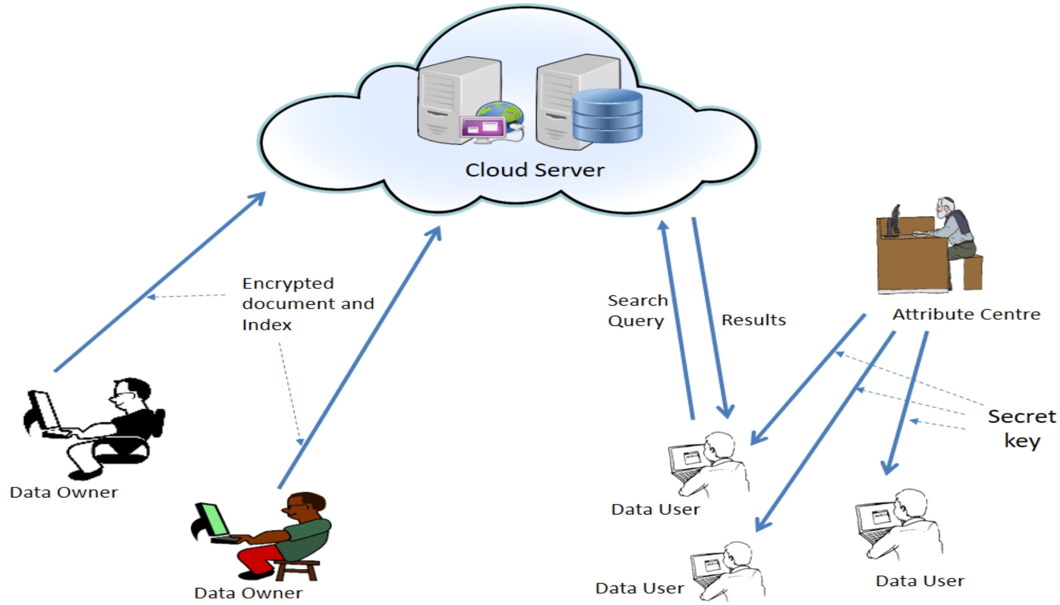


Figure 1.3: The System Model for Searchable CP-ABE suitable for Multi-owner Multi-user Scenario.

who actually sends a message, as well as about the message integrity. For example, when a doctor gets the medical report of a patient from the laboratory staff, then he wants to be sure about the authenticity of the message. Classical crypto-primitive to provide the authentication is the digital signature [34]. In public-key encryption, to generate the digital signature for a message, the data owner signs on message with his private key. The validity of signature is verified on receiver side with data owner's public key.

For accessing both the confidentiality and authentication, the encryption and digital signature both should be applied on message. There are two approaches for obtaining both the security properties. First, encrypt the message and then generate digital signature on it. The other approach is to generate the digital signature and then encrypt it. The first approach fulfills the requirements of confidentiality and authentication, but it does not preserve the sender privacy. The sender privacy refers that other than valid recipients, nobody else should be able to know the identity of sender. When expecting authentication in the cloud storage scenario, it is required that other than a valid recipient, any user should not be able to learn the sender identity. To achieve, this the second approach of digitally sign the data and then encrypt is appropriate. The signature followed by encryption algorithm, increases the computationally complexity because of two different operations. An alternative and better cryptographic technique to provide both the authentication and confidentiality is the signcryption technique [35]. Signcryption performs signature and encryption operations simultaneously in one algorithm. The computational cost of signcryption operations is less than the cost of digitally sign and then encrypt the data.

In an Attribute Based Signcryption algorithm (ABSC) [36], the signer's secret key for signature is constructed from his attributes. The data owner acts as the signer who signcrypts the document and uploads it on the cloud storage. The data owner signs the message using his attributes. A receiver after decryption of data verifies the signature. The signature verification is only possible after a successful decryption. The verification operation gives the knowledge of the signer's attributes to the user. However, in large organization there can be multiple entities who have the same set of attribute values. In case of any malfunctioning the receiver user wants to be sure about who actually sign the data. Therefore, ABSC algorithm should provide sender's attributes information as well as sender's unique identity.

Sharing of encrypted data without compromising data confidentiality and receiver anonymity : In the public cloud storage scenario, both searching and sharing of encrypted data have equal importance. The cloud storage is going to be accessed by multiple receivers. We have discussed that searchable ABE facilitates each receiver to select the subset of data for which he is authorized. In the similar way, proxy reencryption provides ease of encrypted data sharing between multiple users. Sharing of encrypted data is required in many real time scenarios. For example, consider a situation where Dr. Alice is on vacation and wants to forward her patients medical reports to Dr. Bob. Here for the purpose of data security, the patient has encrypted his report with Dr. Alice's public key. The requirement arise in this scenario is to transform the ciphertext generated for Alice into the ciphertext for Bob. The prominent solution for this requirement is proxy reencryption. Proxy reencryption is a cryptographic mechanism that enables a third party acting as proxy to transform the ciphertext for one user (here Alice) into ciphertext for other user (Bob). In cloud storage scenario, the cloud server acts as the proxy server to perform the reencryption task. For finishing the reencryption task, the proxy server requires a rekey generated from Alice's private key and Bob's public key. For preserving the data security, it is essential that the entity such as cloud server, acting as proxy should not learn anything from the ciphertext or from the rekey. For the perfect secrecy, as discussed earlier, it is also required to hide the receiver identity. In the example discussed here, the proxy should not learn the identities of Alice or Bob. Because, the identities of Alice or Bob, will provide the adversary an indication about the contents inside the ciphertext.

Attribute Based Proxy reencryption is a reencryption technique with fine-grained access control [37]. The design of an attribute based proxy reencryption algorithm, requires that if the data is encrypted with an access policy T_1, then any user whose attributes can be satisfied with the access policy can generate a rekey using his secret key constructed from his attributes, and the destination access policy T_2. This rekey is used by the cloud server to update the access policy of a ciphertext and accordingly the contents of the ciphertext as per T_2.

The research challenge arises when the attribute based proxy reencryption should be performed without compromising receiver anonymity. This feature requires that the access policy of a ciphertext and user's attribute information should remain hidden. The access policy before reencryption and after reencryption both should be kept private from the cloud server who is performing the reencryption procedure. The task is challenging because without knowing the access policy contents, they should be updated by a third party who's aim is to learn the information from the ciphertext and access policy.

The searchable encryption and proxy reencryption together makes the data sharing process more effective and secure. The searchable encryption enables a cloud server to select the subset of data as per user's query. Then the rekey sent by the user and generated for proxy reencryption directs the cloud to reencrypt that selected data for other user.

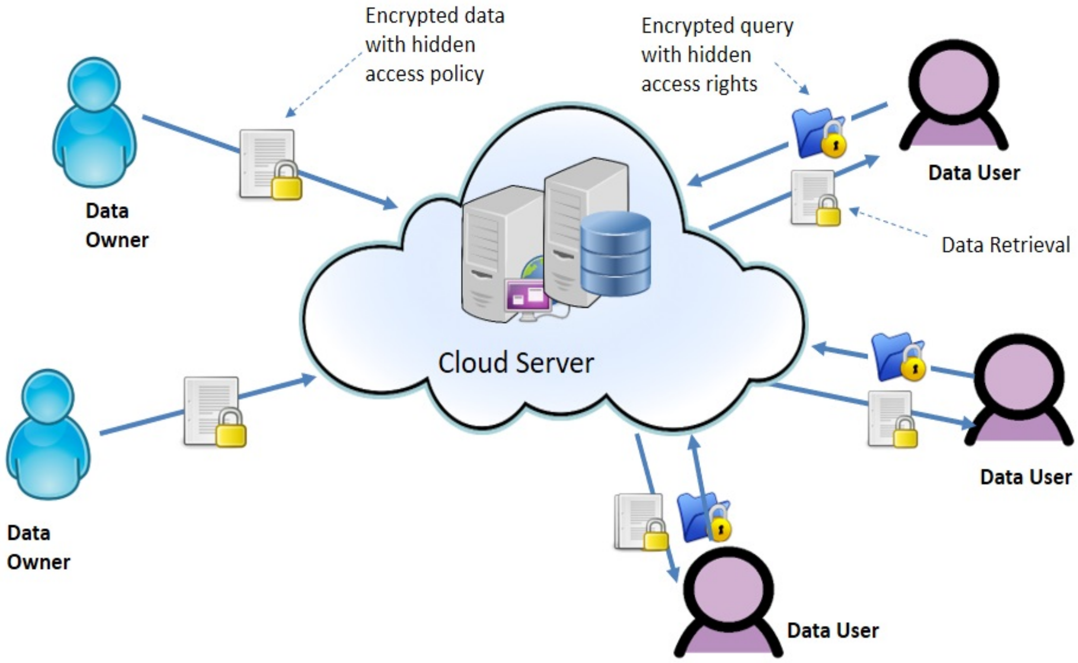


Figure 1.4: Secure Retrieval of data from public cloud storage

1.2 Our Contribution

ABE has promised as an appropriate primitive to provide both the confidentiality and access control together in particular cloud setup. It has also been observed that the access policy information leaks the information related to the ciphertext, which may compromise the data security, as the access policy reveals the receiver's identity. To hide the receiver identity, the access policy must be hidden inside the ciphertext and should not be revealed to anybody. Even a valid receiver can successfully decrypt the message, but can not figure out who else are the recipients of the same message. Considering the receiver anonymity as a primary security requirement, we have designed five new schemes which addresses the functional and security requirements of preserving data privacy and receiver anonymity while using ABE in multi-owner multi-user setting where multiple data owners encrypt and upload the data on cloud storage and multiple data users retrieve the data from public cloud storage.

- We have reviewed the existing literature on searchable encryption techniques.

We have studied the existing attribute based searchable encryption techniques. We observed that not all of them provide the receiver anonymity feature. Only few of them have addressed the issue of receiver anonymity. We have analyzed the security and performance of existing attribute based searchable encryption scheme with hidden access policy and identified their weaknesses in terms of security flaw and performance bottleneck issues. We have proposed a scheme Data Owner based Searchable Encryption (DOSE) that facilitates the search operation using data owner's identity and preserves the secrecy of user's attributes who is making a search operation. The DOSE scheme achieves the receiver anonymity in searchable encryption with fine-grained access control. The search operation in DOSE does not require any mathematical computation on cloud server side. The cloud server performs look-up operation to search for the matched documents. Therefore, the DOSE scheme has the feature of swift searching time. The DOSE scheme does not suitable to provide keyword based searching.

- To provide keyword based searching, we proposed a scheme Receiver Anonymous Searchable Encryption (RASE), that performs the keyword search operation with fine-grained access control with receiver anonymity. The scheme preserves both the data security and receiver anonymity. The construction of scheme is done using the multi-linear pairing operation. The scheme supports access policy in form of AND gate on multi-valued attributes. For each attribute any one value can be placed in the access policy of an encrypted index. The scheme has been proven secure against "Indistinguishability against Ciphertext Policy and Chosen Keyword Attack(IND-CP-CKA)". The mathematical correctness of the scheme has been provided. The scheme is featured with performance efficiency in terms of search operation complexity. The mathematical operations required on CS side to perform the search operation are constant and minimum irrespective to the number of attributes in the access policy. The implementation of search operations on computer system is not done, because the cryptographic libraries providing multi-linear pairing are still in developing stage [38] and therefore, the implementation of RASE is considered as future work.

- The proposed scheme on Privacy preserving Searchable Encryption (PSE) provides keyword based searching over attribute based encrypted data with hidden access policy. Like the RASE scheme it has the access policy in form of AND gate on multi-valued attributes. Unlike the RASE scheme, this scheme is constructed from bilinear pairing and its access policy allows multiple values for an attribute to be included in the access policy. To make the scheme secure against the file-injection attack [39], we have customized the system model of PSE and add a trusted entity known as Token Generator (TG) on data-owner side. The role of TG is to add the master secret key components in the encrypted index generated by data owner before the indexes are uploaded on the cloud storage. The PSE scheme has also been proven secure against the IND-CP-CKA. The security analysis of scheme is provided. We have implemented PSE scheme and tested it on the google cloud computing machine. The search operation time complexity of PSE grows linearly with the total number of attribute values in the system.

- The searchable encryption enables a user to select a subset of data from the cloud storage for which the user is authorized. The user can retrieve these resultant data or forward them to other user. For the former purpose we have designed a signcryption algorithm, that facilitates the user to decrypt the data and, verify the message integrity and sender identity, with minimal computational cost. The proposed scheme - Privacy preserving Attribute Based Signcryption (PASC) provides the feature of performance efficient unsigncryption operation and sender privacy with sender accountability [40]. Unlike the other attribute based signcryption schemes, the PASC scheme has addressed the issue of receiver anonymity. One more important feature of PASC is to allow a receiver to get the knowledge of sender's attributes and the sender's unique identity after a valid decryption operation only. The knowledge of sender's unique identity helps to identify the sender in case of any malicious activity just as sending a fake message. The sender unique identity in addition to his attributes is required information, because in an organization there can be multiple users who possess the same set of attributes. Therefore, to trace the sender, the sender accountability feature is useful. In comparison to existing Anonymous Attribute Based Encryption (AABE) schemes, the PASC scheme provides the signature property and performance efficiency. Thru the performance analysis we have proved that the unsigncryption operation cost of PASC is constant and minimum when compared with the decryption operation of existing AABE schemes. We have proved the PASC scheme is secure against adaptive "Indistinguishability against Ciphertext Policy and Chosen Ciphertext Attack" (IND-CP-CCA) and "Existential Unforgeability against Chosen Plaintext Attack in Adaptive Predicate" (AP-EUF-CPA) model.

- The PASC scheme provides a performance efficient unsigncryption operation and authentication properties. However, the scheme does not support the online encrypted data sharing. To make the online encrypted data accessible to other user, one may take help of proxy reencryption. We have designed a proxy reencryption technique for AABE scheme (PRE-AABE) to support the encrypted data sharing [41]. The PRE-AABE scheme facilitates the encryption of data in such a way, that its reencryption task can be delegated to the cloud server. To preserve the receiver anonymity is a primary motive of PRE-AABE, therefore, we have constructed the reencryption algorithm that does not compromise the receiver anonymity. In our proposed scheme, neither the rekey nor the reencryption operation reveals the user's attribute information. The access policy of a ciphertext before reencryption or after reencryption remains hidden. The task of generating the rekey can be done offline. The decryption operation is also divided in two parts. The costly bilinear pairing operations are performed on cloud server side. The final decryption operation which has computationally negligible cost is performed on receiver side. These features makes the scheme suitable for operating from small handheld battery-driven devices. To control the reencryption chain, we have devised the scheme with reencryption control. A data owner who is uploading the encrypted message or a user who is sending the reencryption key, can set the reencryption control. A ciphertext whose reencryption control is set, can not be reencrypted further by the proxy server. The scheme has been proven secure against "Indistinguishability against Ciphertext Policy and Chosen Plaintext Attack" (IND-CP-CPA) and its performance has been tested on the google cloud computing machine. The performance results of the scheme are presented in the report.

1.3 Thesis Outline

We provide a detailed background in chapter 2. Our proposed schemes use extensively ABE as the basic cryptographic primitive. Therefore, in section 2.1, we have discussed the methodology of ABE. The purpose and logic for describing the access structures are described in section 2.2. Section 2.3 discusses the system model for applying attribute based encryption in cloud storage scenario. The various security models which are used to define the security of our proposed schemes are discussed in 2.4. Computation assumptions used in the security analysis of the proposed schemes are stated in section 2.6.

In Chapter 3, we present the detailed study of searchable encryption techniques. The security notions defined to analyze any searchable encryption scheme are discussed in section 3.3. To facilitate search operation over encrypted document set an encrypted index must be formed. The two approaches to design the encrypted indexes are described in section 3.2. Existing searchable encryption techniques that facilitates authorized searching with the use of fine-grained access control policy are discussed in section 3.4. This discussion is followed by the comparison of existing ABSE schemes in Table 3.1. In the subsequent sections 3.5 and 3.6.1 we provide the analysis carried out for the existing attribute based searchable schemes with hidden access policy. In section 3.5 and 3.6.1 we discuss our proposed schemes to provide anonymous search operation over ABE data with hidden access policy.

In Chapter 4 we present our construction of privacy preserving searchable encryption (PSE). After discussing necessity of the proposed scheme in section 4.1, in section 4.2, we present the PSE scheme. The design goals of the schemes are listed in section 4.2.1, followed by the proposed system model in section 4.2.2. In the system model of PSE we have introduced a new entity known as Token Generator(TG). The roles and responsibility of TG are discussed in section 4.2.3. The section 4.2.4 describes the scheme definition followed by the discussion of security model for PSE in section 4.2.5. In section 4.2.6, we provide the construction for the PSE in detail. The security analysis of proposed searchable encryption scheme is shown in section 4.3. The performance of proposed scheme is presented in section 4.4. Finally we conclude the chapter in section 4.5.

In Chapter 5, we present a privacy preserving signcryption scheme (PASC). In section 5.1, we discuss the existing schemes for ABE with hidden access policy and the schemes providing signcryption. In section 5.1.3, we provide the analysis of Zhang *et al.*'s scheme and show that the scheme suffers from a security flaw. (Our proposed solution to mitigate the security flaw present in Zhang *et al.*'s scheme is included in Appendix A.) In section 5.2.2, we present the method to generate a unique system id for a user from his attributes. In section 5.2, the PASC scheme is presented. In section 5.2.1, the design goals of the scheme are discussed. In section 5.2.3, the scheme definition is presented. The security model for PASC is discussed in section 5.2.4. The detailed construction of PASC is described in section 5.2.5. Section 5.3 presents the security analysis of PASC scheme . The performance evaluation of PASC is shown on section 5.4. In section 5.5, we conclude our work on attribute based signcryption scheme.

Chapter 6 contains the discussion of our work on proxy reencryption. Starting with the detailed analysis of existing attribute based proxy reencryption schemes in 6.1, we have presented our Proxy Reencryption for Anonymous Attribute Based Encrypted Data (PRE-AABE) scheme in section 6.2. In section 6.2.1, the goals defined for PRE are listed. The section 6.2.2 presents the system model for the proposed scheme PRE. In section 6.2.3, we present the scheme definition for PRE. In section 6.2.4, we discuss the security model for PRE followed by its detailed construction in section 6.2.5. In section 6.3, we provide the security analysis of the scheme and in section 6.4 we show the performance analysis of PRE. Section 6.5 presents the conclusion of our work on proxy reencryption.

In chapter 7, we present the conclusion of our all research work and discuss some future scopes of this research work..

CHAPTER 2

Preliminaries

2.1 Attribute Based Encryption (ABE)

Attribute Based Encryption(ABE) is a public key cryptographic primitive that provides confidentiality and fine grained access policy together [42]. Typically in ABE, both the ciphertext and user key are linked with a set of attributes. A user can decrypt the ciphertext if and only if at least threshold number of attributes are matched between the ciphertext and the user's secret key. Unlike the conventional public key cryptography [43], ABE provides one-to-many encryption where one ciphertext is intended for multiple receivers. Initially Sahai and Waters have presented the ABE scheme with threshold access structure [42]. In their scheme, minimum threshold number of attributes should be matched between the attributes inside the ciphertext access policy and attributes of user. However, the threshold structures are not very indicative for designing comprehensive access control policies. An access structure (a.k.a. access policy) is a structured combination of attributes. For example, in a health care organization a patient may decide the access policy of his encrypted report as a "Doctor" working in "Nephrology" or "Urology" department. Later, Goyal *et al.* proposed a key-policy attribute-based encryption (KP-ABE) scheme [12]. In KP-ABE scheme, a ciphertext contains a set of attributes and each user's secret key includes an access policy. A user is able to decrypt the ciphertext, if and only if the attributes in ciphertext satisfy the access policy embedded in the user's key. With the same concept, Goyal *et al.* introduced the variant of ABE in form of Ciphertext Policy Attribute Based Encryption(CP-ABE). In CP-ABE, the ciphertext is associated with an access policy and user's

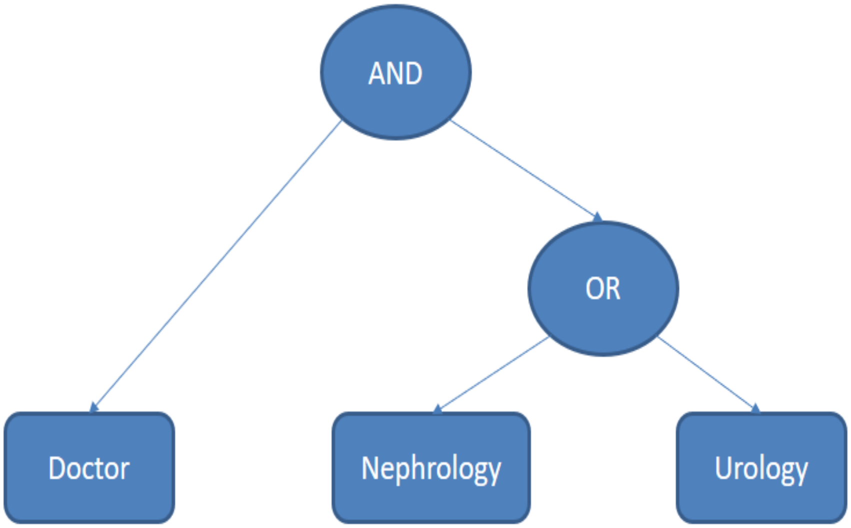


Figure 2.1: Access Policy Structure

attributes are embedded in his key. The formal definition of CP-ABE and KP-ABE are explained later

2.1.1 System Model for ABE schemes

In general, the ABE schemes are applicable in a scenario, where there is a trusted third party in system who is responsible to generate the system parameters and issue the secret keys to each user. This trusted authority is known as Attribute Center(AC) . A data owner encrypt the data using the public key parameters and a data user decrypt the documents using her secret key. In CP-ABE scheme, the decryption is only successful, if the user's attributes in her secret key satisfies the ciphertext access policy. In KP-ABE, the user is able to decrypt the message if and only if the access policy embedded in user's secret key is satisfied with attributes associated with ciphertext. In figure 2.2, the system model for applying the ABE schemes in cloud storage model is shown. As there is no direct interaction required between data owner and data receiver, the ABE schemes gives better performance in multi-owner multi-receiver scenario when compared to other traditional public key cryptographic primitives.

Definition 1 (CP-ABE)**.** *CP-ABE is defined as a tuple (Setup, KeyGen, Encrypt, De-*

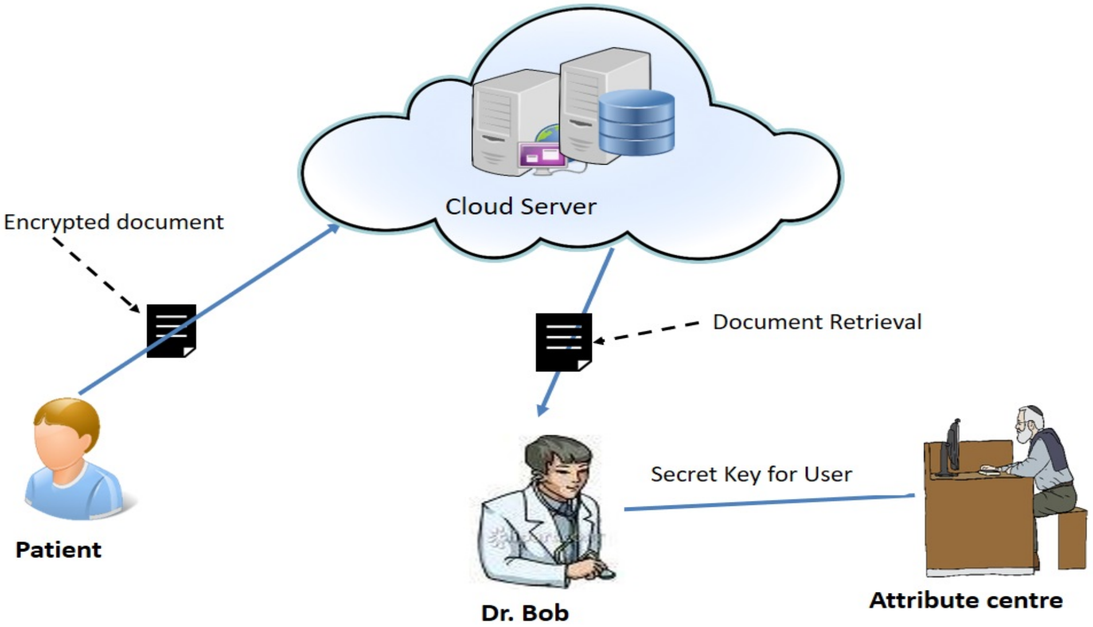


Figure 2.2: The ABE scheme setup in cloud storage model.

crypt) as follows:

Setup(1^{λ}) **→ (***MSK,PK***) :** *This algorithm is run by the attribute center. it takes as security parameter* λ*, and outputs the master secret key MSK and public key parameters PK for the system.*

KeyGen(*MSK,L*)**→ (***SK*$_L$**) :** *The attribute center runs this algorithm for each data user. It takes as input the MSK and the set of user's attribute values. It outputs the user's secret key SK*$_L$*.*

Encrypt(*M,PK,T*)**→ (***CT***) :** *This algorithm is run by the data owner. It takes as input the message M to encrypt, the public key parameters PK and the access policy structure. It outputs ciphertext CT. The encryption algorithm is a probabilistic algorithm and therefore, each time for the same message and same access policy it generates the different ciphertext.*

Decrypt(*CT,SK*$_L$)**→ (***M***) :** *This algorithm is run by the data user. it is a determinis-*

tic algorithm. If user's attributes are able to satisfy the access policy of ciphertext, then the user steps ahead for decryption computation. If the user's attributes satisfy the access structure, then only the user's key is able to decrypt the message correctly.

Definition 2 (KP-ABE)**.** *KP-ABE is defined as a tuple (Setup, KeyGen, Encrypt, Decrypt) as follows:*

***Setup(**1^λ**)** $\rightarrow$ **(**MSK,PK**) :** This algorithm is run by the attribute center. it takes as security parameter λ, and outputs the master secret key MSK and public key parameters PK for the system.*

***KeyGen(**MSK,T**)**$\rightarrow$ **(**SK_T**) :** The attribute center runs this algorithm for each data user. It takes as input the master secret key MSK and the access policy assigned to user denoted as T. It outputs the user's secret key SK_T.*

***Encrypt(**M,PK,L**)**$\rightarrow$ **(**CT**) :** This algorithm is run by the data owner. It takes as input the message M to encrypt, the public key parameters PK and the set of attribute values L. It outputs ciphertext CT. Like in CP-ABE, the encryption algorithm of proposed scheme is also a probabilistic algorithm and therefore, each time for the same message and same set of attribute it generates the different ciphertext.*

***Decrypt(**CT,SK_T**)**$\rightarrow$ **(**M**) :** This algorithm is run by the data user. it is a deterministic algorithm. If user's access policy can be satisfied with attributes associated with ciphertext, then the user performs the decryption computation. If attributes of ciphertext can satisfy the access policy embedded in user's secret key, then the decryption operation is successful.*

There are many variants of ABE schemes [12, 13, 20, 21, 22, 23, 24, 25, 26] in the literature. All the ABE schemes uses bilinear pairing operations in the decryption algorithm [44]. Bilinear pairing operations are costlier than other modular arithmetic operations like modular multiplication and exponentiation. Therefore, the computational cost of decryption algorithm in ABE schemes is very high when compared to other public key cryptography techniques such as RSA. Keeping the decryption algorithm cost reasonable has been a challenge for the designers of (CP/KP)-ABE techniques.

One more security challenge in the use of CP-ABE technique for securing data in cloud storage arise because of the attachment of access policy to ciphertext in plain format.(In case of KP-ABE same problem arise if the list of attributes attached with ciphertext is in clear form). As the cloud storage servers are outside of user premises and trust domain, a data owner desires to reveal as little information as possible from the ciphertext and its access policy. The access policy reveals the attributes of receiver, which in turn helps to guess the contents of ciphertext and thus break the semantic security of ciphertext. The semantic security requires that knowledge of the ciphertext (and length) of some unknown message does not reveal any additional information on the message that can be feasibly extracted. In CP-ABE, the access policy information leaks the receiver identity. To overcome the problem, the research has been carried out for ABE with hidden ciphertext policy. For example, the receiver information of a patient's encrypted medical report indicates the type of disease the patient might be suffering from. This customized version of ABE is denoted as Anonymous Attribute Based Encryption(AABE) [45, 46, 47, 48, 49, 50]. The issue of costlier decryption cost of ABE schemes becomes bottleneck problem for the AABE techniques. Because if the data is encrypted using an AABE technique, then the access policy is hidden inside the ciphertext and each user has to perform the decryption operation on every ciphertext, without knowing whether he is the intended recipient of that ciphertext or not. Therefore, the AABE techniques should be designed, so as to make the decryption operation cost down. This is required especially when the users are accessing the data thru small handheld battery driven devices such as cellphones.

2.2 Access Structure

The ABE is a technique which provides fine-grained access control on the ciphertext. To enforce the fine-grained access control, the access policy T must be defined in a structured way on data owner side in CP-ABE scheme and data user side in KP-ABE schemes. There are various structures used to define the access policy. In this section we discuss the methods of defining the access policy.

2.2.1 Threshold based Access Structure

Suppose that a data owner includes secret shares for n attributes in such a way that d number of attributes from these n attributes are sufficient to derive the master secret key. Therefore, if $L \cap T = d$, then the user is an able to decrypt the ciphertext.

2.2.2 Tree-based Access Structure

Let $\mathcal{T}$ be a tree representing an access structure T. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. If num_x is the number of children of a node x and k_x is its threshold value, then $0 \leq k_x \leq num_x$. When $k_x = 1$, the threshold gate is an **OR** gate and when $k_x = num_x$, it is an **AND** gate. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$.

To facilitate working with the access trees, we define a few functions. We denote the parent of the node x in the tree by **parent(**x**)**. The function **att(**x**)** is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in the tree. The access tree $\mathcal{T}$ also defines an ordering between the children of every node, that is, the children of a node are numbered from 1 to num. Each child of a parent will have unique index number from set [1,num] in an ordered fashion. To assist in traversing the access trees in cryptographic operations, following functions are being used.

- $parent(x)$ = parent of the node x in the tree.

- $att(x)$ = attribute associated with the leaf node x.
- $index(x)$ = index number of node x as a child of its parent node. The value will be between 1 to num.

The encryption algorithm first chooses a polynomial q_x for each node x (including the leaves) in the tree $\mathcal{T}$. The polynomial is chosen in a top-to-bottom fashion, initiating from the root node R. For each node x in the tree, the degree d_x of the polynomial q_x = k_x - 1, that is d_x is one less than the threshold value k_x of that node. For the root node R, the algorithm selects a random $s \in Z_p$ and sets $q_R(0) = s$. Then, it picks d_R number of random points to define the polynomial q_R. For every other node x of $\mathcal{T}$, it computes $q_x(0)$ = $q_{parent}(x)(index(x))$ and selects d_x number of random points randomly to define a polynomial q_x. For each leaf node $\tilde{x}$, $q_{parent}(x)(index(x))$ denotes the value of secret share to be assigned to that attribute $\tilde{x}$.

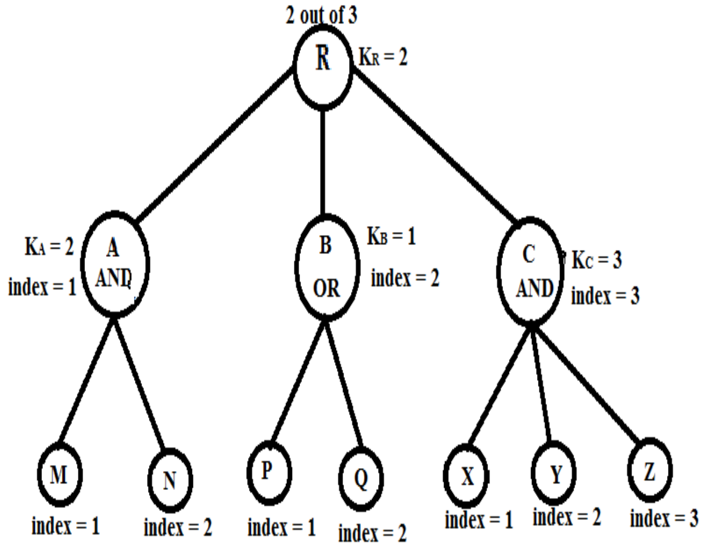


Figure 2.3: Access Tree Construction. {M,N,P,Q,X,Y,Z are the attributes involved in the access policy. The K_x describes the threshold value of a non-leaf node x.}

Satisfying an access tree: Let $\mathcal{T}$ be an access tree with root r. Denote by $\mathcal{T}_x$ the subtree of $\mathcal{T}$ rooted at the node x. Hence $\mathcal{T}$ is the same as $\mathcal{T}_r$. If a set of attributes L satisfies the access tree $\mathcal{T}_x$, we denote it as $\mathcal{T}_x(L)$= 1. We compute $\mathcal{T}_x(L)$ recursively as follows. If x is a non-leaf node, evaluate $\mathcal{T}_{x'}(L)$ for all children x' of node x. $\mathcal{T}_x(L)$ returns 1 if and only if at least k_x children return 1. If x is a leaf node, then $\mathcal{T}_x(L)$ returns 1 if and only if $att(x) \in L$.

2.2.3 Linear Secret Sharing Scheme(LSSS)

Let $\mathcal{P}$ be a set of parties, $\mathcal{M}$ be a matrix of size $l \times m$, and $\rho : \{1, \cdots, l\} \to \mathcal{P}$ be a function mapping a row to a party for labeling. A secret sharing scheme π over a set of parties $\mathcal{P}$ is a linear secret-sharing scheme (LSSS) over $\mathbb{Z}_p$, if

1. The shares for each party form a vector over $\mathbb{Z}_p$.

2. There exists a matrix $\mathcal{M}$ which has l rows and m columns called the share generating matrix for π. For $i = 1, \cdots, l$, the i^{th} row of matrix $\mathcal{M}$ is labeled by a party $\rho(i)$, where ρ: $\{1, \cdots, l\} \to \mathcal{P}$ is a function that maps a row to a party for labeling. Considering that the column vector $v = (\mu, r_2, ..., r_n)$, where $\mu \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \cdots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $\mathcal{M}v$ is the vector of l shares of the secret μ according to π. The share $(\mathcal{M}v)_i$ belongs to a party $\rho(i)$.

It has been noted that every LSSS also enjoys the linear reconstruction property. Suppose that π is an LSSS for access structure A. Let A be an authorized set to decrypt a ciphertext and $I \subseteq \{1, \cdots, l\}$ as $I = \{i|\rho(i) \in A\}$. Then the vector (1, 0, ..., 0) is in the span of rows of matrix $\mathcal{M}$ indexed by I, and there exist constants $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ such that, for any valid shares $\{v_i\}$ of a secret μ according to π, $\sum_{i \in I} w_i v_i = \mu$. These constants $\{w_i\}$ can be found in polynomial time with respect to the size of the share-generating matrix $\mathcal{M}$.

LSSS for ABE: When applying LSSS structure for ABE schemes, the rows of matrix $\mathcal{M}$ are labeled with the attributes in the access policy. Each row of the matrix is labeled with an attribute value that to be included in the access policy. As per the access policy T each attribute in the access policy will get the secret share. A user possessing n number of attributes gets n shares generated from this matrix. Each share belongs to a different attribute value.

2.2.4 AND gate on Positive, Negative and Don't Care Attributes

This is a constrained version of Access Tree Structure representation, which supports only one AND gate between the attributes [20]. Each attribute is assigned a unique index number in the system. Also, each attribute has three possible values : positive, negative and "don't care". Every user in the system possesses either positive or negative value for each attribute. Let there be n attributes in the system and the attributes are indexed as $\mathbb{A}_1$,$\mathbb{A}_2$, $\cdots$,$\mathbb{A}_n$. Now onwards we will just use i to refer to $\mathbb{A}_i$. Each attribute i (for all $1 \leq i \leq n$) is represented as a literal $\underline{i}$ where $\underline{i} = i$ for a positive value, and $\underline{i} = \neg i$ for a negative value. $L = [L_1, L_2, \cdots, L_n]$ is an attribute list for a user, where each L_i represents i if user possesses that attribute else $\neg i$ if user does not possess that attribute. A ciphertext policy is defined as $T = \wedge_{i=1}^{n} T_i$, where each T_i represents either i or $\neg i$ or $*$. $*$ represents the don't care attribute. An attribute list L satisfies an access structure T, if $L_i = T_i$ or $T_i = *$. A binary function $F(L,T)$ gives output 1 if L satisfies an access structure T else outputs 0. Figure 2.4 shows a sample attribute list for an educational institute where all the persons related with the institute are classified as working in Computer Science or other Department, Teaching or Non-teaching, Student or Staff(Non Student). The attributes of a professor working in the computer Science department and the access policy designed for a ciphertext as all the teaching and non-teaching staff working in the computer science department are shown in the figure.

2.2.5 AND gate on multi-valued Attributes

This is a constrained version of Tree based Access Structure because it only includes AND gate and OR gate [47]. But this structure provides more fine-grained representation than that of AND gate on Positive, Negative and Don't Care attributes, because it supports multiple values for an attribute. Let there be n attribute in the universe and each attribute i (for all $1 \leq i \leq n$) has value set $V_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,m_i}\}$. $L = [L_1, L_2, \cdots, L_n]$ is an attribute list, where each L_i represents one value from the value set of attribute i. A ciphertext policy is defined as

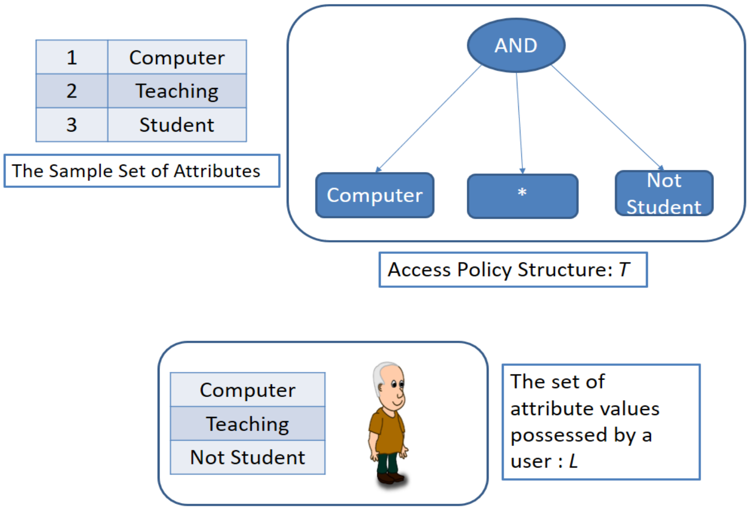

1 Computer
2 Teaching
3 Student
The Sample Set of Attributes
AND
Computer
*
Not Student
Access Policy Structure: T
Computer
Teaching
Not Student
The set of attribute values possessed by a user : L

Figure 2.4: Access Policy Structure with Positive, Negative and Don't care Attributes

$T = [T_1, T_2, \cdots, T_n]$, where each T_i represents the set of permissible values of an attribute i in order to decrypt the ciphertext. An attribute list L satisfies an access structure T, if $L_i \in T_i$ for all $1 \leq i \leq n$. We define a binary function $F(L,T)$ which gives output 1 if L satisfies an access structure T else outputs 0.

Figure 2.5 shows one example of designing the access policy structure for an healthcare organization. As shown in figure - 2.5, there are three attributes, each having the different size of valuesets. The access policy of a ciphertext may contain one or more values for an attribute. A user whose attributes are matched with the access policy is able to decrypt the ciphertext.

2.2.6 Comparison of Access Policy Structures

The threshold access structure is very less expressive and hence it has limited application in ABE schemes. The most expressive access policy structure is in the form of Tree Structure. Because it supports all types of threshold gates. The cryptographic schemes which are using the tree based access structure are proven secure in generic group model. However, the ciphertext size in tree based access structure linearly depends on the number of leaf nodes, and therefor AABE schemes do not prefer the tree based access structure policy, as it reveals the number of attribute values involved in access policy.

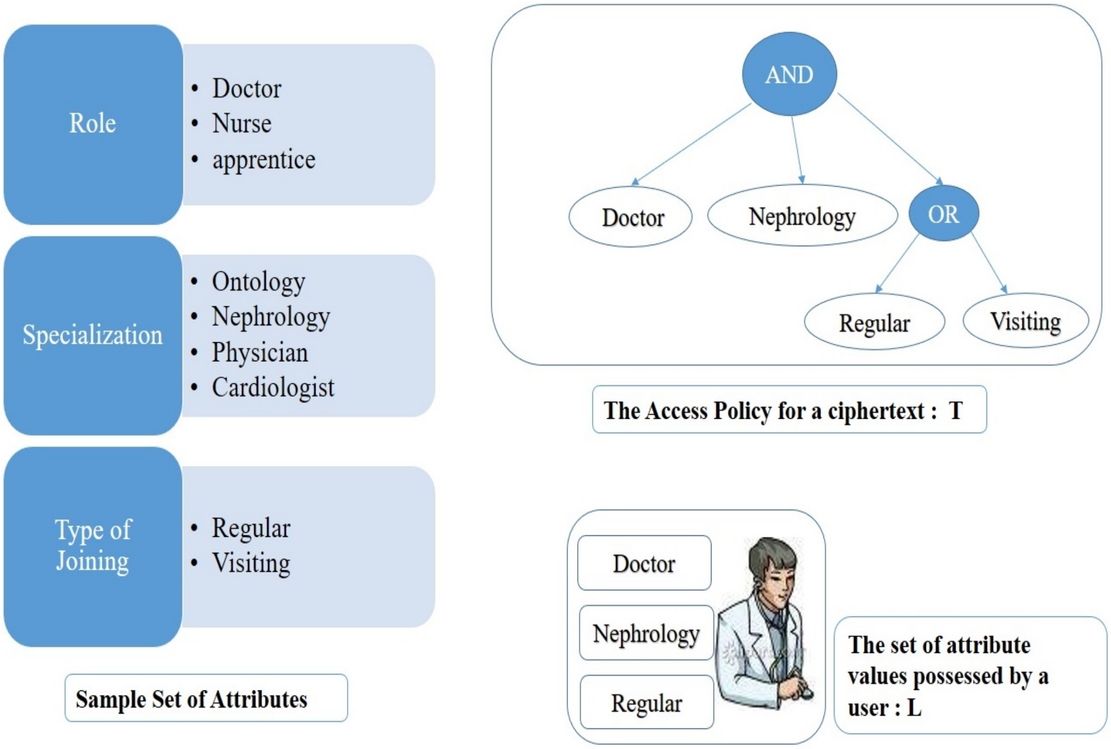


Figure 2.5: Access Policy Structure

LSSS structures are better than tree based structure because unlike the tree based structure, it does not require the recursive operations to reconstruct the secret. When representing the access policy in terms of LSSS matrix, it is required to provide the function ρ which maps each row to an attribute value. The number of rows in LSSS matrix linearly depends on the number of attribute values in the access policy.

Any access policy structure which is represented in form of "AND gate on positive and negative attributes with wild cards", can be represented with the access policy structure "AND gate on Multi-valued attributes" (each negative attribute can be added in the valueset of that attribute). But vice-versa is not true. Many existing AABE schemes have chosen "AND gate on multi-valued attributes" as their access policy. The reasons behind this choice are given below:

- Unlike the tree-based and LSSS-based structure, in this access structure only the AND gate is applied on user's attributes, therefore, no explicit information regarding threshold gates (as in access tree) or the mapping of attribute values to the rows of matrix (as in LSSS) should be given along with the ciphertext.

- The ciphertext size is made constant which is linear to the total number of attribute values in the system irrespective of the number of attribute values included in the access policy. Each ciphertext component represents one attribute value. If the attribute value is included in access policy, then the inclusion of its ciphertext component will reconstruct the secret. Else, it will be a random value and can not be useful in decryption operation.

2.3 System Model

The schemes are being designed for secure and effective data retrieval from public cloud storage. The data is uploaded by data owners and received by data users. There are multiple data owners and multiple data users (receivers). Every person in the system is identified with a set of attribute values. In the system premises, the Attribute Center (AC) setups the system parameters and assigns a secret key to each user in the system as per the attributes of the user.

The data owner uploads encrypted data on the public cloud storage. The uploaded data is encrypted with an access policy defined in terms of receiver attributes. The data user (receiver) sends the search query to the cloud server (CS) and retrieves the resultant documents. To generate the search query and to decrypt the resultant documents, the user makes use of his secret key assigned by the AC. The common goal of our proposed schemes is to preserve the receiver anonymity with the use of hidden access policy while making use of public cloud storage services.

2.4 Security Model

While designing any cryptographic technique, it is required to formalize its security scope. The scope of security for a scheme is defined through a security model. The security models used for analyzing our proposed schemes are defined below.

2.4.1 Chosen Plaintext Attack

This model is used to prove the security of an encryption scheme. It is used to prove that a ciphertext without a valid decryption key does not reveal the plaintext. In this model, the adversary $\mathcal{A}$ is having access to a number of plaintext-ciphertext pairs for his chosen plaintexts and a challenge ciphertext for which he does not possess a valid decryption key. The goal of the adversary is to break the challenge ciphertext and learn the underlying plaintext.

Indistinguishability against Ciphertext policy and Chosen-plaintext attack Model (IND-CP-CPA)

In this model, the $\mathcal{A}$ is allowed to receive a number of secret keys and a challenge ciphertext from the challenger $\mathcal{C}$. The challenge ciphertext has covered both the plaintext and the access policy. The $\mathcal{A}$ makes his efforts to learn the underlying plaintext or the access policy from the ciphertext. This model we have used to define the security of a proxy reencryption scheme. It shows that unless a valid decryption key is available, the $\mathcal{A}$ can not learn the plaintext or the access policy from the challenge ciphertext.

Setup: $\mathcal{A}$ gives l as the security parameter to $\mathcal{C}$. $\mathcal{C}$ runs the setup algorithm and returns the public key PK is sent to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to issue adaptively generated polynomial queries for getting decryption key SK with respect to attribute set L.

Challenge: $\mathcal{A}$ submits two pairs (M_0, T_0) and (M_1, T_1). The input submitted by $\mathcal{A}$ must have to satisfy the below mentioned criteria. If either of them fails, then $\mathcal{C}$ aborts.

1. M_0 and M_1 are of equal length.

2. For any set of attribute values L submitted in queries during Phase - 1, $F(L,T_0) = F(L,T_1)$=0

The challenger $\mathcal{C}$ randomly chooses $b \in \{0,1\}$, then gains the challenge ciphertext CT_b. $\mathcal{C}$ submits CT_b to $\mathcal{A}$.

Phase 2: Same as in Phase 1. $\mathcal{A}$ issues the adaptively generated queries for gaining the secret key SK without violating the restrictions stated during Challenge Phase.
Guess: $\mathcal{A}$ outputs a guess b' of b. The adversary wins the game if $b' = b$. The advantage of $\mathcal{A}$ in this game is defined as Adv$_A$(l)= $|Pr[b' = b] - 1/2|$.

Definition 3. *A scheme is secure in IND-CP-CPA secure, if no polynomially bounded adversary has a non-negligible advantage in the security parameter l with the above game.*

Existential Unforgeability against Chosen Plaintext Attack in Adaptive Predicate (AP-EUF-CPA) model

This model defines the existential unforgeability of a signcryption scheme which provides both the confidentiality and authentication properties. Existential unforgeability ensures that unless a valid signature key is available, the $\mathcal{A}$ will not be able to calculate a valid signature on a message. The AP-EUF-CPA model for a signcryption scheme is defined as follows.
Setup: The $\mathcal{A}$ gives l as the security parameter to the challenger $\mathcal{C}$. $\mathcal{C}$ runs the Setup algorithm and retrieves the master secret key MSK and public parameters PK. The public parameters PK are sent to $\mathcal{A}$.
Query Phase: $\mathcal{A}$ is allowed to issue polynomially bounded number of queries adaptively for following:

- secret keys SK for decryption on attributes L.
- Signature key queries for chosen identities ID.
- Cipher components CT for a pair of encrypted plaintext M with signature identity ID and access policy T.

Forgery: $\mathcal{A}$ outputs (CT^*, M^*, T^*, ID^*) where CT^* is a ciphertext generated from M^*,T^*,ID^* and for which neither $\mathcal{A}$ has got the signature key related to ID^* nor (M^*, T^*,ID^*) = (M,T,ID) for any query generated in Query phase.
$\mathcal{A}$ wins the game if he has correctly generated the ciphertext.

Definition 4. *A scheme is existentially unforgeable against chosen plaintext attack in adaptive predicate model (AP-EUF-CPA), if no polynomially bounded adversary has a non-negligible advantage ϵ in the above game.*

2.4.2 Chosen Ciphertext Attacks

The objective of this model is same as that of Chosen Plaintext Attack model. The objective is to break the challenge ciphertext. But the $\mathcal{A}$ defined in this model is given more power than compared to the previous model. In addition to the pairs of chosen plaintext-ciphertexts, the $\mathcal{A}$ is allowed to submit the ciphertexts of his choice and get them decrypted from the $\mathcal{C}$. In an adaptive chosen ciphertext attack model, the $\mathcal{A}$ is allowed to issue the queries for chosen ciphertext even after he has received a challenge ciphertext from the challenger.

Indistinguishability against ciphertext-policy and adaptively chosen ciphertext attack(IND-sCP-CCA2)

The model is used to prove that without a valid decryption key, $\mathcal{A}$ can not decrypt the ciphertext. It also defines that even if $\mathcal{A}$ gains the valid decryption key, he can not learn the underlying access policy. This later property is essential to prove the security of a receiver-anonymous cryptographic scheme.

Setup: The $\mathcal{C}$ gives l as the security parameter to run the Setup algorithm and retrieves the master secret key MSK and public parameters PK. The public parameters PK are sent to the adversary.

Phase 1: The adversary is allowed to issue polynomially bounded number of queries adaptively for following:

- Decryption keys on attributes L.

- Cipher components for an encrypted message M and access policy T.

- Decryption of CT with respect to attribute set L,

Challenge: $\mathcal{A}$ outputs two messages M_0 and M_1 with respect to the challenge access policy T_0^* and T_1^* on which he wishes to be challenged upon. Here, the restriction is that for any set of attributes L submitted by $\mathcal{A}$ during Phase 1 F(L, T_0^*)= F(L,T_1^*). It is also required that if $\mathcal{A}$ has been issued a secret key for set of attributes L during Phase 1 such that, F(L, T_0^*)= F(L,T_1^*)=1, then M_0 = M_1. $\mathcal{C}$ then randomly chooses $b \in$ {0, 1} and computes CT_b as output of Encryption(PK, M_b, T_b^*). The CT_b is sent to the $\mathcal{A}$.

Phase 2: Same as in Phase 1. $\mathcal{A}$ issues polynomially bounded number of queries with the restriction as specified in Phase 1.

Guess: $\mathcal{A}$ outputs a guess b' of b. $\mathcal{A}$ wins the game if $b' = b$. The advantage of $\mathcal{A}$ in the game is defined as Adv$_{\mathcal{A}}(l)$ = $|Pr[b' = b] - 1/2|$.

Definition 5. *A scheme is secure against indistinguishability against selective ciphertext access policy and adaptively chosen ciphertext (IND-sCP-CCA2) attack, if no polynomially bounded adversary has a non-negligible advantage ϵ in the above game.*

2.4.3 Chosen Keyword Attacks

This model is used to prove the security of a searchable encryption scheme. It is used to prove that a search operation over encrypted data does not compromise the data confidentiality or receiver anonymity. The adversary is allowed to issue the polynomial number of queries for getting the trapdoors which are used to conduct the search operation. In an attribute based keyword search operation, the trapdoor is generated with the input of a keyword and the user credentials. The adversary also gains access to an encrypted index. The security model states that from the available encrypted index and the trapdoor, he can not learn the underlying keyword or the user credentials.

Indistinguishability against Ciphertext-Policy and Chosen Keyword attack (IND-CP-CKA)

Setup: The $\mathcal{A}$ gives l as security parameter to the $\mathcal{C}$. The $\mathcal{C}$ runs the setup algorithm and returns the public key PK is sent to $\mathcal{A}$.

Phase 1:$\mathcal{A}$ is allowed to issue adaptively generated trapdoor queries with input keyword w and set of attribute values L. The $\mathcal{C}$ responds with tw generated with the input w and L.

Challenge: $\mathcal{A}$ submits two pairs (W_0,T_0) and (W_1,T_1). The input submitted by $\mathcal{A}$ must have to satisfy the below mentioned criteria. If either of them fails, then $\mathcal{C}$ aborts.

1. W_0 and W_1 are set of keywords with equal length.

2. The trapdoor tw issued to $\mathcal{A}$ during the Phase 1 satisfy either both the challenge ciphertexts or none of them.

The challenger $\mathcal{C}$ randomly chooses $b \in \{0,1\}$, then computes CT_{W_b} as an Encrypted Index of Keywords. $\mathcal{C}$ submits CT_{W_b} to $\mathcal{A}$.

Phase 2: Same as in Phase 1. $\mathcal{A}$ issues the adaptively generated queries with keyword w and a list of attribute values L which should follow at least one of the following criteria : (i) w should be included in both W_0 and W_1 or should not be included in either of them (ii) F(L, T_0) = F(L, T_1). $\mathcal{A}$ is responded with tw corresponding to (w,L).

Guess: $\mathcal{A}$ outputs a guess b' of b. The adversary wins the game if $b' = b$. The advantage of $\mathcal{A}$ in this game is defined as Adv$_A$(l)= $|Pr[b' = b] - 1/2|$.

Definition 6. *The scheme is secure in IND-CP-CKA secure, if no polynomially bounded adversary has a non-negligible advantage in the security parameter with the game above.*

2.5 Multi-linear Mapping

In 2003, Boneh *et al.* have defined the multi-linear mapping operation as below and found that, the construction of cryptographic multilinear maps will have a remarkable importance in the area of cryptography when compared to bilinear mapping[51].

Definition 7. *Let G_1 and G_{k+1} are cyclic groups and e:$G_1^k \to G_k$ is a mapping for some integers k > 1. We say that e is a k-linear map, if the following conditions are fulfilled :*

- *G_1 and G_k are of same prime order p;*

- *For any a_1, a_2, $\cdots$, $a_k \in G_1$, and $g_{1,1}$, $g_{1,2}$, $g_{1,k} \in G_1$, we have*

  $$e(g_{1,1}^{a_1}, g_{1,2}^{a_2}, \cdots, g_{1,k}^{a_k}) = e(g_{1,1}, g_{1,2}, \cdots, g_{1,k})^{\prod_{i=1}^{k} a_i}$$

- *If g is a generator of group G_1, then $e(g, g, \cdots, g)$ is a generator of G_k.*

If we consider, $k = 2$, then the Definition 7 is a definition for a symmetric bilinear mapping. An alternate form of the above definition for multi-linear map is given below.

Definition 8. *Let there are k cyclic groups (G_1,G_2,$\cdots$, G_k) of order p to define a multilinear map, and the mappings are defined as $e_{i+j} : G_i \times G_j \rightarrow G_{i+j}$ for $i, j \in \{1,\cdots, n\}$ and $i + j \leq k$. The following properties are defined for this k-multi-linear map [51]:*

- *Let $g_i \in G_i$ is a generator of G_i, and $g_j \in G_j$ is a generator of G_j, then $g_{i+j} = e_i(g_i, g_j)$ is a generator of G_{i+j}.*

- $\forall$ *a,b* $\in Z_p$, $e_{i+j}(g_i^a, g_j^b) = e_{i+j}(g_i, g_j)^{ab}$.

- *The computation of e_{i+j} should be done efficiently.*

One of the noticeable applications of a k-linear map is to generate a one-round Diffie-Hellman-like key between $k + 1$ parties. Another applications of multilinear mapping suggested by Boneh et al. includes cost-efficient distinctive signatures and broadcast encryption using short keys to reduce the communication overhead. There have been several other applications designed using multi-linear maps to obtain indistinguishability obfuscation[52, 53].

One of the useful application of pairing based cryptography is Attribute Based Encryption[12, 13]. Most of the well-known constructions of ABE uses bilinear pairing to incorporate policies constructed from arbitrary Boolean equations of the attributes. But the ABE techniques with bilinear pairing fails to include the access policy constructed from random polynomial-size Boolean circuits. One of the challenge field of bilinear mapping is to achieve collusion-resistance property for arbitrary access policies, because of the vulnerability of backtracking attacks[54]. As a countermeasure to this attack, constructing ABE over multilinear maps results in ABE technique suitable for all type of circuits as of now [52, 55]. Later on, Gorbunov *et al.* have presented the construction of ABE for circuits using standard lattice assumptions [56]. However, the problem of a pairing-based implementation still remains unsolved.

The various constructions from multi-linear maps provide useful applications in cryptography only if their mathematical implementation is secure against cryptanalytic attacks. But in reality, the things are not so pretty. In literature, several attacks have been proved against most of the constructions built using multilinear mapping[57, 58]. It yields that the multi-linear pairing operations are useful for designing more efficient application than that can be provided using bilinear pairing. However, the current implementation of multi-linear pairing still needs improvement to make them secure. It should also be considered that there are many but not all applications, which are using the multi-linear maps are prone to attack. There are some tricks and triggers to defend the construction of a scheme against the attacks found on multi-linear maps [59, 60].

The on-going research is heading towards the progress in secure construction of multi-linear pairing, it is expected to output concrete foundations. After analyzing that low-degree multilinear maps are adequate for most of the cryptographic applications, some geometry-based techniques, which were previously ignored by Boneh *et al.*, are revisited to gain some useful results [61]. The study concludes that for any cryptographic mechanisms, the multi-linear mapping can provide essentially the valuable building blocks in the near future.

2.6 Computational Assumptions

An cryptographic scheme is proven secure based on the assumption of hardness of some cryptographic problems. Some of the cryptographic assumptions which forms the basis of the security of the proposed schemes in this report are discussed below.

2.6.1 Discrete Logarithm assumption

Let $a \in \mathbb{Z}_p$ be chosen at random and g be a generator of Group G_1. We say that the Discrete logarithm assumption holds in G_1 if no probabilistic polynomial-time algorithm $\mathcal{P}$ can compute the value of a from the values of g and g^a with non-negligible advantage ϵ_{dl}. The advantage of $\mathcal{P}$ is Pr[$\mathcal{P}$(g,g^a) = a] $= \epsilon_{dl}$.

2.6.2 Decisional Linear (D-Linear) Assumption

Let $z_1, z_2, z_3, z_4, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1 in bilinear group setting. We say that the D-Linear assumption holds in G_1 if no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple $(g, Z_1 = g^{z_1}$, $Z_2 = g^{z_2}$, $Z_3 = g^{z_1 z_3}$, $Z_4 = g^{z_2 z_4}$, $Z = g^{z_3+z_4})$ from the tuple $(g, Z_1 = g^{z_1}$, $Z_2 = g^{z_2}$, $Z_3 = g^{z_1 z_3}$, $Z_4 = g^{z_2 z_4}$, $Z = g^{z})$ with non-negligible advantage ϵ_{dli}. The advantage of $\mathcal{P}$ is Pr[$\mathcal{P}$(Z_1, Z_2, Z_3, Z_4, $g^{z_3+z_4}$)=0] - Pr[$\mathcal{P}$(Z_1, Z_2, Z_3, Z_4, g^z)=0] = ϵ_{dli}.

2.6.3 Computational Bilinear Diffie-Hellman (CBDH) assumption

Let $a, b, c \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1 in bilinear group setting. The computational BDH assumption is that given a tuple ($A = g^a$, $B = g^b$, $C = g^c$), the advantage of computing $e(g, g)^{abc}$ is negligible.

2.6.4 Decisional Bilinear Diffie-Hellman (DBDH) assumption

Let $a, b, c, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1 in bilinear group setting. The decisional BDH assumption is that no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple ($A = g^a$, $B = g^b$, $C = g^c$, $e(g,g)^{abc}$) from the tuple ($A = g^a$, $B = g^b$, $C = g^c$, $e(g,g)^z$) with more than a negligible advantage ϵ_{dbdh}. The advantage of $\mathcal{P}$ is Pr[$\mathcal{P}$(A, B, C, $e(g,g)^{abc}$)=0] - Pr[$\mathcal{P}$(A, B, C, $e(g,g)^z$)=0]=ϵ_{dbdh}.

2.6.5 Decisional k-linear Diffie-Hellman (k-DDH) assumption

Let $a_1, a_2, \cdots, a_k, a_{k+1}, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of G_1 in k-linear group setting. The decisional k-DDH assumption is that no probabilistic polynomial-time algorithm $\mathcal{P}$ can distinguish the tuple (A_1 = g^{a_1}, A_2 = g^{a_2},$\cdots$, A_k = g^{a_k}, A_{k+1} = $g^{a_{k+1}}$, $e(g, \cdots, g)^{z'}$) such that $z' = \prod_{i=1}^{k+1} a_i$, from the tuple ($A_1$ = g^{a_1}, A_2 = g^{a_2},$\cdots$, A_k = g^{a_k}, A_{k+1} = $g^{a_{k+1}}$, $e(g, \cdots, g)^z$) with more than a negligible advantage ϵ_{lddh}. The advantage of $\mathcal{P}$ is Pr[$\mathcal{P}$(A_1, $\cdots$, A_k, A_{k+1}, $e(g,g)^{z'}$)=0] - Pr[$\mathcal{P}$(A_1, $\cdots$, A_k, A_{k+1} $e(g,g)^z$)=0]=ϵ_{lbdh}.

CHAPTER 3

Searchable Encryption with Receiver Anonymity

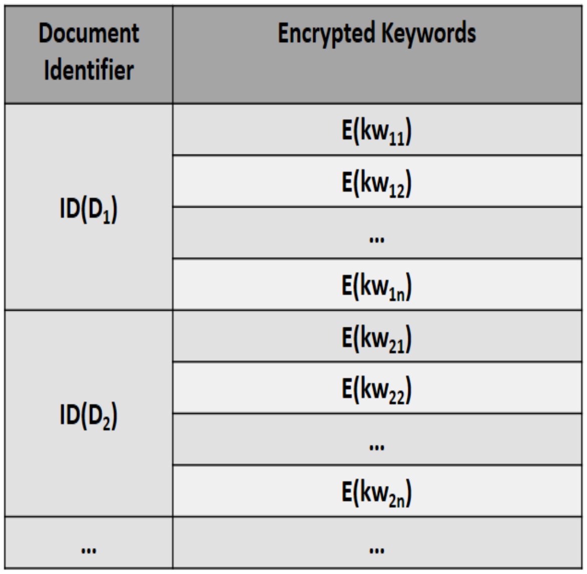
The data stored in cloud storage is directly visible to the cloud server(CS) and could be disclosed to an adversary($\mathcal{A}$) in case of possible collusion between the CS and the $\mathcal{A}$. The obvious solution for mitigating this risk is to encrypt the data before outsourcing it to the cloud. The security challenge arises when the user wants to selectively retrieve a subset of documents from the cloud storage. To satisfy user's need, the cloud should be able to search over the data. Searchable encryption is a cryptographic primitive that enables a third party such as cloud server to search over the encrypted data without learning it. In this chapter, first we study the fundamentals related to Searchable encryption schemes and discusses the existing Searchable encryption schemes for their pros and cons. Then we present our searchable encryption schemes that provide the search functions over attribute based encrypted data with hidden access policy.

3.1 Introduction

As the bulk of data is continuously increasing on public cloud storage, search facility has become one of the most important necessity for effective usage of data. Because it allows a user to work with a selected subset of data from the huge amount of data available in cloud storage. Keyword based search is most preferable type of search in our day-to-day life where the user wants to retrieve the documents containing a specific keyword(s).

To facilitate keyword based searching a keyword index is built from a collection of documents, and the user's query is matched against the index. Index is a data structure that assists the search process. The entry of an index defines a mapping between a keyword and a document identifier, so that during a search process the storage server can find the documents which contain the queried keyword. The query can be identified as single keyword query or multi-keyword query as per the number of keywords in the query. The Multi-keyword queries can be again categorized into conjunctive and disjunctive queries. The result of a conjunctive query includes the documents which contain all the words included in the query. In the result of a disjunctive query all the documents which contain at least one of the queried keyword are included. Other keyword based search approaches include ranking based search and fuzzy keyword search. In ranking based search, the documents matching with the user query are assigned score values calculated by the cloud storage server and from the score result of each document the user retrieves the best matched documents. In fuzzy search approach the similarity distance between the keyword in query and keyword in document is calculated. If the distance is below the predefined threshold value, then that document becomes the part of the result. There are many existing searching schemes available which provide either of the above mentioned features on plaintext data. The challenge arises when the search functionality has to be provided on encrypted data.

Providing search operation over the encrypted data has to accomplish two goals simultaneously. One is the functional goal which requires the correct computation of search results and second is security goal which demands to preserve the confidentiality of data. The cryptographic technique known as searchable encryption is intended to achieve both these goals. Based on the underlying encryption mechanism the existing searchable encryption techniques are further classified as symmetric searchable encryption or asymmetric searchable encryption. In symmetric searchable encryption scheme [28, 62, 63], a user himself encrypts and uploads the data on cloud for his future use. The user generates an encrypted query and fires it on the cloud. The cloud performs the search operation over the user's data and returns him the resultant documents. When a user wants to share the data with other users then he has to share his secret key or generate another se-



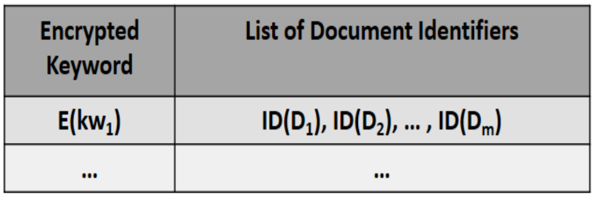
Document Identifier	Encrypted Keywords
ID(D_1)	E(kw_{11})
	E(kw_{12})
	...
	E(kw_{1n})
ID(D_2)	E(kw_{21})
	E(kw_{22})
	...
	E(kw_{2n})
...	...

Figure 3.1: Forward Index Structure

cret key computed from his secret key [28]. In asymmetric searchable encryption schemes [29, 64], a user encrypts the data using the public key of the receiver and uploads it on the cloud. The receiver generates the encrypted query using her private key. In both symmetric or asymmetric searchable encryption, the basic idea is to construct an encrypted index and encrypted query. Both the encrypted index and query should not reveal any statistical information to the storage server who is performing the search operation.

3.2 Index Generation

For enabling a keyword based search operation, it is necessary to create the index. Index is a data structure which lists the keyword-document mapping. There are two varieties in index generation. Forward Index generation where each tuple contains the list of keywords included in each document. The another approach is the Inverted Index. The inverted index contains the entry for each keyword and the list of documents which contain that keyword. The inverted index structure facilitates more efficient search operation then the forward index. The inverted index can give better performance because once identified the required keyword in encrypted form, then all the documents which contain the keyword can be re-



Encrypted Keyword	List of Document Identifiers
E(kw_1)	ID(D_1), ID(D_2), … , ID(D_m)
…	…

Figure 3.2: Inverted Index Structure

trieved in $O(1)$ operation. Forward index operation requires the storage server to search for match in every entry of the index. Thus, if total d number of documents are stored in the storage, then forward index operation requires $O(d)$ search operation. However, the inverted index operation is possible in searchable symmetric encryption, where there is a single data owner and using a single symmetric key, he encrypts all the keywords extracted from the document collection. In a public key searchable encryption algorithm multiple data owners upload their documents on the public cloud storage server. Before uploading, the data owners encrypt their documents and keywords extracted from those documents using the public key of receiver. To achieve the keyword secrecy, the public key searchable encryption algorithm provides probabilistic encryption algorithms. The probabilistic encryption algorithms include random values (chosen by the encryptor) in the output of encryption. Therefore, for the same keyword two different data owners will have different output in encryption algorithm. It yields that, for the public key searchable encryption algorithms, the generation of inverted index is troublesome.

3.3 Security Notions

The security goal of any searchable encryption scheme is not to compromise the data security from encrypted index, user's query and search operation. To measure the security strength of a searchable encryption scheme certain notions are defined as follows [65].

Definition 9. ***History :*** *Let D denotes a document collection. A history of n number of queries over D is defined in form of a tuple H = (D,w) where w = {w_1, w_2, ...,w_n} is a list* *of keywords concealed inside the n queries.*

The contents of history should not be revealed by the adversary. The adversary tries to learn the contents of history from the search and access pattern.

Definition 10. ***Search Pattern :*** *The search pattern defined for a n-query history H = (D,w) is a $n \times n$ symmetric binary matrix τ such that for $1 \leq i, j \leq n$ $\tau[i][j] = 1$, if $w_i = w_j$, and 0 otherwise.*

The search pattern is the information which states that whether any two queries are generated for the same keyword or not.

Definition 11. ***Access Pattern :*** *The access pattern retrieved from an n-query history H = (D,w) is defined as $\phi = (D(w_1), D(w_2), \cdots ,D(w_n))$, where $D(w_i)$ represents the number of documents which contain the keyword w_i.*

It is the information about which documents contain the queried keyword. It is collected from the search result of a query.

Even though the index and the user query are in encrypted form, the cloud server can learn the search pattern and access pattern from the search process. From the result of a search operation, the cloud is able to identify if any two documents contain the same keyword or not. The probabilistic query generation algorithm incorporates randomization in the query and therefore, helps to preserve the search pattern privacy from an outsider, but the cloud server who is performing the search operation can find out whether two queries are for same keyword or not by collecting the tuple from index which matches with the queries. The only technique that helps to preserve the search and access pattern privacy is the "Oblivious RAM" technology. But this technique requires logarithmic number of rounds in search operation. Therefore, to achieve the effective performance of an searchable encryption scheme, the search outcome is the minimum acceptable leakage [28, 66, 39]. In [28], the authors have redefined the security definition of a searchable encryption scheme and state that a searchable encryption scheme is secure, if from the search operation, the adversary should not learn anything beyond the search outcome.

3.4 Searching with Fine-grained Access Control

The public cloud storage are used by multiple users. For secure and effective utilization of data, in addition to searching over the encrypted data, it is required to enforce the access policies. The access policy enforced by the data owner directs the cloud server for assigning access privileges of the data to a cloud user. In symmetric searchable encryption schemes, to enforce the access policy the data owner himself can distribute the shared secret key with a group of users who are allowed to access the data. In [28, 67] the authors have provided multi-user keyword based searchable symmetric encryption techniques which can efficiently work in single-sender-multiple-receiver scenario. In the multi-user searchable symmetric encryption scheme proposed in [28] the data owner encrypts the index of keywords with his secret key. To share the data with other users, the data owner makes use of broadcast encryption scheme. For each data user the data owner generates a new secret key and assigns it to that user. The scheme also facilitates the data owner to revoke the access rights from the user. The scheme requires that the secret key should be delivered to the data user in a secure manner. In [67], the authors have presented multi-client searchable encryption scheme, that supports multi-keyword search query. In their scheme, the data users are required to retrieve a search token from the data owner. Therefore, the data user can only make search for keywords, for which he has been issued the search tokens. The multi-user searchable symmetric encryption schemes as presented in [28, 67] requires the data owner to make a secure communication with each authorized data user for issuing the secret key or search token. Therefore, such schemes can not perform well in multi-sender-multi-receiver scenario. As an alternative, searching over attribute based encrypted data is more optimal idea. As discussed in chapter - 2, the ABE is a public key cryptographic technique which provides confidentiality with fine-grained access control. Therefore, searching over attribute based encrypted data provides both searching over encrypted data and enforcing fine-grained access control. The ABE works well in multi-sender, multi-receiver scenario. It does not require a direct interaction between data owner and data receiver.

Definition 12. *The attribute based searchable encryption scheme is defined as a tuple as follows:*

Setup(1^l) $\rightarrow$ **(***MSK,PK***):** *The AC runs this algorithm. The algorithm takes as input the security parameter l and sets up the master secret key MSK and public key PK.*

KeyGen*(MSK,L)*$\rightarrow$ **(**SK_L**):** *This algorithm is run by the AC for each user in the system. It takes as input the MSK and the set of attributes L. It outputs the secret key* SK_L *for that user.*

Encrypt_Index **(***W,PK,T***)** $\rightarrow$ **(**CT_w**):** *Each data owner generates an encrypted index* CT_w *for keyword set (W) extracted from his data collection and encrypt it using the public key parameters PK and access policy T.*

Trapdoor*(kw,*SK_L*)*$\rightarrow$ **(***tw***):** *The data user generates a trapdoor using this algorithm, with the input of a keyword kw and the secret key* SK_L *The output of the algorithm is a trapdoor tw which works as the search token.*

Search*(*CT_w*,tw)*$\rightarrow$ **(**$true/false$**):** *The cloud server runs this algorithm with the input of an encrypted index and the trapdoor tw. The algorithm returns true if the encrypted index contains the keyword hidden inside tw and the access policy of encrypted index and attributes of user who sent the tw are matched.*

3.4.1 Literature Review on Searchable Encryption with Fine-grained Access Control

First in [33], Wang *et al.* have presented an attribute based keyword searchable encryption scheme that provides single keyword based search function on ciphertext policy attribute based encrypted data. They have integrated public key searchable encryption scheme (PEKS) proposed by Boneh *et al.* in [29] with classical CP-ABE. The access policy structure in their scheme is represented in form of LSSS matrix. However, their scheme requires the number of keywords to be finite in the universe.

In 2014, Zheng *et al.* have presented attribute based keyword searchable encryption scheme which also facilitates to verify the search results [68]. The access structure in their scheme is presented in form of "Access Tree" structure. For facilitating the search verification process, both the data owner and data user obtains secret keys from trusted authority. Using the secret key for encryption, the data owner encrypts the data and its associated index of keywords. A data user uses his secret key to generate the search token and verify the search process results. The authors have used the "Bloom Filter" technology to generate the keyword signature which helps the user to verify the search results.

In [69], Liu *et al.* have claimed that the search verification operation cost in the scheme of [68] is costly. Liu *et al.* have presented a keyword searchable encryption scheme with the same provision of verifiable keyword based search encryption. The verification operation in [69] is more efficient when compared to that of [68]. The verification operation in [68] requires a number of pairing operations and exponentiation operation linear to the number of user attributes. In the searchable encryption scheme presented in [69], the verification process does not include any bilinear pairing operations. Despite all these advantages, the scheme also requires that to generate a search token the data user has to interact with the trusted third party who has setup the systems. This creates an overhead in search operation, because every time before making a search request, the user first interacts with the trusted authority to derive the search token and then sends that token to the cloud server for conducting search operation. This also creates a bottleneck problem on trusted third party who has setup the system. Because in a system, there are multiple users and each user sends multiple search queries.

In [70], Liang *et al.* have provided keyword based search functionality and proxy reencryption both together on attribute based encrypted data. They have addressed both the issues of data sharing and data searching. The keyword based search operation allows a user to retrieve a subset of documents containing that keyword, and proxy reencryption allows the data user to share the encrypted data with other users. To achieve the keyword secrecy, Liang *et al.* have used the asymmetric pairing in the construction of scheme. The access structure in their scheme is represented in form of LSSS matrix. The scheme has been proven secure in the random oracle model. The scheme also supports partial decryption cost to be outsourced on cloud server side and reduce the final decryption cost on user side.

In [71] two schemes are provided for attribute based keyword searchable encryption. In both the schemes the data owner defines the access structure in form of LSSS matrix. In one scheme the user expresses the search policy in form of "OR" gate between his attributes. In second scheme the user expresses the search policy in form of "AND" gate between his attributes. In [72], the authors have presented a searchable encryption scheme, where they have taken the approach of "Online-offline" computation to prepare a searchable encryption scheme suitable for small hand-held battery-driven devices. In their scheme the keyword encryption and search token generation operations both are divided in two phases. The first preparation phase performs the computationally intensive tasks to generate the partial ciphertext or search token without revealing the actual ciphertext or keyword. This phase can be accomplished offline. The second phase is performed online and it generates the final ciphertext or search token from the results of preparation phase and the input of actual keyword. The computation in second phase is light-weight and fast. Therefore, the scheme is applicable to mobile devices.

In the scheme of [73], Hu *et al.* have addressed the issue of dynamically updating the access policy of an encrypted index. The authors of [73], have integrated proxy reencryption services within the keyword based searchable encryption to support the dynamic update of access policy. The access structure in their scheme is presented in form of Access tree structure. Unlike the access policy representation in [13], the access policy in their scheme only supports AND gate and OR gate for the non-leaves nodes of the tree and does not support k-out-of-m threshold gates. The access policy update operation can be initiated by the data owner. One constraint in the data update operation is that the data owner is allowed only to increase or decrease the leave nodes under the threshold gates in the access policy tree. For example, if in the access tree a non-leaf node represents m children, then the access policy can be updated to make total m+1 or m-1 children under that "AND" gate node. But the data owner can not change a threshold gate "AND" to an "OR" gate. Also, he can not increase the number of non-leaf nodes.

All the above discussed schemes have presented attribute based keyword searchable encryption schemes. These schemes supports keyword based search function with fine-grained access control. But all of them require the access policy details to be in clear form attached with encrypted index. Also, the user attributes should be stated in clear form inside the search query. This information affect the data confidentiality as going to be discussed in next section.

3.4.2 Need of receiver anonymity in attribute based keyword searchable encryption

Many existing schemes have provided keyword based searching on attribute based encrypted data, but not all of them have addressed the receiver anonymity. In all the above discussed schemes the access control policies are in clear form, which reveals the receiver information. The receiver information leads to compromising the data security. Therefore, in addition to data confidentiality, preserving privacy of data access is a practical need, as one could guess the purpose of the ciphertext by identifying the receiver of the ciphertext. In attribute based keyword searchable encryption techniques, if the access policies are in plaintext form, then a user who sends the search query, will send his attribute information in clear form to the storage server. The cloud storage server as an adversary has a view to the access policy and the user attribute information. From these information and the result of search outcome, the adversary's advantage of guessing the contents within the ciphertext will be increased.

Let us take the example of e-healthcare organization. Let a patient encrypts

Scheme	Type of ABE	Type of Pairing	Access Policy Structure	Public Key Size	Encrypted Index Size	Search Token Size	Search Operation Time Complexity	CO
[33]	CPABE	Symmetric	LSSS	$2\|G_0\| + \|G_1\|$	$(2N' + 1)\|G_0\| + (N_w)\mathbb{Z}_p$	$(n + 2)\|G_0\|$	$O(n)T_P + O(n)T_M + O(n)T_E$	$O(1)$
[68] -1	KPABE	Symmetric	Tree Based Structure	$4\|G_0\|$	$(N' + N_w + 2)\|G_0\|$	$(2n + 2)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$
[68] -2	CPABE	Symmetric	Tree Based Structure	$4\|G_0\|$	$(2N' + N_w + 2)\|G_0\|$	$(2n + 3)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$
[69]	KPABE	Symmetric	Tree Based Structure	$4\|G_0\|$	$(N' + 6 + N_w)\|G_0\|$	$(2n + 2)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$
[70]	KPABE	Asymmetric	LSSS	$(2N + 10)\|G_0\| + 3\|G_1\|$	$(N' + 3)\|G_0\| + N_w\|G_1\|$	$(n + 2)\|G_0\|$	$O(1)T_p + O(n)T_M + O(n)T_E$	$O(\tau)$
[71]	KPABE	Symmetric	LSSS	$(N + 2)\|G_0\| + \|G_1\|$	$(N' + 1)\|G_0\| + (2N_w)\|\mathbb{Z}_p\|$	$(3n + 1)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$
[72] -1	KPABE	Asymmetric	Tree Based Structure	$4\|G_0\|$	$(N' + N_w + 2)\|G_0\|$	$(2n + 2)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$
[72] -2	KPABE	Asymmetric	Tree Based Structure	$4\|G_0\|$	$(2N' + N_w + 2)\|G_0\|$	$(2n + 3)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$
[73]	CPABE	Symmetric	Tree Based Structure	$4\|G_0\|$	$(2N' + N_w + 2)\|G_0\|$	$(2n + 3)\|G_0\|$	$O(n)T_p + O(n)T_M$	$O(1)$

Table 3.1: Comparison of existing keyword based searchable ABE; CO: Communication Overhead between data user and trusted authority for generating τ number of search tokens; n: Number of attribute possessed by a user; N: Total number of attributes in the universe; N': number of attribute values included in access policy; n_w: number of keywords included in an encrypted index; T_P : time of pairing T_M,T_E and T_H denotes the time of pairing operation, multiplication, exponentiation operation and hash operation

and uploads his medical report on the cloud storage and sets the access-policy as “Doctor”, “Hospital-A”, “Oncology”. From the access policy of a document, the adversary guesses that the report might be a cancer patient report. When a doctor working as an oncologist in Hospital- A sends his search query to the cloud server, then from the search operation results, the adversary gets confirmed that the query might contain a word from the terminology related to Cancer disease. Therefore, the access policy of an encrypted document (and its encrypted index) and the attributes of user who sends the search query should be hidden to preserve the keyword and ciphertext confidentiality.

Many few schemes in the literature have addressed the issue of receiver anonymity in attribute based searchable encryption technique such as those in [74, 75, 76]. However, these schemes suffer either from performance barrier or from security flaw.

3.4.3 Security Model

The security model for the searchable attribute based encryption with hidden access policy is denoted as IND-CP-CKA (Indistinguishability against Ciphertext Policy and Chosen Keyword Attack) model as desscribed in chapter 2.The formal description of IND-CP-CKA model is as follows.

Let the Π denote the Attribute based searchable encryption scheme with the tuples $\langle$Setup, KeyGen, Encrypt_Index, Trapdoor and Search$\rangle$. In IND-CP-CKA model, the $\mathcal{A}$ is given access to the Oracle for Trapdoor. It can retrieve a number of trapdoors for his chosen keyword and chosen set of attribute values. In the challenge phase the $\mathcal{A}$ issues two pairs of collections of Words, and access policy as (W_0, T_0) and (W_1, T_1) where $|W_0| = |W_1|$. A bit b is selected in random and accordingly the encrypted index of W_b with respect to access policy T_b is computed as $CT_{W,b}$ and given to $\mathcal{A}$. Once again $\mathcal{A}$ is given access to the Trapdoor oracle. The restriction imposed on the $\mathcal{A}$ is that, he can retrieve trapdoor tw from Trapdoor Oracle which can give a successful search with both the $CT_{W,0}$ and $CT_{W,1}$ or none of them. If $\mathcal{A}$ has retrieved a trapdoor for a word w and an attribute list L scuh that L can satisfy both T_0 and T_1, then $w \in W_0$ and $w \in W_1$. At last the $\mathcal{A}$ issues a bit-value b'. The $\mathcal{A}$ wins the game if $b = b'$.

```
$IND - CP - CKA_{\pi}^{\mathcal{A}}(l)$

$(PK, MSK) \leftarrow_{\$} Setup(1^l)$
$(W_0, T_0^*)(W_1, T_1^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{Trapdoor}}(1^l, PK, L, w)$
If $|W_0| \neq |W_1|$ then return $\perp$
If F($L$,$T_0^*$) $\neq$ F($L$,$T_1^*$) then return $\perp$
If F($L$,$T_0^*$) = F($L$,$T_1^*$) = 1 then $w \in W_0$ AND $w \in W_1$
$(CT_{W,b}) \leftarrow_{\$} Encrypte\_Index(PK, W_b, T_b^*)$
$b' \leftarrow_{\$}$ $\mathcal{A}^{\mathcal{O}_{Trapdoor}}(1^l, PK, L, w, EI_b)$

    If F($L$,$T_0^*$) $\neq$ F($L$,$T_1^*$) then return $\perp$
    If (F($L$,$T_0^*$) = F($L$,$T_1^*$) = 1) AND (($w \in W_0$ AND $w \notin W_1$) OR($w \notin W_0$ AND $w \in W_1$))
            then return $\perp$
    return $b'$
return $b' = b$
```

Definition 13. *A searchable ABE scheme is secure in IND-CP-CKA secure, if the advantage of adversary $\mathcal{A}$ as defined below is negligible.*

$$\mathsf{Adv}_{\Pi,\mathcal{A}}^{\mathrm{ind-cp-cka}} l = \frac{1}{2} - \Pr\left[1 \leftarrow IND - CP - CKA_{\pi}^{\mathcal{A}}(l)\right]$$

3.5 Data-Owner based Searchable Encryption Scheme - DOSE

The security essentials for a searchable encryption scheme as specified in [29, 77] from the side of an unauthorized users are as follows:

- Data Privacy: Any information about the data content should not be revealed from the search process and its results.

- Data Owner Privacy: From the encrypted data, the data owner's identity should not be revealed.

- Data Receiver Anonymity: The information about the intended receivers of a ciphertext should be remain hidden.

The searchable encryption schemes presented in [22, 78, 79] satisfy these above mentioned security requirements at the cost of reducing the search efficiency. They provide keyword based searching but at the same time they require a computation intensive cryptographic operations for matching and verifying the contents in encrypted index and search query. They require the initial complex cryptographic operations before performing, the keyword matching verification between the query and encrypted index. As the searching unit should perform the above discussed computation for every entry of encrypted contents thoroughly per search request, the retrieval performance degrades significantly in cloud storage systems, where frequent queries are sent to the CSP. This issue demands more security concern when there are number of intended receivers for one subset of data content uploaded by a data owner, resulting in redundant copies of encrypted contents for unique identity of each receiver. This property of complex mathematical operations and redundant copies of data restrict the one-to-many feature provided by the ABE.

To overcome this limit of inefficiency in data retrieval process, Koo *et al.* have presented a searchable encryption scheme with improved search efficiency and data retrieval process in [75]. Their scheme supports only data owner based search and not feasible to provide the keyword based searching.

The concept of data owner based search helps in many real time scenarios. When searching over the encrypted documents, identity of data owner who has encrypted and uploaded the document can be one of the search criteria. For example, a doctor wants to search for the medical records which have been uploaded by her patient "Bob". In a system, each data owner can be assigned a unique identity for the purpose of data owner based searching. In the searchable anonymous attribute based encryption scheme proposed by Koo *et al* in [75], search on encrypted data is done using data owner's identity and data retriever's attributes. The scheme claimed that a user in the system can search on encrypted data stored in cloud with preserving sender and receiver anonymity. While analyzing the scheme we found that the scheme of [75] have serious security flaws and it fails to preserve the receiver anonymity. After identifying those security flaw, we have presented the searchable encryption scheme, that provides the search operation using data owner's identity. The search operation of proposed scheme preserves the receiver anonymity.

In the scheme of [75], the access structure is represented in form of an access tree $\mathcal{T}$. The system model of [75] is also same as defined in the section 2.3. The scheme definition of [75] is similar to the definition of CP-ABE scheme described in section 1. In the scheme of [75], the attributes of a ciphertext access policy are published in scrambled format and the user performs the search operation about which access policy is satisfied with his attributes without revealing his attribute information.

In the scheme of [75], the data owner scrambles the attributes involved in the access tree with a random pseudonym generated from his unique identity. A user who wants to perform the search operation needs to scramble his attributes using this pseudonym. The list of pseudonyms from all data owners is published by the cloud server (CS). The cloud does not know which pseudonym refers to which data owner. The user picks any pseudonym, scrambles his attributes using that pseudonym and then sends these scrambled attributes to the CS. The CS performs the look-up and match operation to retrieve the documents which are accessible for the user. Because of the scrambled attributes, the CS is not able to learn the attributes of the receiver. In this way, Koo *et al.* have addressed the issue of receiver anonymity in searchable encryption.

In the scheme of Koo *et al.*, the list of pseudonyms is published by CS. Each pseudonym is generated randomly by the data owner. We have identified that the construction of the scheme allows a curious CS to generate the pseudonym himself using the public key parameters and a randomly chosen value. The user gets the pseudonym from the CS and he has no way to identify that the pseudonym is from a valid data owner or it is a fake one. If the user scrambles his attributes using the pseudonym generated by the CS and send that scrambled attributes to the CS, then the CS is able to learn the attributes of the receiver.

3.5.1 An Improved Scheme

To mitigate this flaw, we have designed a scheme with the objective of search using the data owner's identity [80]. The proposed scheme enables a user to retrieve the data which is accessible to him without revealing his attribute information to the CS. The improved scheme has the following phases.

Setup(1^l) $\rightarrow$ (MSK,PK): The AC performs the setup. It chooses a bilinear group G_1 of prime order p with generator g and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. It picks two random exponents α and β from Z_p and also selects a cryptographic hash function $H_0 : \{0,1\}^* \rightarrow G_1$. AC computes the public parameter PK and master secret MSK for the system as: $PK = \langle G_1\ G_2, g, \omega = e(g,g)^{\alpha}, h = g^{\beta} \rangle$, $MSK = \langle\ g^{\alpha}, \beta \rangle$.

KeyGen$_L$(MSK,L)$\rightarrow$ (SK_L): Each receiver gets a secret key SK from AC for decryption operation.

- The TA chooses a random $r \in Z_p$ for each individual user in the system and $r_j \in Z_p$ for each attribute $L_i \in L$. Here L is the set of attributes that belongs to user. The private key SK_L for generating search query is computed as

  $$SK_L = \langle \{D_j = H(L_i)^{\beta}\}_{L_i \in L} \rangle$$

  Here we do not provide the secret key components for decryption of a ciphertext, because we are providing the construction only for searching over and retrieving the encrypted documents. The secret key components for decryption of a cipher document are same as that provided in classical CP-ABE scheme [13].

KeyGen$_O$(MSK,ID)$\rightarrow$ (SK_O): Each data owner gets a secret key SK_O from AC in which data owner identity (ID) is hidden.

- For the data owner having identity ID_0, AC computes and returns him the anonymous key, $SK_O = H(ID_0)^{\beta}$.

Encrypt_Index(SK_O,PK,T) $\rightarrow$ (CT_w): This scheme represents a data owner based searching over the encrypted dataset. Therefore, instead of keyword set W, this algorithm takes as input the data-owner's secret key SK_O, which contains the data owner's identity. The data owner encrypts data M as per the access policy T by running the Encrypt algorithm as in the conventional CP-ABE scheme [13]. The access policy is represented in the form of access tree $\mathcal{T}$. To generate the encrypted index entry, the attributes involved in the access policy of T are scrambled using the pseudonym SK_O and placed at the appropriate leaf nodes in $\mathcal{T}$. This updated access tree $\mathcal{T}'$ of scrambled attributes is paired with the identifier of M (denoted as ID_M) in the index. The data owner garbles each attribute value included in T using his secret key SK_O. Let S is the set of attributes which are included in the access policy T (an accordingly in leaf nodes of $\mathcal{T}$). For each attribute $\{attr_i\}_{1 \leq i \leq n}$ included in T, the data owner computes

$$\begin{aligned} K_{O,T} &= \{e(SK_O, H(attr_i))\}_{attr_i \in T} \\ &= \{e(H(ID_O)^{\beta}, H(attr_i))\}_{attr_i \in T} \\ &= \{e(H(ID_O), H(attr_i))^{\beta}\}_{attr_i \in T} \end{aligned}$$

and replaces each leaf node $attr_i$ in $\mathcal{T}$ generated for access policy T with the corresponding element scm_{att_i} from $K_{O,T}$. This results in the access tree $\mathcal{T}'$. After the scrambling phase, the data owner uploads CT and an index entry CT_W = ($\mathcal{T}'$, ID_M) to the storage managed by the CS.

Trapdoor(ID_O,SK_L)$\rightarrow$ (tw) : Instead of kw, this algorithm takes as input the data owner's identity ID_O for which the user wants to make the search operation. Unlike the scheme of [75], in this algorithm there is no need for a retriever to acquire a pseudonym of any data owner. When the retriever determines to retrieve a data with identity ID_O from the CS then the retriever generates cryptographic index terms for corresponding attributes as

$$\begin{aligned} K_{O,L} &= \{e(D_i, H(ID_O))\}_{L_i \in L} \\ &= \{e(H(ID_O), H(L_i)^{\beta})\}_{L_i \in L} \\ &= \{e(H(ID_O), H(L_i))^{\beta}\}_{L_i \in L} \end{aligned}$$

After that, the retriever submits the data request query in form of trapdoor tw in form of $K_{O,L'} \subseteq K_{O,L}$ to the CS.

Search(CT_w,tw**)**$\rightarrow$ **(**$true/false$**)** After receiving search query tw in form of scrambled index terms $K_{O,L'}$, the CS searches in his database if the set of attributes included in $K_{O,L'}$ is satisfied by any of the $\mathcal{T}'$ listed in the index. This is done by the algorithm $\mathcal{C}(\mathcal{T}', K_{O,L'})$. The algorithm returns *true* or *false*.

Let $\mathcal{T}'_x$ be a subtree of $\mathcal{T}'$ with root node x and X' = {$x' \in Y_x$ and parent(x') = x}. $\mathcal{C}(\mathcal{T}'$,K_{O,Λ'_i}) is computed recursively as follows. If x is a leaf node, $\mathcal{C}(\mathcal{T}'_x$,$K_{O,\Lambda'_i}$) returns true if and only if $attr_x \in K_{O,\Lambda'_i}$. If x is a non-leaf node in $\mathcal{T}$, $\mathcal{C}(\mathcal{T}'$,K_{O,Λ'_i}) returns true if and only if at least k_x children return true. For each ciphertext CT_i, where $0 \leq i \leq m$, the CS simply follows the access tree $\mathcal{T}'_i$ and determines whether $\mathcal{C}(\mathcal{T}'_i$,$K_{O,\Lambda'_i}$) returns *true* or not. The CS sends the ciphertexts to the retriever for which the algorithm $\mathcal{C}(\mathcal{T}'_i$,$K_{O,\Lambda'_i}$) returns true.

3.5.2 Security Analysis of DOSE

Theorem 3.1. *The improved scheme provides sender and receiver anonymity.*

Proof. We prove that if an adversary $\mathcal{A}$ is able to break the sender or receiver anonymity, then we can build a simulator that can break the hardness of decisional bilinear diffie-hellman assumption. We consider a challenger $\mathcal{C}$, a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$.

Init : $\mathcal{A}$ submits two pairs (ID^0,L^0) and (ID^1,L^1) to $\mathcal{S}$ on which he wishes to be challenged upon. Here L^b ($b \in$ {0,1}) is set of attributes{L^b_1, L^b_2,$\cdots$,L^b_n}.

Setup: $\mathcal{C}$ setups the bilinear groups G_1 and G_2 of prime order p, a generator g of group G_1 and a bilinear map operation e: $G_1 \times G_1 \to G_2$. He flips a coin μ, outside of $\mathcal{S}$'s view. If $\mu = 0$ then $\mathcal{C}$ sets {g, $X = g^x$, $Y = g^y$, $Z = g^z$, $\Delta = e(g,g)^{xyz}$}, else he sets {g, $X = g^x$, $Y = g^y$, $Z = g^z$, $\Delta = e(g,g)^{\theta}$} for some random value θ. The $\mathcal{S}$ selects a random value α from the field $\mathbb{Z}_p$, and sets $\beta = z$. Accordingly the public key parameters are computed and given to $\mathcal{A}$.

Phase 1 : $\mathcal{A}$ issues a polynomial number of queried for following:

- Hash queries : The $\mathcal{A}$ issues number of queries for retrieving $H(ID)$ and $H(L_i)$. For each $ID \neq H(ID^0) \neq H(ID^1)$, the $\mathcal{S}$ returns $g^{H_0(ID)}$, where H_0: {0,1}$^* \to Z_p$ is a random oracle function. Similarly, for $H(L_i) \neq H(L_i^0) \neq H(L_i^1)$, the $\mathcal{S}$ returns $g^{H_0(L_i)}$. Each of these queries and its responses are recorded by $\mathcal{S}$. For $H(ID^b)$, the $\mathcal{S}$ returns $A \cdot g^{H_0(ID^b)} = g^{a+H_0(ID^b)}$. In response to query for retrieving $H(L_i^b)$, the $\mathcal{S}$ returns $B \cdot g^{H_0(L_i^b)} = g^{b+H_0(ID^b)}$. For ID^b and for each of L_i^b, the output hash values are recorded.

- Secret Key for search query generation : $\mathcal{A}$ submits the list of attributes L to retrieve the secret key for generating search query with respect to the set of attributes L. $\mathcal{C}$ aborts the operation if for any attribute $L_i = L_i^b$ for $1 \leq i \leq n$. Else it generates the secret key for search operation as follows.
  For each $L_i \in L$, generate $\langle\{D_i = g^{H_0(L_i)c}\}\rangle$.

- Secret Key for generating Encrypted Index : $\mathcal{A}$ submits an identity to retrieve the secret key for generating encrypted index with respect to the identity ID. $\mathcal{C}$ aborts the operation if $ID = ID^b$. Else it generates the secret key for encrypt_index as $SK_O = g^{H_0(ID)c}$.

Challenge: $\mathcal{S}$ flips coin and selects a number $b \in$ {0,1} and accordingly submits an encrypted index for the pair (ID^b, L^b) as $e(H(ID^b), H(L^b))^{\beta} = \Delta \cdot e(X,Z)^{H_0(L_i^b)} \cdot e(Y,Z)^{H_0(ID^b)} \cdot e(g,Z)^{H_0(ID^b)H_0(L_i^b)}$ for $1 \leq i \leq n$}.

Phase 2: $\mathcal{A}$ issues polynomial number of queries with the same restrictions as imposed in Phase 1.

Guess: $\mathcal{A}$ guess the value of b'. If $b' = b$, then $\mathcal{S}$ replies with $\mu' = 0$, else $\mu' = 1$.

If $b' = b$, then $\mathcal{S}$ outputs μ=1 to indicate that it was given a valid DBDH-tuple, else, it outputs μ=0 to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about b, in turn, $Pr[b \neq b'|\mu = 0]$= $\frac{1}{2}$. As the $\mathcal{S}$ guesses μ'=0 when $b \neq b'$, $Pr[\mu = \mu'|\mu = 0] = \frac{1}{2}$. If $\mu = 1$, then the $\mathcal{A}$ is able to view a valid encryption of message with advantage $\epsilon_{dbdh}(l)$, a negligible quantity in security parameter l. Therefore, $Pr[b = b'|\mu = 1] = \frac{1}{2} + \epsilon_{dbdh}(l)$. Similarly, the simulator $\mathcal{S}$ guesses μ'=1 when $b = b'$, in turn, $Pr[\mu' = \mu|\mu = 1] = \frac{1}{2} + \epsilon_{dbdh}(l)$. The overall advantage of the simulator in DBDH game is $\frac{1}{2} \times Pr[\mu = \mu'|\mu = 0]$ $+\frac{1}{2} \times Pr[\mu = \mu'|\mu = 1]$ - $\frac{1}{2}$ = $\frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times (\frac{1}{2} + \epsilon_{dbdh}(l))$ - $\frac{1}{2}$ = $\frac{\epsilon_{dbdh}(l)}{2}$. Therefore, if the $\mathcal{A}$ has advantage $\epsilon_{dbdh}(l)$ in the above game instance, then we can build a simulator (S) which can break the DBDH problem with negligible quantity $\frac{\epsilon_{dbdh}(l)}{2}$.

Hence, the theorem. □

3.6 Receiver Anonymous (keyword)Searchable Encryption - RASE

In an AABE scheme to preserve the data secrecy, the receiver identity should be kept anonymous and for this purpose the access policy of a ciphertext is remain hidden inside the ciphertext. The receiver anonymity should not be compromised when making the search operation. Therefore, when a user wants to retrieve a subset of documents containing a keyword and accessible to him, the user keeps his attributes hidden inside the search query. Let a patient suffering from "Psoriasis" may set the access policy of encrypted report as doctor in Dermatology, and keep the access policy hidden inside the ciphertext to keep secret his type of disease. If the dermatologist while making the search query, reveals his attributes to the server then the server will learn that the search result contains the reports of patients having skin disease. In this way, the receiver identity can help an adversary to learn the information from the ciphertext. Therefore, it is required to preserve the receiver anonymity when making the search operation.

3.6.1 Related Work

Our previous scheme DOSE [80] helps a user to anonymously retrieve the cipher documents from cloud storage whose access policy is satisfied by the user's attributes. However, that scheme supports only data owner identity based search. Most of the times, the user is interested to retrieve the documents which are accessible to him and which contains a specific keyword. Therefore, a searchable scheme can be more practical and useful if it provides a keyword based search over attribute based encrypted data with hidden access policy.

In 2014, Shi *et al.* have presented the scheme Authorized Searchable-Public Key Encryption (AS-PKE) in [74] which facilitates the multi-keyword search over attribute based encrypted data. Their system supports multi-valued attributes, and the access policy is represented in form of LSSS matrix. When analyzing the scheme we have found some performance bottleneck issues in the scheme. The AS-PKE scheme of Shi *et al.* supports the multi-valued attributes, however to hide the access policy at most only one attribute value can be placed in the access policy. This is because, as defined in the definition -2.2.3 for LSSS, the function ρ maps each attribute to a single row. The performance issues found in the AS-PKE scheme by Shi *et al.* are listed as below.

- The AS-PKE scheme supports multi-valued attributes, but in the access policy only one value per attribute can be placed. This will be a drawback, when a data owner wants to send the ciphertext to more than one user having the different attribute values for an attribute. For example, a patient's medical report suffering from leukemia should be sent to the person working as doctor in either hematology or oncology department. In such cases for each different attribute value of an attribute a different ciphertext should be generated.

- AS-PKE supports arbitrary boolean predicates constructed from multi-keyword. However, it should be noted that the scheme supports only finite number of keyword fields. The search query predicate is also represented in the style of LSSS. Each row of the matrix in search query refers to one keyword field.

Therefore, at most one keyword from each keyword field can be included in the access policy. Also, the function which maps the row of a search predicate matrix to a keyword field can disclose the type of keyword for which the user is making the search. This can give benefit to an adversary for guessing the keywords hidden inside the search query. Also, the number of keyword fields are bounded at the time of setup. Therefore, a user can not make a search query for an keyword which does not belong to any of the predefined keyword fields.

- The scheme AS-PKE requires that a user can not generate the search query himself. To generate a search query, he has to acquire the search token from the trusted third party. This creates a performance bottleneck, because every time before making a search query, the user has to interact with trusted third party. This yields an delay in getting the results of search operation.
- The construction of AS-PKE is based on composite order bilinear group. The study on bilinear pairing has proved that in case of performance the prime-order bilinear pairing operations are more effective when compared to composite order bilinear pairing operations [81]. Shi *et al.* have also agreed on the fact that because of composite order bilinear pairing operations, their scheme can not be highly practical and their major contribution is on theoretical aspects only.

Recently in 2017, Wang *et al.* have provided an attribute based keyword searchable encryption scheme [76] that supports "AND gate on multi-valued attributes", as discussed in 2.2. However, in that scheme, T_i contains only one element for the attribute i. It means $T_i \in V_i$ instead of $T_i \subseteq V_i$. They have used 3-multi-linear map. The authors of [76] have claimed that their scheme provides keyword based searching with receiver anonymity. We have analyzed that their scheme has a security flaw and it fails to preserve the receiver anonymity. Also, the scheme fails to preserve the keyword privacy. The security issues identified in the scheme of [76] are discussed as below.

- In the scheme of [76], during the setup phase, the Attribute Center AC generates one of the public key parameter g_0 as the generator of group G_1. When making the encrypted index, the data owner selects the access policy T = $\{T_1,T_2,\cdots,T_n\}$ where T_i is an element $v_{i,j}$ chosen from the valueset V_i for an attribute i ($1 \leq i \leq n$). For each element T_i, the ciphertext generated as C_i = $H(T_i)^s$. One of the other cipher component generated is $\hat{C}$ = g_0^s. An adversary can calculate if an attribute value $v_{i,j}$ is included in access policy or not with the result of following equation.

$$e_0(H(v_{i,j}),\hat{C}) \stackrel{?}{=} e_0(C_i,g_0)$$

If the equation returns true, then T_i =$v_{i,j}$. In a system where number of attributes are n and each attribute is having valueset of maximum size m, then at most $2 * m * n$ pairing operations are performed by the adversary to disclose the access policy.

- Let the adversary $\mathcal{A}$ has gain an access to a trapdoor for a known keyword w as tw = $e_0(\prod_{i=1}^{n} H_0(T_{i0})^w,\check{D})$ = $e_0(\prod_{i=1}^{n} H_0(T_{i0}),\check{D})^w$. Here $\check{D}$ is a secret key component issued to a user. To generate the new trapdoor for a keyword w', the $\mathcal{A}$ has to replace the value of w with the value of w' using following equation.
$$tw' = tw^{\frac{w'}{w}} = e_0(\prod_{i=1}^{n} H_0(T_{i0}),\check{D})^{\frac{w'w}{w}}$$

3.6.2 The Proposed Scheme

In this subsection, we present a scheme which provides a receiver anonymous keyword searchable attribute based encryption. The system model and access policy of RASE are same as those of scheme provided by Wang *et al.* in [76]. To fulfill, the purpose of searchable encryption, we have designed the solutions for encrypted index, trapdoor and search algorithm. We do not provide the encryption and decryption of a document. The scheme is designed with 3-multi-linear mapping [82, 83]. The scheme works as follows:

Setup(1^l)→ (MSK,PK): The AC performs the setup phase. In this phase, the multi-linear groups and other parameters required for multi-linear mapping are generated as Y = {p, G_1, G_2, G_3, e_0, e_1 }. The pairing functions are defined as e_0:G_1 × G_1 → G_2 and e_1:G_1 × G_2 → G_3. The AC also chooses two random generators g_1 and g_2 from group G_1, and a hash function H_0 ia defined as H_0: $\{0,1\}^* \rightarrow G_1$. The master secret key MSK is chosen as $\langle$ α, $\beta \in_R Z_p$ $\rangle$. The corresponding public key PK $\langle$ Y, g_1, g_2, g_1^{α}, g_2^{β} $\rangle$ is published.

KeyGen(MSK,L)→ (SK_L): Let L=[$L_1, L_2, \cdots, L_n$] be the attribute list for the user who requires a secret key. For every user in the system the AC picks a random value ρ and generates a user's secret key SK for search operation as follows.
$\hat{D}$ = $(g_2 \prod_{i=1}^{n} H_0(i \| L_i))^{\alpha} \cdot g_2^{\rho}$
$\bar{D}$ = $g_1^{\frac{\rho}{\beta}}$

Encrypt_Index(W,PK,T) → (CT_w): We provide the construction of cipher components for encrypted index only. We do not include encryption of a message. The algorithm takes the access policy T and public key PK as input. Here T =$\{T_1, T_2, \cdots T_n\}$ where T_i $\{1 \leq i \leq n\}$ is a value chosen from valueset V_i for an attribute i. To prepare the cipher components for encrypted index the encryptor performs following steps :

- Select random secret values s and t from $\mathbb{Z}_p$.
- For attribute values included in T, create the cipher component $\tilde{C} = g_2^t \prod_{i=1}^{n} H_0(i \| T_i)^{st}$.
- Compute $\hat{C} = g_1^{st}$,$\bar{C} = g_2^{st\beta}$
- Calculate $C' = e_0(g_1, g_2)^{\alpha(s-1)t}$. For each keyword $w \in W$, set $C_w = e_1(H_0(w), C')$

The encrypted index CT_w is formed as $\langle$ {$\tilde{C}$, $\hat{C}$, $\bar{C}$, C_w for each $w \in W$ $\rangle$

Trapdoor(kw,SK_L)→ (tw): The data user selects a random number ψ from $\mathbb{Z}_p$. It prepares the trapdoor tw with following steps

- Calculate $\tilde{D} = g_1^{\alpha\psi}$

- Compute $\bar{D}' = \bar{D}^{\psi}$ and $\hat{D}' = \hat{D}^{\psi}$
- Also compute, $D_w = H_0(w)^{\frac{1}{\psi}}$

The output of the algorithm is a trapdoor $tw = \langle \tilde{D}, \bar{D}', \hat{D}', D_w \rangle$

Search(CT_w,tw)$\rightarrow$ ($true/false$): A CS when receiving a trapdoor tw, performs this operation to search for the indexes which contains the keyword w sent in the tw, and whose access policy matches with the user credentials.

$$\begin{aligned} C'' &= \frac{e_0(\hat{C}, \hat{D}')}{e_0(\tilde{C}, \tilde{D}) e_0(\bar{C}, \bar{D}')} \\ C'_w &= e_1(D_w, C'') \end{aligned}$$

The server tries to match the value of C'_w with each entry C_w in the encrypted index. If for any value C_w is equal to the value of C'_w, then the server returns 1 else 0.

Correctness:

The search process only returns true if L satisfied T, that is for $1 \leq i \leq n$, $L_i \stackrel{?}{=} T_i$. The correctness of the search computation is as follows.

$$\begin{aligned} C'' =& \frac{e_0(\hat{C}, \hat{D}')}{e_0(\tilde{C}_{i,j}, g_1^{\alpha\psi}) e_0(\bar{C}, \bar{D}')} \\ =& \frac{e_0(g_1^{st}, (g_2 \prod_{i=1}^{n} H_0(i||L_{i,j}))^{\alpha\psi} \cdot g_2^{\rho\psi)}}{e_0((g_2^t \prod_{i=1}^{n} H_0(i||T_{i,j})^{st\psi}), g_1^{\alpha\psi}) e_0(g_2^{st\beta}, g_1^{\frac{\rho\psi}{\beta}})} \\ =& \frac{e_0(g_1^{st}, (g_2 \prod_{i=1}^{n} H_0(i||v_{i,j}))^{\alpha\psi} \cdot g_2^{\rho\psi)}}{e_0((g_2^t \prod_{i=1}^{n} H_0(i||v_{i,j})^{st\psi}), g_1^{\alpha\psi}) e_0(g_2^{st\beta}, g_1^{\frac{\rho\psi}{\beta}})} \\ =& e_0(g_1, g_2)^{(\alpha(s-1))t\psi} \end{aligned}$$

$$\begin{aligned} C'_w =& e_1(D_w, C'') \\ =& e_1(H_0(w), e_0(g_1, g_2))^{\frac{1}{\psi}(\alpha(s-1))t\psi} \\ =& e_1(H_0(w), e_0(g_1, g_2))^{(\alpha(s-1)t)} \end{aligned}$$

3.6.3 Security Analysis of The Proposed Scheme

To implement the scheme RASE we have used multi-linear mapping as a base operation. We have analyzed the weaknesses in multi-linear maps. The weaknesses of multi-linear maps and the safeguard we have taken to withstand against these weaknesses are discussed below.

All the candidate constructions of multi-linear mapping are found to be vulnerable against Zeroizing attack. An abstract view of the reason behind this is as follows. Let the groups involved in mult-linear mapping are G_1, G_2,$\cdots$,G_k. All the current multi-linear mapping groups are not based on elliptic curve as done with bilinear pairing. Instead, they provide encoding formula at different levels. For example, an element of $\mathbb{Z}_p$ is encoded at level 0. To map that point on group G_1, the element of level 0 is encoded to level 1. In this way, each point on group G_i can be mapped to a group G_{i+1} with further encoding. At each level i, an element m is encoded as $[\frac{e_m}{z^i}]_q$. Here e is the encoding of m at level 0 with some secret parameters depending upon the chosen implementation of multi-linear mapping (GGH or CLT). Similarly, the value of q is a large integer and its value is decided by the chosen implementation and the security parameter. The values required to derive the encoding e from m and the value of z are kept secret. The security of multi-linear maps lie beneath the belief that the parameters for encoding are secure. To protect the secrecy of encoding parameters, the level 0 encoding of some randomized vectors whose linear combination can produce the encoding of any plaintext element, the level-1 encoding of element 1, and several level-1 encoding of element 0 are placed in the set of public parameters. Other than these, a PointZero p_{zt} for the level k is also provided in set of public parameters (The p_{zt} is only used to check whether an element is zero or not at level k). The encoding of level 1 is provided as $[\frac{e_1}{z}]_q$. The encoding of 1 is multiplied with an element encoded at level i to map it to the level $i+1$. To randomize the encoding of any element, so that an adversary getting an element of group $i+1$ can not predict its i^{th} encoding, each element is added with encoding of 0 at level 1 which is in form $[\frac{e_0}{z}]_q$. The encoding of 0 provides a randomness in the encoding of the element, but in computation, it will not alter the result, because for any element a+0 = a. Thus the use of encoding of 0 is necessary to protect against backtracking. However, in literature it has been found that the multilinear mapping constructions are vulnerable to attack because of these encoding of zero at level one. The inclusion of several level - 1 encoding of 0 helps an adversary to conduct a mathematical cryptanalysis and find the secret parameters used for encoding. This will break the whole system.

We claim that our scheme defends against this attack, because in our scheme we do not require to provide the encoding of zeros. The encoding of 0 is required to randomize the encoding of other element at level 1. We will use the encoding of 0 to randomize the public parameters and provide these public parameters to the adversary along with other multi-linear mapping public parameters, but we do not provide encoding of 0s. The encoding of 0 at level one are kept as secret possessed by the system administrator. The public parameters g_1, g_2 from group G_1 are generated by the system administrator. While generating those system parameters the administrator will use 0 encoding to randomize those elements, therefore any element generated from g_1 and g_2, such as g_1^{α}, g_2^{β} and any other elements generated during KeyGen or Encrypt_Index phases will be randomized. Another requirement comes for randomization is during the use of H_0:$\{0,1\}^* \leftarrow G_1$, where we map the string to an element of group G_1. Here, we require to randomize the encoding otherwise, the adversary can break the pre-image property essential for Hash function. We overcome this problem, by providing the level 1 random encoding of each of the attribute value included in the universe of the system instead of providing 0 encoding at level 1. This is feasible because, the size of universe of attribute values is fixed in the system. Next we require the H_0 function to map a keyword kw to an element of group G_1. Here without the help of 0 encoding, we may produce a non-randomized encoding of w for level 1. But as shown in the construction of scheme, whenever making search operation, the user selects a random value ψ and randomize the $H_0(w)$ with the value of $\frac{1}{\psi}$. Therefore, here we do not require to provide 0 encoding to randomize the value. So, because of the above discussed reasons, we do not require to provide the encoding of 0 and as far as the encoding of 0 are not available, the cryptanalytic attacks on multi-linear mapping are infeasible. It yields that even though the scheme uses the insecure multi-linear mapping as the base operation, There are enough countermeasures in the construction of scheme to defend against the attacks on multi-linear mapping.

Below we are providing the security proof for the proposed scheme RASE. The underlying computation assumption in the construction of this security proof is Trilinear Decisional Diffie-Hellman Assumption which is a customized version of k-linear DDH assumption with k=3.

Theorem 3.2. *The proposed scheme is IND-CP-CKA secure based on the hardness of the Trilinear DDH assumption and the construction of secure multi-linear maps.*

Proof. In the proof, we show that the advantage of adversary to uncover the keyword from the encrypted index is negligible.

With the assumption that we provide a customized multi-linear mapping parameters which exclude the level - 1 encoding of 0, we prove that the advantage of an adversary in breaking the security of RASE in IND-CP-CKA model is same as that of breaking the hardness assumption of the Trilinear DDH assumption.

Setup: The $\mathcal{A}$ gives l as security parameter to the $\mathcal{C}$. The $\mathcal{C}$ runs the setup algorithm and returns the public key PK is sent to $\mathcal{A}$.

Phase 1:$\mathcal{A}$ is allowed to issue adaptively generated trapdoor queries with input keyword w and set of attribute values L. The $\mathcal{C}$ responds with tw generated with the input w and L.

Challenge: $\mathcal{A}$ submits two pairs (W_0,T^0) and (W_1,T^1). The input submitted by $\mathcal{A}$ must have to satisfy the below mentioned criteria. If either of them fails, then $\mathcal{C}$ aborts.

1. W_0 and W_1 are set of keywords with equal length.

2. $\mathcal{A}$ has not gained a trapdoor tw for the attribute set L, which satisfies either of T^0 or T^1.

The challenger $\mathcal{C}$ randomly chooses $b \in \{0,1\}$, then computes CT_{W_b} as an Encrypted Index of Keywords as per the input of W_b and T^b . $\mathcal{C}$ submits CT_{W_b} to $\mathcal{A}$.

Phase 2: Same as in Phase 1. $\mathcal{A}$ issues the adaptively generated queries with keyword w and a list of attribute values L which should follow at least one of the following criteria : (i) w should not be included in W_0 and W_1 (ii) F(L, T^0) = F(L, T^1) = 0. $\mathcal{A}$ is responded with tw corresponding to (w,L).

Guess: $\mathcal{A}$ outputs a guess b' of b. The adversary wins the game if $b' = b$. The advantage of $\mathcal{A}$ in this game is defined as Adv$_A$(l)= $|Pr[b' = b] - 1/2|$.

In the proposed scheme-RASE to uncover the challenge ciphertext, an adversary has to compute the value of $e_1(H_0(w), C')$. The adversary possess, the challenge ciphertext component $\hat{C}$ = g_1^{st}. The pairing operation of $\hat{C}$ with g_2 gives the adversary the value of $e_0(g_1, g_2)^{st}$. In the same way the adversary knows the public parameter g_1^{α}. Pairing of g_1^{α} with g_2 gives the adversary the value of $e_0(g_1, g_2)^{\alpha}$. Let us consider $H_0(w)$ as a random oracle function which responds with a value of the form $g_1^{H_1(w)}$, where $H_1(w)$ is a random oracle function for mapping the string w to an element of $\mathbb{Z}_p$. In response to the query for $H_0(w)$, the $\mathcal{A}$ is only given the value of $g_1^{H_1(w)}$ and not the value of $H_1(w)$.

To resolve a Trilinear DDH assumption, one has to decide whether for a given value $e_1(g_1, g_1, g_1)^z$, the $z \overset{?}{=} abcd$ when given the values g_1^a,g_1^b,g_1^c,g_1^d. We further make customization in the input and instead of providing g_1^a and g_1^b, we provide $\hat{g_2}$ = $e_0(g_1, g_1)^{ab}$. Here $\hat{g_2}$ is an element of group G_2. This will change the challenge tuple as $e(\hat{g_2}, g_1)^z \overset{?}{=} e(\hat{g_2}, g_1)^{abcd}$. Consider that g_2 = g_1^x for some number x. Now from the challenge ciphertext and the public key parameters, the adversary has following components which we can resemble with the challenge tuple of k-decisional Diffie-Hellman components as follows:

- $\hat{g_2}^{ab}$ = $e_0(g_1, g_2)^{st}$ = $e_0(g_1, g_1)^{xst}$

- g_1^c = g_1^{α}

- $g_1^d = g_1^{H_1(w)}$

The challenge task of $\mathcal{A}$ is to compute $e(\hat{g_2}, g_1)^{x\alpha st H_1(w)}$. In the proposed scheme, to discover the keyword inside the encrypted index, the $\mathcal{A}$ has to compute the value of $e_1(H_0(w), C_w)$, where $C_w = \frac{e_0(g_1,g_2)^{\alpha st}}{e_0(g_1,g_2)^{\alpha t}}$, and $H_0(w) = g_1^{H_1(w)}$. The advantage of this computation is equivalent to computing $e(\hat{g_2}, g_1)^{abcd}$ which is negligible and denoted as ϵ_{lbdh}.

Also, to uncover the value of $e_0(g_1, g_2)^{\alpha t}$, the $\mathcal{A}$ has to retrieve the value of g_2^t from the cipher component $\tilde{C}$. To uncover the value of g_2^t, $\mathcal{A}$ has to compute the value of $\prod_{i=1}^{n} H_0(i \| T_i)^{st}$ from the given values $\hat{C}$ and hash values of T_i. If we consider, g_1^{st} as g^x and $\prod_{i=1}^{n} H_0(i \| T_i)$ as g^y then the value of $\prod_{i=1}^{n} H_0(i \| T_i)^{st}$ is equivalent to g^{xy}. Based on the hardness of computation Diffie-Hellman assumption, the advantage of adversary for calculation of $\prod_{i=1}^{n} H_0(i \| T_i)^{st}$ is negligible defined as ϵ_{cdh}.

Therefore, we conclude that the advantage of adversary for breaking the challenge ciphertext is $\epsilon_{lbdh} + \epsilon_{cdh}$. □

3.7 Performance Analysis of DOSE and RASE

Our both proposed schemes provide searching over Anonymously encrypted data with fine-grained access control. The scheme DOSE is used to retrieve the subset of encrypted documents which are uploaded by a specific user and whose access policy is satisfied by the user's attributes. It does not support keyword based searching. To perform the search operation, the CS has to search for the access policy from the index which has been satisfied with the user's attributes. No mathematical operations are required on CS side.

The RASE scheme provides the search for both keyword and the matching access policy without compromising receiver anonymity and keyword secrecy. The RASE scheme requires only constant no. of bilinear pairing operations on CS side, which effectively reduces the time of search operation. The performance comparison of both the schemes is shown in below table.

Scheme	Wang *et al.*'s Scheme [76]	DOSE [80]	RASE
Access Policy Structure	AND-gate on Multi-valued Attributes	Tree Structure	AND-gate on Multi-valued attributes
Type of Search	Keyword based	Data Owner based	Keyword based
Type of pairing	Multi-linear	Bilinear	Multi-linear
Trapdoor Generation Time Complexity	$T_P + 2T_E$	$(n)T_P$	$4T_E + T_H$
Search Operation Time Complexity	$3T_P + (n)T_M$	Lookup	$4T_P + 2T_M$
Preserving Receiver Anonymity during Search Operation	No	Yes	Yes

Table 3.2: Comparison of Properties of Wang *et al.*'s scheme [76],DOSE and RASE. n: Number of attribute possessed by a user; T_P : time of pairing T_M,T_E and T_H denotes the time of pairing operation, multiplication, exponentiation operation and hash operation

The performance of DOSE on real time system is equivalent to any other searchable encryption scheme [28, 29], where the CS has to perform only lookup and retrieve operations. Because, no mathematical operations are required on CS side.

We have implemented the RASE scheme using the JPBC library [84] on a linux machine with Intel i5 processor and 8 GB RAM. The JPBC library provides an implementation of the Multilinear pairing over integers as presented by Coron *et al.* in [82]. For the implementation, the 165 primes, each of 757 bit long are used. We have simulated the search operation for 100 to 1000 encrypted indexes, considering each index is having 4 keywords. In Figure 3.3, we have shown the search computation time for a single encrypted index with varying number of attributes. The graph shows that the search computation remains constant for an index irrespective to the number of attributes in the system. Next, we simulate the scheme operations to derive the computation time for searching from a bunch of encrypted indexes. For this experiment, we fixed the total number of attributes (n) as 10. However, it should be noted that the number of attributes does not affect the search operation time in the proposed RASE scheme. The computational

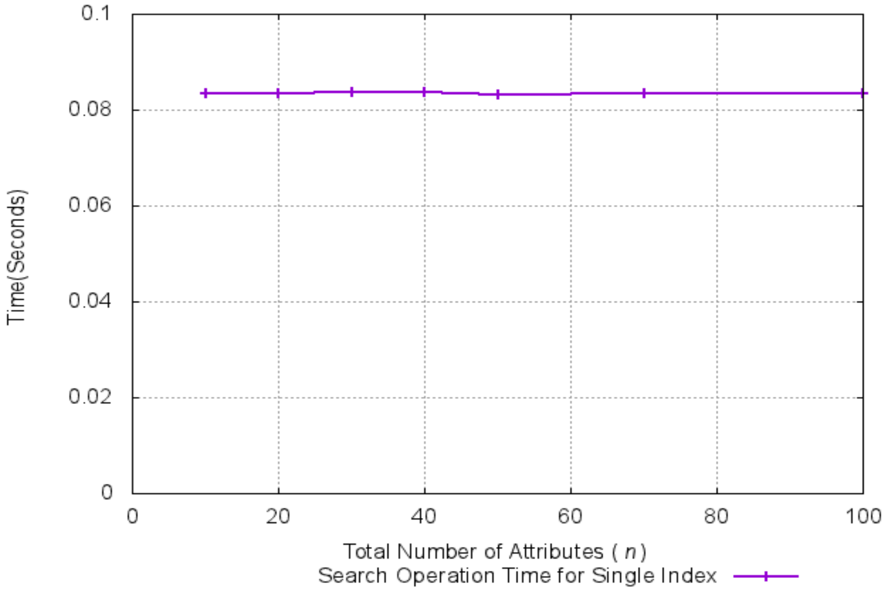


Figure 3.3: Computation Time of Searching a single index in the RASE scheme

cost is measured on a Google cloud instance of type N1 series with 8 vCPUs. The results we obtained for the search operation time for the indexes from 100 to 1000 are given in the figure 3.4, where the results are captured from average of 10 experiments of each case.

3.8 Conclusion

For an efficient and secure searchable encryption,the leakage of search and access pattern is acceptable provided if the keyword secrecy is preserved. For providing searchable encryption with fine-grained access control in multi-sender-multi-receiver scenario, ABE is an appropriate technique. However, when using the ABE technique as a base of a searchable encryption technique, it is highly required to hide the access policy inside the ciphertext and thereby provide receiver anonymity. Because the access policy in clear form may help an adversary to break the keyword (ciphertext) secrecy. There are some existing attribute based searchable encryption techniques which claims to provide receiver anonymity. When analyzing those schemes we have found that they are having either security flaws or performance barriers. We have suggested two searchable encryption schemes with fine-grained access control. One of those scheme denoted as

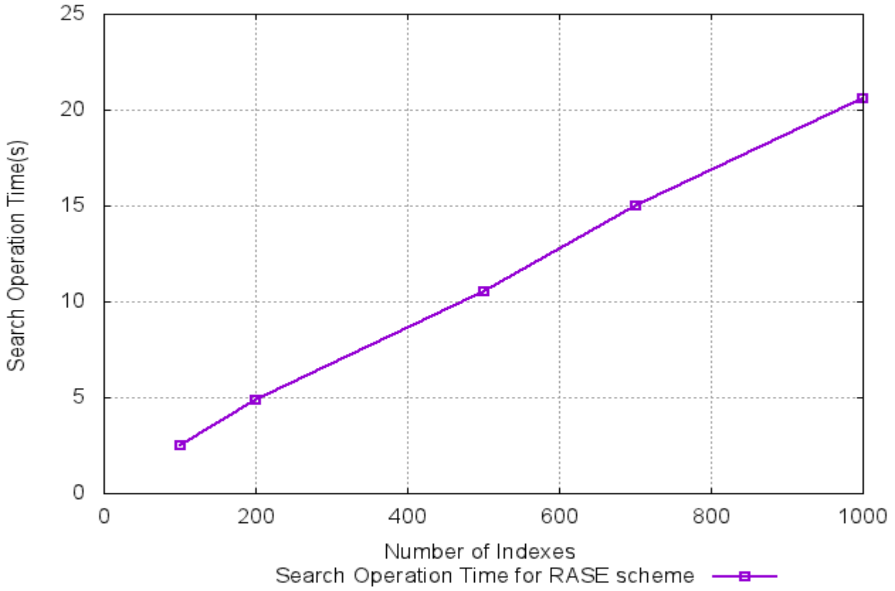


Figure 3.4: Computation time of Search operation in the RASE scheme

Data Owner based Searchable Encryption (DOSE) provides Data Owner Based Searching and does not support keyword based searching. The second scheme defined as Receiver Anonymous Searchable Encryption (RASE) scheme facilitates keyword based searching with receiver anonymity. The design of RASE scheme supports efficient search operation with constant number of mathematical operations irrespective of the number of attributes. However, the access policy of encrypted index in RASE can include only one value per attribute. The encryptor can not place more than one value for an attribute in the access policy.

CHAPTER 4

Privacy preserving Searchable Encryption (PSE)

4.1 Background

As discussed in chapter 3, searchable ABE schemes facilitates searching with fine-grained access control over encrypted data. At the same time it is also required to hinder the access policy in ciphertext to provide receiver anonymity. Many schemes have provided their construction for searchable ABE, but only few of them have claimed to provide receiver anonymity. In the previous chapter we have analyzed that the existing searchable ABE schemes with receiver anonymity have either security flaws or performance bottleneck issues. Koo *et al.* have suggested a searchable ABE scheme with receiver anonymity in [75]. But the scheme allows an adversary such as Cloud Server(CS) to learn the receiver information. We have proposed a scheme DOSE that facilitates a user to retrieve a subset of documents accessible to the user without revealing his attributes. But, the scheme supports only data owner based searching and does not provide keyword based searching. Shi *et al.* have provided a keyword based searching in [74] with receiver anonymity denoted as AS-PKE. The AS-PKE scheme is less efficient because of the use of composite order bilinear group. Also, the scheme supports at most only one value to be placed in the access policy of an encrypted index. Wang *et al.* have also suggested a keyword searchable encryption scheme with receiver anonymity which gains the performance efficiency in search operation. We have found the security flaw in it and proposed a new construction denoted as RASE. Like the scheme AS-PKE, the RASE scheme supports only one value per attribute to be placed in the encrypted index' access policy. Also, the schemes of [74] and

RASE do not support don't care attributes in the access policy.

In this chapter we provide a privacy preserving searchable encryption scheme that facilitates keyword based searching over attribute based encrypted data with hidden access policy (receiver anonymity) and which supports multiple values for an attribute to be included in the access policy.

4.2 The Proposed Scheme

We present a Privacy preserving single keyword based Searchable Encryption scheme (PSE) with fine-grained access control. Our scheme provides a keyword based search facility over attribute based encrypted data with hidden access policy. In PSE, a trusted authority verifies the user's attributes and assigns him a secret key. One of the key feature of this scheme is that once obtained the secret key, the user can generate the search query himself in form of a trapdoor, using the secret key assigned to him. This is more practical when compared to the scheme of [74], where each time of making a search, the user needs to obtain the search token from a trusted authority. The trapdoor generated by user do not reveal the keyword being queried or the user's attributes. Each data owner prepares an encrypted index of keywords for his document with hidden access policy.In the access policy the data owner can add multiple values for an attribute. The access policy also supports don't care attributes. To encrypt the index, the data owner needs to get assistance from a trusted authority. This feature was added to make the scheme adaptively secure against chosen keyword attack. The assistance do not require the data owner to disclose the data and receiver information to the trusted authority.

The cloud server performs the search operation with the input of trapdoor and encrypted index. The search operation returns true if (1) the keyword inside the trapdoor is included in the index, and (2) the access policy of ciphertext is fulfilled with user's attributes. Despite having access to encrypted index and trapdoor, the cloud server can only learn the information whether the search is successful or not. Both the keyword and receiver access policy remains secured during the search operation. Unlike the scheme in [74], we have proved the scheme to be adaptively secure in random oracle model against chosen keyword attack.

We have compared the various parameters of PSE with the scheme of [74] and our another proposed scheme-RASE. Finally we implemented all the operations of PSE. The search operation we have tested on google cloud instance and its results are included in the section 4.4. The results prove the feasibility of the scheme and also show that the performance of PSE is better with moderate number of attribute values.

4.2.1 Design Goals

Functional Goals:
Each user gets a secret key from the Attribute Center as per his attributes. After retrieving the secret key, the user should be able to generate the search query himself. The search query enables a cloud server to conduct the search operation correctly over the encrypted index. The search operation should return true if (1)the keyword inside the trapdoor is included in the index and (2) the attributes of user satisfies the access policy of an encrypted index.

Security Goals:
The encrypted index contents should not reveal the contents inside the index and the access policy of the index. In a similar fashion, the trapdoor preserves the security of keyword and the user's attributes.Without a valid secret key the trapdoor can not be generated. Other than the search operation outcome, the cloud can not learn anything from the search operation such as the keyword inside the encrypted index or trapdoor, user's attributes or access policy of an encrypted index.

4.2.2 System Model

To make the PSE scheme adaptively secure against chosen keyword attack, we have customized the basic system model which was presented in section 2.3 for this particular scheme. We have added one more trusted entity on user side, denoted as *Token Generator*. Now the system model for PSE comprises the following entities.

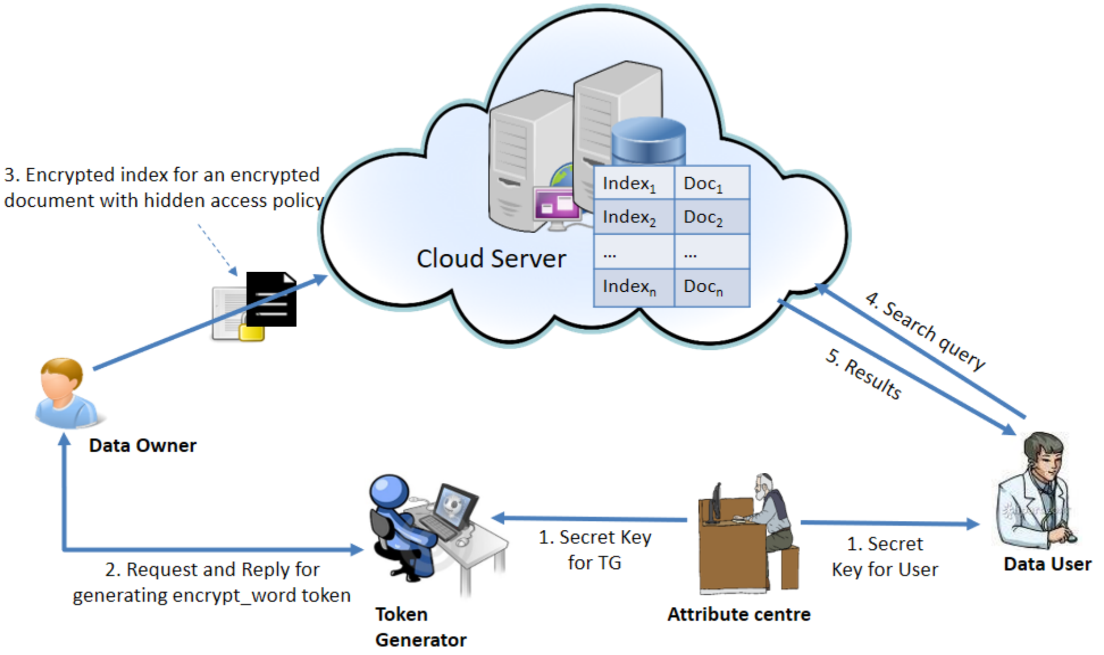

3. Encrypted index for an encrypted document with hidden access policy
Cloud Server
Index_1 Doc_1
Index_2 Doc_2
... ...
Index_n Doc_n
4. Search query
5. Results
Data Owner
Data User
2. Request and Reply for generating encrypt_word token
Token Generator
1. Secret Key for TG
Attribute centre
1. Secret Key for User

Figure 4.1: System Model for PSE

- *AC*: The *AC* is a trusted third party of the system. It is responsible for generating system parameters and issuing keys to users of the system.

- *Token Generator*: The Token Generator (TG) is also a trusted part of the system which assists a data owner for generating encrypted index. To achieve the security against chosen keyword attack, TG is involved in the process of generating encrypted index. For a small system/organization the AC itself can play the roll of TG. But in case of a system with sufficiently large number of users, autonomous entities can be established, which performs the roll of TGs. The TGs are assigned a set of derived master secret key from AC.

- *Data Owner*: Data owners encrypt and store the data on cloud. The encrypted data consists of two parts. (i) the index of encrypted keywords for which the document should be searched for and (ii) the encrypted document.

- *CS*: The *CS* is outside of the system boundaries and it provides storage and computation services for the entities of the system.

- *Receiver User*: Receiver user generates and submits a trapdoor to *CS*. The *CS* searches over the encrypted indexes using this trapdoor. The documents corresponding to the indexes for which the search operation returns true are returned to the user. Finally, the user decrypts the resultant documents.

4.2.3 Role of Token Generator

The data owner executes the function of creating encrypted index with the help of *TG*. The involvement of *TG* is for inclusion of the master secret key parameters inside the ciphertext components of encrypted index. The *TG* helps to achieve following goals.

- To make the scheme adaptively secure against chosen keyword attack it is necessary to protect the keywords inside the index or in search query with the master secret key components. If *TG* is placed on data receiver side, to include the master secret key components in search query, then the search response time will be increased for an end-user, because the user has to interact with trusted authority every time before making a search operation. Therefore, the roll of *TG* on data owner side helps to achieve both the goal of keyword secrecy and effective response time. Also, in real life data is encrypted and uploaded once, but search operation is conducted many times on that data.

- The placement of *TG* on data owner side, limits the amount of data uploaded on cloud storage system. Any user who has knowledge of public key parameters requires the help of *TG* to encrypt the index contents before uploading them on the cloud storage media. This is beneficial especially to protect against file-injection attacks [39]. In File injection attack, with the use of public parameters, an adversary such as *CS* constructs an encrypted index for some chosen keywords and chosen access policies. Whenever a data query is fired from the user side in form of trapdoor the *CS* runs the search algorithm on each real and fake encrypted indexes. From the results of search operation over this fake encrypted indexes and other real encrypted index give the user information about which keyword and access policies are hidden inside the trapdoor and other encrypted indexes.

4.2.4 Scheme Definition

Definition 14. *The PSE scheme is a 5-tuple scheme defined as below:*

Setup(1^l)→*(MSK,PK,TSK): The Setup algorithm is run by AC. It takes as input parameter a security parameter l and outputs the master secret key MSK, TG's secret key TSK and public key PK.*

KeyGen(MSK, L)→(SK_L)*: It is a randomized algorithm and it is run by the AC. The algorithm takes as input the master secret key MSK along with a set of attributes L of a user. It outputs a secret key* SK_L *for that user. The key* SK_L *is used to generate a trapdoor for performing search operation.*

Encrypt_Index(PK, W, T, TSK)→(CT_W)*: This algorithm is run together by the data owner and TG . W is the set of keywords associated with document M. Data owner starts the computation to generate the encryption for each keyword w included in keyword set W. The data owner gets an encrypted token for each w, from TG to perform the encryption of keyword. To generate encrypted word tokens, TG uses his secret key TSK. At the end of this phase, the data owner outputs a set of encrypted words* CT_W*, also known as encrypted index for document M.*

Trapdoor(PK, SK_L, w)→(tw)*: Receiver user invokes this randomized algorithm to make a trapdoor for retrieving the documents from CS whose associated index contains an encrypted entry for the word w and for which he possess the sufficient access rights. The algorithm outputs a trapdoor tw generated for w.*

Search(tw, CT_W)→(true/false)*: It is a deterministic algorithm and run by CS. The*

Search algorithm takes as input the trapdoor tw sent by the user and encrypted index CT_W. The algorithm returns true if the word in tw matches with any of the keyword included in CT_W and user's key satisfies the access policy of CT_W.

4.2.5 Security Model

The security model for the proposed scheme is IND-CP-CKA as described in chapter 2. In the security model, we assume that the cloud as an adversary does not possess a valid secret key. The goals of an adversary are listed below.

- The Adversary can retrieve the information about underlying access policy.
- Adversary can learn the information about the word being search for.

In theorem 4.1, we will show that the PSE scheme is secure in IND-CP-CKA (Indistinguishability against ciphertext policy and chosen keyword attack) model. The IND-CP-CKA model requires that if an adversary is given access to an encrypted index and a trapdoor, then he can only perform the search operation, but can not learn (1) the keywords inside and the access policy of an encrypted index, (2) the keyword inside the trapdoor and the user's attributes.

4.2.6 Detailed Construction

Setup(1^l)→(MSK,PK,TSK): Attribute Center chooses a security parameter l, and performs the following steps to generate system keys and public parameters.

- choose two multiplicative cyclic groups G_1 and G_2 with a prime order p where length of p is determined by the security parameter l.

  select g_1, g_2 as two generators of group G_1 and define a bilinear mapping e:$G_1 \times G_1 \rightarrow G_2$.
- choose two collision resistant hash functions H_1: $\{0,1\}^* \rightarrow \mathbb{Z}_p$ and H_0: $\{0,1\}^* \rightarrow G_1$.
- choose $\sum_{i=1}^{n} m_i$+3 random exponents $\{\alpha, \beta, \gamma, \{r_{i1}, r_{i2}, \cdots, r_{im_i}\}_{1\leq i\leq n}\}$ from $\mathbb{Z}_p$. These elements serve as the master secret key MSK of the system.

- Next the AC computes TSK = $\langle$ $\{\{\frac{r_{ij}}{\gamma}\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$ $\rangle$ and assigns these values to TG.

- publish the public key as PK = $\langle$ g_1, g_2, $e(g_1,g_2)^\alpha$, $g_2^{\frac{\alpha}{\beta}}$, $g_1^{\frac{\alpha}{\gamma}}$, g_2^γ, $\{g_2^{r_{i1}}$, $g_2^{r_{i2}}$, $\cdots$, $g_2^{r_{im_i}}\}_{1\leq i\leq n}$ $\rangle$.

KeyGen(MSK, L)$\rightarrow$(SK_L): The user submits his set of attribute values to the AC. The AC runs this algorithm and generates a secret key for the user. This secret key is used to generate the trapdoor for conducting a search operation over encrypted data. Let the user possess j^{th} value $v_{i,j}$ for an attribute i, 1$\leq$ i $\leq$ n. The AC chooses a random numbers r from $\mathbb{Z}_p$ and generates the search key as follows.

D_0 = $g_1^{r\beta}$
$\{D_{i1}$=$g_1^{(H_1(i\|v_{i,j})+r)\frac{\alpha}{r_{i1}}}$, D_{i2}=$g_1^{(H_1(i\|v_{i,j})^2+r)\frac{\alpha}{r_{i2}}}$,
$\cdots$, D_{im_i}=$g_1^{(H_1(i\|v_{i,j})^{m_i}+r)\frac{\alpha}{r_{im_i}}}\}$ $_{1\leq i\leq n}$ ($v_{i,j}$ $\in$ L).

The output of the algorithm is the secret key SK_L = $\langle$ D_0, $\{\{D_{ij}\}$ $_{1\leq j\leq m_i}\}$ $_{1\leq i\leq n}\rangle$

Encrypt_Index(PK, W, T, TSK)$\rightarrow$(CT_W): This algorithm is run collectively by the data owner and TG. It takes as input keyword set W associated with a document M, PK and T from data owner side and TSK from TG. Data owner computes following.

- chooses a random secret values s from $\mathbb{Z}_p$. Then randomly picks s_1, s_2, $\cdots$, s_{n-1} from $\mathbb{Z}_p$ and calculates s_n = s - $\sum_{i=1}^{n-1}$ s_i.

  For every attribute field i choose a_i' from Z_p for $1\leq i\leq n$ and then compute $f(x_i)$ = $a_i'(x_i - H_1(i\|\hat{v}_{i,1}))(x_i - H_1(i\|\hat{v}_{i,2}))\cdots$ $(x_i - H_1(i\|\hat{v}_{i,m_i}))$ + s_i, where $\hat{v}_{i,j}$=$v_{i,j}$(j^{th} value of attribute i) if $v_{i,j}$ $\in$ T_i; else, it will be a random value. If an attribute λ is a don't care for the current access policy, then include all the values from the valueset V_λ in the equation. The resultant equation is

$$f(x_i) = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots + a_{im_i}x^{m_i} \tag{4.1}$$

Summation of all coefficients except a_{i0} from all equations is calculated as $A_1 = \sum_{i=1}^{n} (\sum_{j=1}^{m_i} a_{ij})$.

- To perform the encryption of each of the keyword $w \in W$, the data owner picks a random value ϑ from $\mathbb{Z}_p$ and computes $\{\{H_0(w)^{a_{ij}\vartheta}\}_{1 \leq j \leq m_i}\}_{1 \leq i \leq n}$. These values are sent to the TG for requesting the encryption token for keyword w. The TG returns an encrypted token as $\prod_{i=1}^{n} (\prod_{j=1}^{m_i} H_0(w)^{\frac{a_{ij} r_j \vartheta}{\gamma}}) = H_0(w)^{\frac{\sum_{i=1}^{n} (\sum_{j=1}^{m_i} a_{ij} r_j) \vartheta}{\gamma}}$. From this encrypted token the data owner remove the value of ϑ. The use of ϑ is to hide the coefficient values when the communication takes place between data owner and TG. Let's denote the term $\sum_{i=1}^{n} \sum_{j=1}^{m_i} a_{ij} r_{ij} = A_2$.

- The data owner now computes the encryption of keywords as $\{C_w = g_1^{\frac{(s - \sum a_{i0})\alpha}{\gamma}} \cdot H_0(w)^{\frac{A_2}{\gamma}}\}$ for each $w \in W$, $C_1 = g_2^{\frac{A_1 \alpha}{\beta}}$, $\{C_{i1} = \mathrm{g}_2^{a_{i1} r_{i1}}$, $C_{i2} = \mathrm{g}_2^{a_{i2} r_{i2}}$, $\cdots$, C_{im_i}=$\mathrm{g}_2^{a_{im_i} r_{im_i}}\}$
  for $1 \leq i \leq n$.

The final output generated as encrypted index is $CT_W = \langle\ \{C_w\}_{\forall w \in W}, C_1, \{\{C_{ij}\}_{1 \leq j \leq m_i}\}_{1 \leq i \leq n}\ \rangle$.

Trapdoor(PK, SK_L, w)→(tw): In order to search for the documents having the keyword w, a user picks a random value ψ from $\mathbb{Z}$ and prepares a trapdoor for search operation using secret key components.

- $D'_0 = D_0^{\psi} = g_1^{r\beta\psi}, D'_1 = g_2^{\psi\gamma}$
  $\{D'_{i1} = D_{i1}^{\psi} = g_1^{(H_1(i \| v_{i,j}) + r)\frac{\alpha\psi}{r_{i1}}} H_0(w)^{\psi},$
  $D'_{i2} = D_{i2}^{\psi} = g_1^{(H_1(i \| v_{i,j})^2 + r)\frac{\alpha\psi}{r_{i2}}} H_0(w)^{\psi}, \cdots$
  $D'_{im_i} = D_{im_i}^{\psi} = g_1^{(H_1(i \| v_{i,j})^{m_i} + r)\frac{\alpha\psi}{r_{im_i}}} H_0(w)^{\psi}\}_{1 \leq i \leq n}$.

The outputs of Trapdoor(PK, SK_L, w) is the trapdoor $tw = \langle\ D'_0, D'_1, \{\{D'_{ij}\}_{1 \leq j \leq m_i}\}_{1 \leq i \leq n}\rangle$, which is used in the Search algorithm.

Search(tw, CT_W)→(true/false): After receiving a trapdoor tw, the CS initiates following procedure with each encrypted index (CT_W) to find a match.

$$\begin{aligned}
R_{s1} = &\quad \prod_{i=1}^{n}\prod_{j=1}^{m_i} e(C_{ij}, D'_{ij}) && (4.2)\\
R_{s2} = &\quad e(C_1, D'_0) = e(g_1, g_2)^{A_1 \alpha r \psi}\\
R_s = &\quad \frac{R_{s1}}{R_{s2}}\\
= &\quad e(g_1, g_2)^{(s-\sum a_{i0})\alpha\psi} \cdot e(H_0(w), g_2)^{A_2\psi} && (4.3)
\end{aligned}$$

The correctness of (4.2) is given below.

$$\begin{aligned}
R_{s1} = &\quad \prod_{i=1}^{n}\prod_{j=1}^{m_i} e(g_1^{(H_1(i\|v_i)^j+r)\frac{\alpha\psi}{r_{ij}}} H_0(w)^{\psi}, g_2^{a_{ij}r_{ij}})\\
= &\quad (g_1, g_2)^{\sum_{i=1}^{n}(s_i - a_{i0})\alpha\psi} \cdot e(g_1, g_2)^{A_1\alpha r\psi} \cdot e(H_0(w), g_2)^{A_2\psi}\\
= &\quad e(g_1, g_2)^{(s-\sum(a_{i0}))\alpha\psi} \cdot e(g_1, g_2)^{A_1\alpha r\psi} \cdot e(H_0(w), g_2)^{A_2\psi}
\end{aligned}$$

Next the CS computes for each encrypted keyword C_w included in CT_W as.

$$e(C_w, D'_1) = e(g_1^{\frac{(s-\sum a_{i0})\alpha}{\gamma}} \cdot H_0(w)^{\frac{A_2}{\gamma}}, g_2^{\psi\gamma}) \qquad (4.4)$$

If the resultant value of (4.3) is equal to value of (4.4) for any C_w, then the algorithm returns true; else false.

4.3 Security Analysis

As shown in the security model, we assume that the adversary has not gain a valid trapdoor token whose attributes can satisfy the challenge access structure. The following theorem proves the security of PSE.

Theorem 4.1. *The proposed scheme is IND-CP-CKA secure under DBDH assumption if there is no polynomial time adversary who can win the game, with non-negligible advantage $Adv_A(l)$ in terms of security parameter l.*

- *Proof.* As defined in the security model in Section 2.4.3, the adversary submits two access policies T_0', T_1' and two keyword sets W_0, W_1 such that $|W_0| = |W_1|$. The adversary gets the challenge ciphertext CT_{W_b} which is encryption of W_b with respect to T_b. To identify the value of b, $\mathcal{A}$ needs to identify the encryption of a word $w \in W_0$ and $w \notin W_1$(or a word $w \in W_1$ and $w \notin W_0$) To get the knowledge of encrypted word from the ciphertext CT_{W_b}, the adversary computes following for word w:

$$\begin{aligned} e(C_w, g_2^{\gamma}) &= e(g_1^{\frac{(s-\sum a'_{i0})\alpha}{\gamma}} \cdot H_0(w)^{\frac{A_2}{\gamma}}, g_2^{\gamma}) \\ &= e(g_1, g_2)^{(s-\sum a'_{i0})\alpha} \cdot e(H_0(w), g_2)^{A_2} \end{aligned} \quad (4.5)$$

Next, the adversary computes the value of

$$e(H_0(w'), \textstyle\prod_{i=1}^{n} \{\prod_{j=1}^{m_i} C_{ij}\}) = e(H_0(w'), g_2)^{A_2}$$

for a word w'. To find whether $w \overset{?}{=} w'$, the adversary is required to find the value of $e(g_1, g_2)^{\alpha(s-\sum_{i=1}^{n} a_{i0})}$ from the ciphertext components $\{\{C_{ij}\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$, and C_1. We prove that, the advantage of adversary in calculating the value of $e(g_1, g_2)^{\alpha(-\sum_{i=1}^{n} a_{i0})}$ without a valid trapdoor is negligible under the DBDH assumption. We denote the result of equation (4.5) as C_w'. In the following game, we provide $\mathcal{A}$ the values of C_w' instead of C_w and show that the advantage of adversary in calculating the value of $e(g_1, g_2)^{\alpha(s-\sum_{i=1}^{n} a_{i0})}$is negligible. Rest of the ciphertext components will be assigned as in the real scheme. The adversary is given as challenge to distinguish between $e(g_1, g_2)^{\alpha(s-\sum_{i=1}^{n} a_{i0})}$ and a random element of group G_2. If the adversary is able to fulfill the challenge with non-negligible advantage, then we can build a simulator $\mathcal{S}$ that can break the DBDH problem with non-negligible advantage. The DBDH challenger sets the group G_1 and G_2. Then the challenger flips a binary coin μ outside of $\mathcal{S}$ view. If $\mu = 0$ then the challenger sets $(g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g,g)^{abc})$. Else the challenger sets $(g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g,g)^z)$ for some random value $z \in \mathbb{Z}_p$. In the following game $\mathcal{S}$ plays the roll of $\mathcal{C}$.

Setup: $\mathcal{S}$ assumes $g_2 = B$ and $g_1 = A$. Rest of the master secret key components are chosen by $\mathcal{S}$ as in the original scheme. $\mathcal{S}$ calculates the PK with these chosen values and submit it to $\mathcal{A}$. Two random oracles $\mathcal{O}_{H0}$ and $\mathcal{O}_{H1}$ are defined to simulate the hash functions. $\mathcal{O}_{H0}$: {0,1}* $\rightarrow$ G_1 and $\mathcal{O}_{H1}$: {0,1}* $\rightarrow$ $\mathbb{Z}_p$ work as follows(LH_0 and LH_1 are list of pairs of (request, response) generated respectively for $\mathcal{O}_{H0}$ and $\mathcal{O}_{H1}$).

- To compute $H_0(w)$, $\mathcal{O}_{H0}$ searches in LH_0. If a tuple (w, h_0) already exists in the LH_0 , then h_0 is returned, else pick $h_0 \in_R G_1$, add (w, h_0) to LH_0 and return h_0.

- To generate $H_1(i\|v_{i,j})$, $\mathcal{O}_{H1}$ first makes a search in LH_1. If a tuple ($i\|v_{i,j}$, h_1) already exists in the LH_1 , then h_1 is returned, else pick $h_1 \in_R Z_p^*$, add ($i\|v_{i,j}$, h_0) to LH_1 and return h_1.

Phase 1: $\mathcal{A}$ issues adaptively generated queries to obtain trapdoors for keyword w and set of attribute values L. To generate a trapdoor tw, $\mathcal{S}$ first runs the *Key_Gen* algorithm to generate the secret key SK_L. To compute the secret value $\mathcal{S}$ makes queries to $\mathcal{O}_{H1}$ to obtain $H_1(i\|v_{i,j})$ for each value L_i = $v_{i,j}$ (1 $\leq i \leq n$ and 1 $\leq j \leq m_i$). $\mathcal{S}$ performs following computation to derive a secret key SK_L. {D$_0$ = g$_1^{r\beta}$ = $A^{r\beta}$,{{D_{ij} = $g_1^{(H_1(i||v_{i,j})^j+r)\frac{\alpha}{r_{ij}}}$ = $A^{(H_1(i||v_{i,j})^j+r)\frac{a}{r_{ij}}}$ }$_{1\leq j\leq m_i}$ }$_{1\leq i\leq n}$}. Next $\mathcal{S}$ runs *Trapdoor* algorithm with input PK, SK_L and w. To generate trapdoor tw, $\mathcal{S}$ fires a query to random oracle $\mathcal{O}_{H0}$ with input w and retrieve h_0 as a substitute for $H_0(w)$. Next, he fetches $\psi \in_R Z_p^*$ and then calculate value of tw as in the real scheme.

Challenge: $\mathcal{A}$ submits two pairs (W_0,T_0) and (W_1,T_1), where $|W_0|$ = $|W_1|$ and the trapdoor tw gained by $\mathcal{A}$ in Phase-1 should satisfy either both the challenge ciphertexts or none of them. Consider c as s - $\sum_{i=1}^{n} a_{i0}$ where s is the secret value used to encrypt the keyword. The simulator $\mathcal{S}$ flips a random coin $b \in$ {0,1}. With the outputs obtained from oracles $\mathcal{O}_{H0}$ and $\mathcal{O}_{H1}$, the simulator $\mathcal{S}$ computes the challenge ciphertext with following values.

- For 1 $\leq i \leq n$-1 select a_i', z_i and build the equations for each attribute category as follows :

$$f(x_i) = a_i'(x - H_1(i\|\hat{v}_{ii})) \cdots (x - H_1(i\|\hat{v}_{im_i})) + z_i \tag{4.6}$$

$$f(x_i) = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots a_{im_i}x^{m_i}$$

where in (4.6) $\hat{v_{ij}} = v_{ij}$ if $v_{ij} \in T_b$; else, if $v_{ij} \notin T_b$ then $\hat{v_{ij}}$ is some random value chosen from $\mathbb{Z}_p$ for $1 \leq j \leq m_i$.

$\mathcal{S}$ computes $C_{i1} = B^{a_{i1}r_1} = g_2^{a_{i1}r_1}$, $C_{i2} = B^{a_{i2}r_2} = g_2^{a_{i2}r_2}$, $\cdots$, $C_{im_i} = B^{a_{im_i}r_{m_i}} = g_2^{a_{im_i}r_{m_i}}$ for $1 \leq i \leq n$-1

- For the n^{th} attribute category choose a random value $a'_n \in \mathbb{Z}_p$ and compute the following equation

$$\begin{aligned} f(x_i) = \;& a'_n(x - H_1(\hat{v}_{ni})) \cdots (x - H_1(\hat{v}_{nm_n})) \\ = \;& \acute{a}_{n0} + a_{n1}x + a_{n2}x^2 + \cdots a_{nm_n}x^{m_n} \end{aligned}$$

  Note that $\hat{v_{nj}} = v_{nj}$ if $v_{nj} \in T_b$; else, $\hat{v_{nj}}$ is some random value chosen from $\mathbb{Z}_p$ for $1 \leq j \leq m_n$. The value of s_n will be considered as C - S - $\sum_{i=1}^{n-1} a_{i0}$ - $\acute{a}_{n0}$. Now, $\mathcal{S}$ computes $C_{n1} = B^{a_{n1}r_1} = g_2^{a_{n1}r_1}$, $C_{n2} = B^{a_{n2}r_2}$ =$g_2^{a_{n2}r_2}$, $\cdots$, $C_{nm_n} = B^{a_{nm_n}r_{m_n}}$ =$g_2^{a_{nm_n}r_{m_n}}$.

- Compute $C_1 = B^{\frac{A_1\alpha}{\beta}} = g_2^{\frac{A_1\alpha}{\beta}}$,
  where $A_1 = \sum_{i=1}^{n} (\sum_{j=1}^{m_i} a_{ij})$

- Compute $C'_{w_b} = Z^{\alpha} \cdot e(H_0(w_b), \prod_{i=1}^{n} \prod_{j=1}^{m_i} C_{ij})$ for each $w_b \in W_b$. This is valid, because as discussed before, without the correct tw, the adversary will try to discover the encrypted value of w_b with an attempt of recovering the value of $e(g_1, g_2)^{s - \sum_{i=1}^{n} a_{i0}}$ from rest of the ciphertext components.

Now, $\mathcal{S}$ gives ciphertext $CT_{W_b} = \langle$ {C'_{w_b}} for each $w_b \in W_b$, C_1, and {C_{i1}, C_{i2} , $\cdots$, C_{im_i}} for $1 \leq i \leq n$ $\rangle$.

Phase 2: $\mathcal{A}$ repeats the queries for keyword w and attribute values L, as it did in Phase 1 with the restrictions that either w is not included in any of W_0 and W_1 or F(L,T_0) = F(L,T_1) = 0.

Guess: $\mathcal{A}$ outputs a guess b' of b. If $b' = b$, then $\mathcal{S}$ outputs μ=1 to indicate that it was given a valid DBDH-tuple, else it outputs μ=0 to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about b, in turn, $Pr[b \neq b' | \mu = 0]$= $\frac{1}{2}$. As the simulator guesses μ'=0 when $b \neq b'$, $Pr[\mu = \mu' | \mu = 0]$ =

$\frac{1}{2}$. If μ = 1, then the adversary $\mathcal{A}$ is able to view a valid encryption of message with advantage $\epsilon_{dbdh}(l)$, a negligible quantity in security parameter l. Therefore, $Pr[b = b'|\mu = 1]$ = $\frac{1}{2}$ + $\epsilon_{dbdh}(l)$. Similarly, the simulator $\mathcal{S}$ guesses μ'=1 when $b = b'$, in turn, $Pr[\mu' = \mu|\mu = 1]$ = $\frac{1}{2}$ + $\epsilon_{dbdh}(l)$. The overall advantage of the simulator in DBDH game is $\frac{1}{2} \times Pr[\mu = \mu'|\mu = 0]$ +$\frac{1}{2} \times Pr[\mu = \mu'|\mu = 1]$ - $\frac{1}{2}$ = $\frac{1}{2} \times \frac{1}{2}$ + $\frac{1}{2} \times (\frac{1}{2}$ + $\epsilon_{dbdh}(l))$ - $\frac{1}{2}$ = $\frac{\epsilon_{dbdh}(l)}{2}$. Therefore, if the $\mathcal{A}$ has advantage $\epsilon_{dbdh}(l)$ in the above game instance, then we can build a simulator (S) which can break the DBDH problem with negligible quantity $\frac{\epsilon_{dbdh}(l)}{2}$. □

4.3.1 Leakage Analysis

The searchable encryption schemes have always the concern about the search and access pattern leakage. The search pattern denotes whether any two queries are for same keyword or not. The access pattern reveals the number of documents who have the same keyword. From any searchable encryption scheme, the minimum leakage which is acceptable is in the form of search outcome which reveals the search and access pattern [66, 39].

The proposed scheme only leaks the information, which an adversary can obtain from the outcome of a search. The outcome of a search includes whether two different trapdoors are generated for the same keyword or not, and the number of documents containing common keyword. Other than this search outcome, the proposed scheme do not provide any additional information to the adversary. We provide the following corollary to define the leakage analysis of proposed scheme.

Corollary 1. *Let the trapdoor tw_i has underlying keyword w_i and user attributes L_i. The encrypted index collection for which the search operation with input tw_i returns true is denoted as I_i. Given an adversary the collection of {(tw_i,I_i)} for $1 \leq i \leq q$ where q is polynomially bounded number, then the adversary can only reveal the information as, for which tw_i and tw_j ($1 \leq i, j \leq q$, $i \neq j$), I_i = I_j.*

Each tw is generated with the input of keyword w and user's secret key L, where L contains user's attribute values. The trapdoor algorithm is probabilistic and the output domain of trapdoor is G_1. The order of group G_1 is a large prime number p. For two inputs (w_i,L_i) and (w_j,L_j), if w_i=w_j and L_i=L_j then, the probability that tw_i = tw_j is $1/p$. This proves that for sufficiently large value of prime number p with bitlength l

$$\Pr[(i \neq j),(w_i = w_j),(L_i = L_j);(tw_i = tw_j)] \leq 1/2 + \epsilon(l)$$

where, $\epsilon(l)$ is a negligible quantity. Therefore, an adversary who has just a view of trapdoors can not learn any information about the keyword and user's attributes hidden inside the trapdoor.

Now let the adversary has access to trapdoors and encrypted index. As proved in the Theorem - 4.1, the scheme is secure against chosen keyword attack and indistinguishability in ciphertext policy. Therefore, the advantage of adversary in learning the keyword or the user's attributes from tw_i and tw_j is negligible. An adversary can only learn if tw_i and tw_j represent the same pair of keyword and user attribute set or not, from the result of a search operation. Therefore, other than search outcome, the adversary can not learn anything from the search operation.

4.4 Performance Analysis

In the Table 4.1 we compare our scheme with the scheme AS-PKE from [74] and our earlier scheme RASE that we provided in section 3.6.2. We are not considering the scheme of [76] as the scheme is having a security flaw which uncovers the access policy. We have also not considered the DOSE scheme as it is providing data owner based search, not keyword based search. We have considered the scheme provided by Shi *et al.* in [74] because it also provides keyword based searching and receiver anonymity. For ease of comparison we consider the parameters of AS-PKE scheme for making a single keyword search operation and where all attributes are involved in access policy.

In the scheme of AS-PKE, only one keyword from each keyword field can be included in an encrypted index. Also there must be predefined number of keyword fields because the setup of AS-PKE requires to set that many number of public parameters. This construction varies from our two proposed schemes for searchable ABE, where we have omitted the concept of keyword fields and each index contains a number of keywords extracted from the document. Therefore, to avoid any complexity while making the comparison, we have assumed that each encrypted index contains a single keyword. However, it should be noted that as the number of keywords are increased in AS-PKE, the number of public key parameters, ciphertext size and search operation complexity increases even though the search is conducted for a single keyword. While in our proposed two searchable ABE schemes, only the ciphertext size increases as the number of words are increased.

As shown in Table 4.1, the size and operation complexity of PSE scheme is more than AS-PKE. However, it should be noted that the AS-PKE scheme works for a finite number of keyword fields and a user can place a search query only for keywords which belongs to the predefined keyword fields in the system. In AS-PKE the user has to acquire the trapdoor from a trusted authority for making a search query. Therefore, the trapdoor generation process includes both the computation as well as communication overhead between the user and the trusted authority. Also, as discussed in section 3.6.1, the AS-PKE scheme has weaker security notions, because the user has to declare the fields of keywords which are included in the search query. This feature leaks the information to the adversary as for which type of keywords the user is making the search. One of the limitation of AS-PKE scheme is that at most one value per attribute can be placed in the ciphertext access policy. In our previous scheme RASE also, only one value per attribute can be placed in the access policy. In PSE scheme more than one value for each attribute can be placed inside the ciphertext access policy. This feature is shown with the size of T_i in the Table 4.1. Another limitation of AS-PKE is that if a data user wants to issue τ different search queries, then he has to communicate τ times with trusted third party, which creates a per-query communication overhead on user side. In PSE scheme and our earlier scheme RASE, once the user obtains the secret key from the trusted third party, then he can generate the search token independently without interacting with the trusted third party. This feature

Scheme	AS-PKE [74]	RASE	PSE
Access Policy Structure	LSSS	AND-gate on Multi-valued attributes	AND-gate on Multi-valued attributes.
Attribute values included in access policy	$0 \leq \lvert T_i \rvert \leq 1$	1	$1 \leq \lvert T_i \rvert \leq m$
Group Order	Composite	Prime	Prime
Type of pairing	Bilinear	Multi-linear	Bilinear
Universe of Keywords fields	bounded	Unbounded	Unbounded
Public Key Size	$(n+6)\lvert G_1 \rvert$	5	$(n * m + 5)\lvert G_1 \rvert$
Encrypted Index Size	$(2n+2)\lvert G_1 \rvert$	$3\lvert G_1 \rvert$ + $\lvert G_2 \rvert$	$(n * m + 1)\lvert G_1 \rvert$
Search token size	$(n+4)\lvert G_1 \rvert$	$3\lvert G_1 \rvert$ + $\lvert G_2 \rvert$	$(n * m + 2)\lvert G_1 \rvert$
Trapdoor Generation Time	$(2n + 6)(T_E + T_M)$ + T_{comm}	$4T_E$ + T_H	$(n * m + 3)T_E$ + $(n * m)T_M$
Search Operation Time Complexity	$(2n + 1)T_P$ + $(2n)T_E$ + $(3n)T_M$	$4T_P$ + $2T_M$	$(m * n + 2)(T_P + T_M)$
Communication Overhead for generating τ number of search tokens	$O(\tau)$	$O(1)$	$O(1)$
Type of information learned by adversary	Search outcome and field of keyword	Search Outcome	Search Outcome

Table 4.1: Comparison of PSE scheme with AS-PKE scheme presented in [74] and our earlier scheme RASE; n: Number of attribute fields; m = $max(|V_i|)_{1 \leq i \leq n}$ where V_i = value-set for attribute i; $T_i \subseteq V_i$.The values in T_i are included in the access policy; T_P, T_M,T_E denotes the time of pairing operation, multiplication and exponentiation operation in group G_2; T_{comm} defines the time for communication between user and trusted authority to output the trapdoor.

No. of Attribute Values	$\lvert PK\rvert$	$\lvert SK_L\rvert$	CT_W
9	4391	3447	4394
15	6586	5329	6278
25	9409	8441	9412
35	13174	11603	12554
50	17879	16308	17264

Table 4.2: Size of Public Key PK, User's Secret Key SK_LF, and Encrypted Index CT_W for varying number of total attribute values ($m * n$);

improves the search efficiency in PSE scheme.

4.4.1 Implementation Results

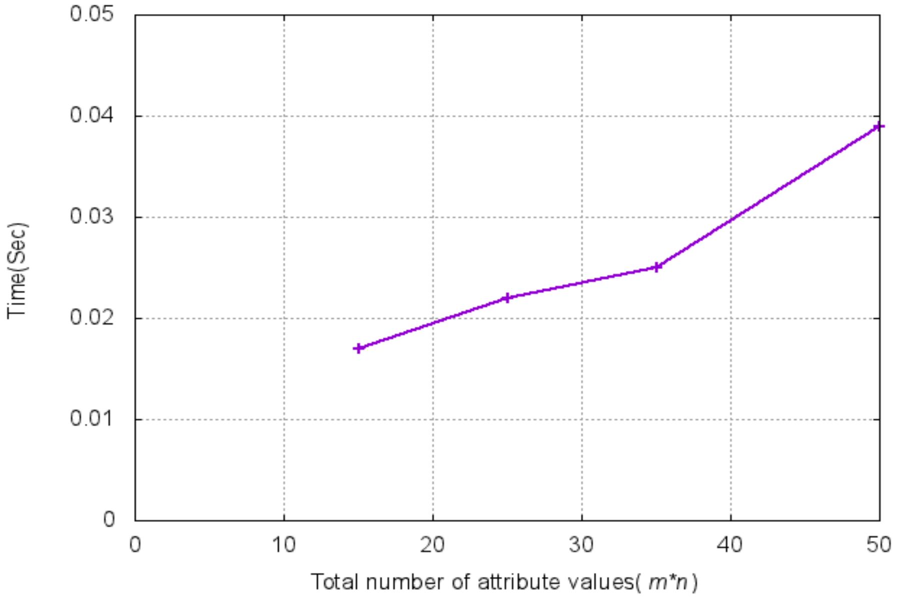


Figure 4.2: Time to generate an encrypted index for a record. (The record contains four keywords)

We have implemented the PSE scheme using Pairing Based Cryptography (pbc) library [86]. Bilinear pairings are constructed on the curve $y^2 = x^3 + x$ over the field F_q for some prime q=3 mod 4. The order p of the groups G_1 and G_2 is a prime number of size 160 bits, where the length of q is 512 bits. We evaluated the scheme with number of attributes n = 3, 5, 7 and 10. The scheme is also assessed with varying sizes of valuesets for the attributes. The performance of search operation is tested on the google cloud instance. The user side operations

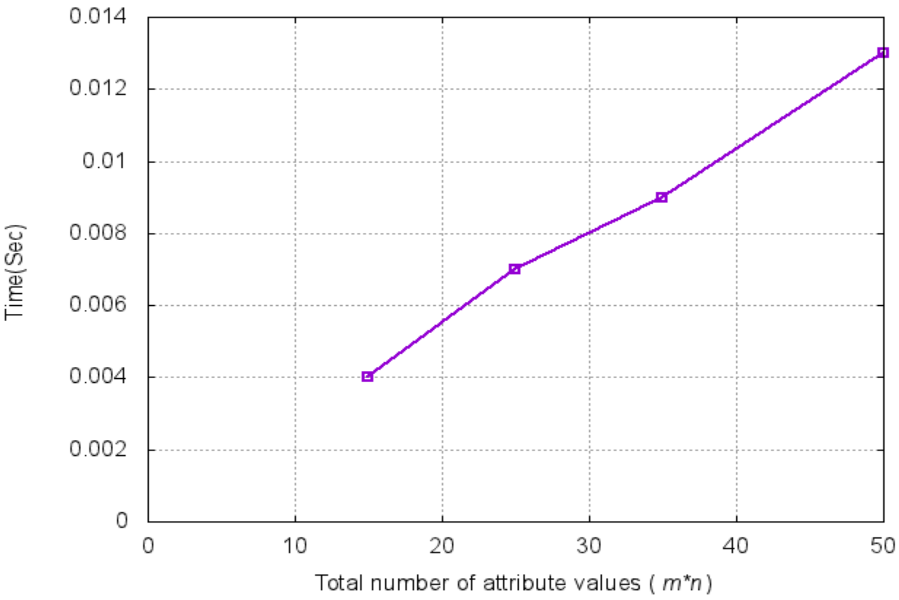


Figure 4.3: Trapdoor generation time in PSE.

are performed on a machine with 2.30 GHz Intel-i5 Processor configuration. We have shown here the results of the experiments with the valueset $m = 5$. The X-axis in the graphs shown in figures- 4.2,4.3 and, 4.4 represents the total number of attribute values ($m * n$) The experiments are performed on a data-set available from [87]. The data-set contains the diabetes patient's records. Each record contains four fields which represents the date of report, time of report (breakfast, lunch, dinner, bedtime), type of report(insulin dose, glucose level etc.) and value of report. We have taken these fields as keywords of that report for which the report should be searched for. Assuming that each record is related to a different patient, we have generated encrypted index for each record with a different secret key. Each encrypted index includes four keywords as discussed before.

The Setup, KeyGen, Trapdoor algorithms, and Encrypt_Index protocol are run on a linux system with Intel core-i5 processor running at 2.30 GHz and 8 GB RAM. The Setup and KeyGen algorithms are run by the AC and we are not showing the results of those algorithms. The timing results of Encrypt_Index protocol and Trapdoor algorithms are shown in figures 4.2 and 4.3. The size of public key parameters, trapdoor and an encrypted index are given in table 4.2 for varying size of total number of attribute values. It is apparent from the results that the perfor-

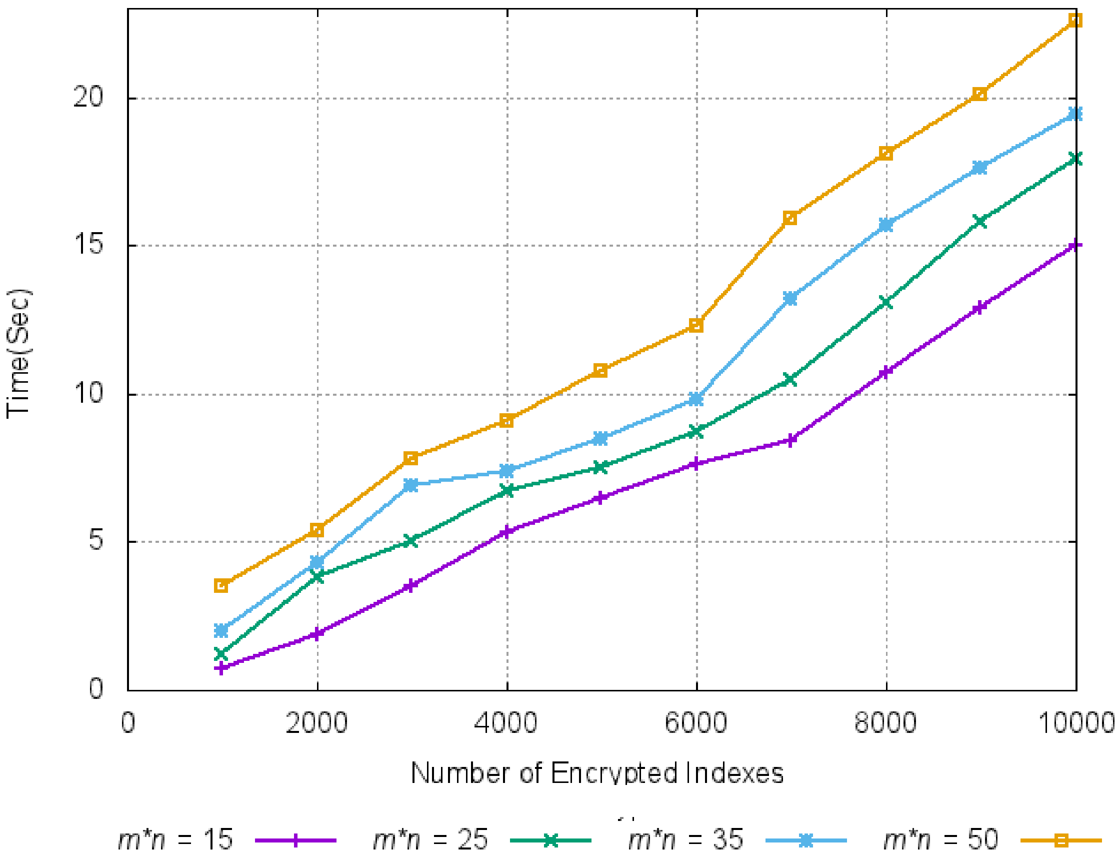


Figure 4.4: Time to make search operation over encrypted indexes on Google Cloud Computing Engine.

mance of the scheme operations linearly depends on the number of attributes.

The Search algorithm is tested on a Google cloud computing instance of n1 series with 16 virtual CPUs. Each virtual CPU is implemented as a single hardware hyper-thread on a 2.6 GHz Intel Xeon E5. To provide the inputs to the search algorithm, the results obtained from Encrypt_Index and Trapdoor algorithms are uploaded on the google cloud instance. In figure 4.4 we have shown the time to search over different number of encrypted index files with different number of attributes. As state before, each encrypted index is related to a different record and each index contains four keywords.

4.5 Conclusion

The proposed scheme PSE provides single keyword based search facility over anonymously attribute based encrypted data. The scheme has been proven adaptively secure against chosen keyword attack. To check the feasibility of the proposed scheme, the scheme has been implemented and tested. The performance results shows that the timing results of PSE are affected by the number of attributes and the valueset size of attributes. The performances of PSE scheme is affected by the number of attributes, because the computation cost increases linearly with the number of attributes and their size of valuesets. Therefore, we suggest that PSE scheme is best applicable for the system with limited number of attributes (such as 5) and their limited size of valuesets. It is feasible to apply PSE scheme for a system with large universe of attributes, but to get the optimal timing results in a large system the processor capability should be increased. With a powerful processor, the better search time can be obtained. The ongoing research in processor technology can help to obtain better timing results for search operation of PSE.

CHAPTER 5

Privacy preserving Attribute Based Signcryption (PASC)

5.1 Background

After retrieving the encrypted documents for which the search operation returns true, the user decrypts those documents. For efficiency in data utilization, it is required that the decryption operation should be cost-efficient. The existing Anonymous Attribute Based Encryption schemes suffer from the issue of costly decryption cost. We have analyzed the existing AABE schemes and proposed a solution which can be hooked with any existing AABE scheme to improve its decryption operation. We have also proposed an anonymous attribute based signcryption scheme. The anonymous attribute based signcryption is a cryptographic primitive that merges the advantage of Anonymous Attribute Based Encryption (AABE) and Attribute Based Signature (ABS) schemes. We have analyzed both the existing AABE and ABS schemes and also studied the existing Attribute based signcryption schemes.

5.1.1 Anonymous Attribute Based Encryption(AABE)

When using the public cloud storage services for storing the confidential data, the first requirement comes is storing the data in encrypted form, so that the cloud server (CS) can not learn about the data. ABE provides the confidentiality and fine-grained access control together. It also allows a data owner to target multiple recipient with a single ciphertext by embedding the required attributes of receiver in the access policy of ciphertext. To achieve the semantic security of data, the access policy should also be hidden from adversary $\mathcal{A}$. AABE schemes [45, 46, 47, 48] hides the access structure inside the ciphertext. As AABE hides legitimate receiver's identity and every user who receives a ciphertext may attempt to decrypt it believing that he is the intended recipient of the ciphertext. Therefore, One of the research issue related to AABE scheme is of performance improvement. To obtain the effective performance, the decryption operation in AABE scheme should be cost-efficient such that a recipient should not spend a significant cost for the decryption operation. In other words, the detection of wrong-person in wrong-ciphertext should be done with minimum use of computing resources.

The AABE scheme presented by Kapadia *et al* in [45] is not collusion-resistant and needs an online semi-trusted server that must know the attributes' values every user in the system has and re-encrypt ciphertexts for each user when the user retrieves the ciphertexts. Yu *et al.* have designed an AABE scheme. But their scheme supports AND gate on Single-valued Attributes. Later on the researchers have started to work on AABE scheme with access policy having "AND gate on Multivalued Attributes" [47, 48, 49].

Nishide *et al.* have presented two AABE schemes in [47]. Their first construction presents an AABE scheme which uses the symmetric bilinear pairing operation and has been proved selectively secure under the standard assumption. Their second scheme provides the construction for a flexible AABE scheme where the new attributes can be added in the system, even after the setup phase is over. Their second scheme is constructed with the use of asymmetric bilinear pairing operations and has been proven secure in the generic group model.

In [48], Li *et al.* have also presented an AABE scheme. They have addressed the issue of illegal key sharing. To thwart the illegal key sharing among the users, Li *et al.* have proposed the idea of user accountability. In their scheme, the user's key is embedded with user's identity. In case of any malfunctioning, a pirated device tests the key to disclose the user's identity to whom the key has been issued. The problem they have discussed is interesting from research point of view. However, the solution proposed by them is not efficient, because the decryption is going to be performed on user's end. A pirated device can not access the user's secret key without user's permission. In such scenario, it seems to be unrealistic that a pirate device takes the secret key from a malicious user and check the identity of user to whom the key has been issued.

The schemes [46, 47, 48] suffers from performance bottleneck on receiver side because the receiver is enforced to perform decryption operation that involves a number of bilinear pairing operations for every ciphertext he receives, whether or not he is the intended recipient. To address this problem, Zhang *et al* [49] proposed an approach called *Match-then-Decrypt*, where a receiver performs a matching operation on the received ciphertext using his key. If the match function succeeds, then the decryption operation is performed, else not. However, we have analyzed the scheme presented by Zhang *et al* and found that the scheme suffers from security flaws [88]. The ciphertext components computed for matching phase operation helps an $\mathcal{A}$ to uncover the whole access policy. We have discussed that security flaw and presented our proposed construction to perform Match operation in section 5.1.3.

In [50], Rao *et al.* have presented an AABE scheme which claims to provide a constant ciphertext length and an efficient decryption cost. However, it should be noted that the acccess policy in [50] supports only one value per attribute in the ciphertext. Also, their construction is based on composite bilinear group which is computationally less effective than prime order bilinear group.

5.1.2 Signcryption with Fine-grained access Control

To provide confidentiality and authentication, signcryption is an efficient approach for fulfilling both the security requirements at low operational cost. To provide confidentiality, authentication and fine-grained access control all together in one scheme, ABSC [36] is an performance effective approach. ABSC satisfy the goals of both the ABE and Attribute Based Signature(ABS). ABSC allows a data owner to signcrypt a document using his attribute-based signature key. Every user whose attribute values satisfy the access policy of the signcrypted document is able to decrypt the document and verify the signature. Many ABSC techniques have been developed with this objective.

In [36], Gagné *et al.* have presented a threshold ABSC scheme. In their scheme, there are fixed number of attribute values. For each attribute value a unique public parameter is set. Each user possesses a subset of this attribute values, for which he has been provided the signature key and decryption key. For signing a message, the encryptor can use either all or subset of his attributes. The size of user's secret key components is also linear to the number of attribute values possessed by the user. They have proven their scheme secure in the standard model. In this scheme the authors have chosen the approach of Encrypt then sign. There fore, the scheme does not provide signer's attribute privacy. Any outsider viewing the ciphertext and having the knowledge of public key parameters can verify the signer's attributes. This scheme is not preferable, when to provide the data privacy it is also necessary to hide the signer's attribute values.

Wang *et al.* have also presented an ABSC scheme in [89]. In their scheme, the access structure is constructed in form of Access Policy. For generating signature also, the access policy for signer's attributes is created. The secret shares of encryption and signature are embedded in ciphertext along with the user's signature key components. The authors of [89] have proved their scheme to be secure in Random Oracle model. In their scheme, the verification of signature can be done after a successful decryption. However, the access policy of signer is embedded in ciphertext in clear form. Therefore, the $\mathcal{A}$ can read the signer attribute information.

In [90], the Emura *et al.* have presented a scheme, with dynamic property. In their scheme, the access policy constructed from signer's attributes can be changed dynamically after signing the message. Their scheme facilitates that after signing a message with the old set of attributes, if the user's attribute values are changed, then the user does not have to resign the document with the new secret key generated from his updated attribute value set. The trusted authority takes care of this computation. Their scheme supports "AND gate on single-valued attributes with positive, negative and don't care values".

In [91], Wei *et al.* have presented an traceable ABSC scheme. The construction in their scheme makes use of composite bilinear group and it uses the access structure in form of Access tree. In the scheme, the user's signature key is generated from his attributes and his identity. However, the signer's identity can not be verified by the receiver. The receiver can only verify the sender's attributes. A signer's identity can only be verified by a trusted authority.

In [92], Pandit *et al.* have presented an strongly unforgeable ABSC. In the strongly unforgeable signcryption scheme the $\mathcal{A}$, when given a signcrypted message for M with respect to an attribute set L, can not regenerate a new signcrypted message for M with the same set of L. To provide this feature they have adopted the approach of dual signature. The first signature is constructed from signer's attributes. After then a second signature is constructed using a strong One Time Signature (OTS) method. They have constructed their scheme using composite bilinear group.

Liu *et al.* have proposed an ABSC scheme in [93] for secure sharing or personal health records in an e-healthcare organization system. In [93], the signer's attributes remains hidden. A receiver after decrypting a message can verify the signature of the message to check its integrity, but can not identify the signer's attributes. Hong *et al.* presents a scheme in [94], where the access structure is presented in form of LSSS structure to avoid the recursive calls to bilinear pairing operations in unsigncryption operation.

All the existing ABSC schemes are constructed to solve various functional and security objectives. However, none of them have addressed the issue of receiver anonymity. In all such schemes, the ciphertext access policy is placed in ciphertext in clear form. As discussed earlier, the receiver anonymity can not be neglected when we have to achieve the data secrecy. Therefore, we have developed an ABSC scheme with receiver anonymity. Also, we have identified that in ABSC schemes sender identification is necessary. In an ABSC a user signcrypts the message using his attributes. In a large organization, there can be multiple employees who possess the same set of attributes. In case of any malfunctioning, for example spreading a wrong information amongst other employees, the receiver can identify the attributes of sender who signcrypted and send the information. However, it will be difficult to find out the unique identity of the person from the given attributes. Wei *et al.* have addressed this issue, but in their scheme, only a trusted third party can verify the user's identity.

To target multiple recipient for a single ciphertext and to provide the unique identity of sender to the receivers, an alternative approach available in literature is Multi-receiver Identity Based Signcryption (IDSC) [95]. Multi-receiver IDSC schemes are a custom version of IDSC. The IDSC is the combination of Identity Based Encryption and Identity Based Signature techniques. As like in ABSC schemes, in IDSC also a trusted third party establishes the system parameters and it is responsible to issue the identity based secret key to each user. The data is encrypted with public key parameters of system and receiver's ID. A receiver who possess the ID listed in ciphertext is able to decrypt the ciphertext using his ID based secret key. To signcrypt a document the data owner uses his ID based signature key issued from trusted authority. Over the years, many IDSC schemes [95, 96, 97, 98, 99] have been proposed, which work for a single sender and a single receiver scenario. Subsequently, single sender and multiple receiver IDSC schemes [100, 101, 102] have been constructed. In multi-receiver IDSC schemes, a single ciphertext can be generated for multiple recipients.

In [100], Duan *et al.* have presented an multi-receiver IDSC scheme that provides unsigncryption cost with constant number of bilinear pairing operations. The ciphertext contains an embedded list of all user ids for which the ciphertext is generated. Therefore, as the number of recipients increases, the length of ciphertext also increases. Also, in [103] the scheme of [100] has been proven insecure. In [101], Ming *et al.* have presented a multi-receiver identity based signcryption scheme which they have proven secure in standard model. As like in scheme of [100], the length of ciphertext in scheme of [101] increases, as the number of recipients increases. Also, none of these schemes provides receiver anonymity. The identities of recipients are clearly listed along with the ciphertext.

In [102], Pang *et al.* have addressed the issue of receiver anonymity in multi-receiver IDSC. In their scheme, the ciphertext does not contain the list of valid recipient IDs. Instead, each user tries his ID based secret key with the ciphertext components and find if she is valid recipient or not. However, the problem of ciphertext linear to the number of recipients still not resolved.

All these multi-receiver identity based signcryption schemes can work in a cloud storage scenario where there are multiple data owners and multiple data receivers. However, their complexity increases as the number of recipient increases. Also, the data owner has to be aware about the identities of each user to whom he wants to send the message.

These analysis of existing ABSC and MIDSC have proved that the anonymous ABSC is a better approach to provide confidentiality, authentication and fine-grained access control all together. Unlike the IDSC schemes, the ciphertext in ABSC does not rely on the number of receivers of the ciphertext. The ciphertext length depends the number of attributes in the system. There can be a number of users in the system, but the universe of attributes for a system is fixed. Every data user whose attributes can satisfy the access policy of a ciphertext, is able to decrypt the ciphertext. There can be one such user or more than one such users. Along with these essential security properties satisfied by the ABSC scheme, we have also identified that it is an essential requirement for data authentication, to trace the unique sender identity after a successful decryption operation. As like in [91], our scheme also includes the idea of signer traceability by including the sender's unique identity in the signature portion. However, unlike the scheme of [91], in our scheme, the signer's unique identity can be verified by the receiver after a successful decryption only.

5.1.3 Analysis of Zhang *et al.*'s scheme [49]

5.1.3.1 Scheme Definition

Zhang *et al.* have proposed an anonymous CP-ABE scheme with four algorithms **Setup, KeyGen, Encrypt**, and **Decrypt**. The Setup and KeyGen phase are run by the Attribute Center (*AC*). The *AC* is responsible to setup the system parameters and generate the secret key for each user in the system. The encryption is performed by the data owner using the public key parameters and the access policy in form of AND gate on multi-valued attributes. The ciphertext contains cipher component for each attribute in the system. If the attribute is included in the access policy, then it will be a valid cipher components, else it will be a random element.

The decryption algorithm in Zhang's scheme consist of two phase. First phase is match operation which enables a user to test if his attributes are able to satisfy the ciphertext access policy. The match operation is cost-effective because it requires only 3 bilinear pairing operations. If the match operation returns true, then only the user will go for the costly decryption operation. We have identified that the cipher components in Zhang's scheme generated for Match operation leaks the access policy information and helps an adversary to break the receiver anonymity.

5.1.3.2 Security Model

We have analyzed that the Scheme proposed by Zhang *et al.* is insecure in IND-CP (Indistinguishability against Ciphertext Policy) Model, because the ciphertext components for matching phase operations reveal the hidden ciphertext policy.

IND-CP Model

We consider the IND-CP(Indistinguishability against ciphertext policy) model to analyze the modified scheme. This model is a customized model of IND-CP-CPA. In this model, we exclude the encryption and decryption of a message as they are not part of the proposed matching scheme. Therefore, the goal of adversary for proposed scheme is to identify the access policy hidden inside the matching phase. The modified scheme is simulated with the following security game.

Setup: The challenger $\mathcal{C}$ chooses l as a security parameter and chooses α at random from Z_p. $\mathcal{B}$ also defines a bilinear mapping function from $G_1 \times G_1 \rightarrow G_2$ and chooses two generators from G_1 as g_1, g_2. The master private key is α. The public parameters g_1, g_2, g_1^{α}, g_2^{β} and $e(g_1, g_2)^{\alpha}$ are sent to $\mathcal{A}$.

The game has following four steps.

Phase 1: With this phase $\mathcal{A}$ issues polynomially bounded number of queries and gathers following items from the challenger.

- Secret key SK_L for attribute set L.
- matching phase elements for different access policies T.

Challenge: The adversary $\mathcal{A}$ submits two challenge access policies T_0^* and T_1^* with the condition that for any secret key SK_L issued to $\mathcal{A}$ during Phase 1, F(T_0^*,L) = F(T_1^*,L). The challenger $\mathcal{C}$ randomly picks a bit b = 0 or 1 and submits the encrypted elements for matching phase using two random secret values s, t.

Phase 2: The adversary $\mathcal{A}$ is allowed to run a number of queries as done in phase 1 without violating the restrictions imposed during challenge phase.

Guess: The adversary $\mathcal{A}$ outputs a guess b'. $\mathcal{A}$ wins the game if $b' = b$.

The formal description of the IND-CP model is given below.

Let the Φ denote the cryptographic scheme with the tuples ⟨Setup, KeyGen, Encrypt and Decrypt⟩. In IND-CP model, the $\mathcal{A}$ issues two pairs of message and access policy as (m_0, T_0) and (m_1, T_1) where $|m_0|$ = $|m_1|$. A bit b is selected in random and accordingly the encryption of m_b with respect to access policy T_b is computed and given to $\mathcal{A}$. At last the $\mathcal{A}$ issues a bit-value b'. The $\mathcal{A}$ wins the game if b = b'.

```
$IND - CP_{\Phi}^{\mathcal{A}}(l)$

$(PK, MSK) \leftarrow_{\$} Setup(1^l)$
$(m_0, T_0^*)(m_1, T_1^*) \leftarrow_{\$} \mathcal{A}(PK)$
$(c_b) \leftarrow_{\$} \mathsf{Enc}(PK, m_b, T_b^*)$
$b' \leftarrow_{\$} \mathcal{A}(PK, c_b)$
return $b' = b$
```

Definition 15. *The proposed scheme is secure in IND-CP secure, if the advantage of adversary $\mathcal{A}$ as defined below is negligible.*

$$\mathsf{Adv}_{\Phi,\mathcal{A}}^{\mathrm{ind-cp}} l = \tfrac{1}{2} - \Pr\left[1 \leftarrow IND - CP_{\Phi}^{\mathcal{A}}(l)\right]$$

5.1.3.3 Security Analysis

In Zhang's scheme g_1 is an element of a bilinear group G_1 published as one of the public key parameter. Some of the cipher components generated for match operation are $\hat{C}_0 = g_1^{s'}$ and $\{\{C_{i,j,\Delta}\}_{1 \le j \le m_i}\}_{1 \le i \le n}$. If for an attribute i, its value $v_{i,j}$ is included in the access policy then $C_{i,j,\Delta} = H_0(i||v_{i,j})^{s'}$ else it will be a random value. Here H_0 is a hash function which maps an arbitrary boolean string to an element of a bilinear group G_1. We have identified that these cipher components uncovers the access policy information.

An adversary with the knowledge of public parameters, can use these cipher components to learn the access policy of the ciphertext. The adversary will check if the following equation holds true or not for an attribute value $v_{i,j}$.

$$e(\hat{C}_0, H_0(i||v_{i,j})) = e(C_{i,j,\Delta}, g_1)$$

If there are n attributes and each attribute is having at most m number of values, then the adversary needs to perform $2 * m * n$ pairing operation to uncover the whole access policy.

5.1.3.4 Outcome of the analysis

The decryption operation of ABE becomes costlier, when provided with hidden access policy. Zhang *et al.* have addressed the issue and provided an approach of match-then-decrypt that helps the user to reduce the decryption overhead. However, as shown in this section their construction fails to provide the receiver anonymity. We have analyzed, that the concept of "Match-then-decrypt" is advantageous to improve the performance of any existing AABE scheme. However, the construction provided by Zhang *et al.* for matching phase is found to be faulty. It gives an outcome that if there is a Matching Phase construction, which can be linked with any existing AABE scheme, then the performance of that AABE scheme can be much improved. We develop a new construction for the same approach and same motive of enhancing the performance of existing AABE schemes.

5.1.4 Proposed Construction for Performance Improvement of AABE

In this proposed scheme we provide an approach of matching phase procedure that can be incorporated with any existing Anonymous ABE schemes supporting the "AND gate on Multi-valued attributes"[88]. It is required that the parameters of the proposed construction should be isolated from the parameters of AABE schemes for which we want to improve the performance. Inclusion of matching phase in an AABE scheme will help the receiver to check whether an encrypted message is intended to him or not without performing the whole decryption procedure. We do not include the cipher components and key components required for encryption and decryption of a message as it depends on the AABE scheme used for encryption and decryption of message and they are isolated from the parameters of the proposed scheme.

Definition 16. *The proposed scheme for matching operations is defined as a tuple (Setup, KeyGen, Encrypt, Match) as follows.*

Setup$(1^l) \rightarrow (MSK,PK)$ *: The Setup algorithm takes as input a security parameter* l*. The* *output of this phase is MSK and PK.*

***KeyGen**(MSK, PK, L) →(SK_L): On input of an attribute list L, MSK and PK, the algorithm outputs user's secret key SK_L.*

***Encrypt**(PK, T)→(CT): The Encrypt algorithm takes as input the ciphertext access policy T required for decryption and PK. The output of this algorithm is cipher components CT, which are used for matching phase.*

***Match**(PK, SK_L, CT)→(True/False): The Match algorithm takes as input PK,SK_L, and CT, and returns whether the SK_L is matched with CT or not.*

5.1.4.1 Detailed Construction

The scheme works as follows.

- **Setup**(1^l)→ (MSK,PK): The Attribute Center (AC) performs the setup phase. It selects two groups G_1 and G_2 of prime order p and a bilinear mapping function $e : G_1 \times G_1 \to G_2$. The AC chooses two random generators g_1 and g_2 from group G_1 and one hash function H_0 is defined as H_0: $\{0,1\}^* \to G_1$. The master secret key MSK is chosen as $\left\langle \alpha, \beta \in_R Z_p \right\rangle$. The corresponding master public key $PK \left\langle g_1, g_2, g_1^{\alpha}, g_2^{\beta}, e(g_1, g_2)^{\alpha} \right\rangle$ is published.

- **KeyGen**(MSK,PK,L)→ (SK_L): Let L=[$L_1, L_2, \cdots, L_n$] be the attribute list for the user who requires a secret key. For every user in the system the AC picks a random value ρ and generates a user's secret key SK_L for performing the matching phase operation as follows.
  D = $(g_2 \prod_{i=1}^{n} H(i||v_{i,j}))^{\alpha} \cdot g_2^{\rho}$ where L_i=$v_{i,j}$.
  $\bar{D}$ = $g_1^{\frac{\rho}{\beta}}$

- **Encrypt**(PK,T)→ (CT): We provide the construction of cipher components for matching phase only. Hence, we have not included encryption of message. The algorithm takes the access policy T and public key PK as input. Here T =$\{T_1, T_2, \cdots T_n\}$ where T_i $\{1 \leq i \leq n\}$ is the set of values permissible for decryption. To prepare the cipher components for matching phase the encryptor takes secret values s, t and t' from $\mathbb{Z}_p$ and makes n portions of t as t_i such that $\sum_{i=1}^{n} t_i = t$. For attribute values in each attribute category T_i create the following cipher components.

- If $v_{i,j} \in T_i$ $\tilde{C_{i,j}}$ = $g_2^{t_i} H_0(i \| v_{i,j})^{st+t'}$
- If $v_{i,j} \notin T_i$ $\tilde{C_{i,j}}$ is a random value.

The other cipher components are $\hat{C}$ = g_1^{st},$\bar{C}$ = $g_2^{st\beta}$, $\check{C}$ = $g_1^{t'\alpha}$ and $C' = e(g_1, g_2)^{\alpha(s-1)t}$. The final output of this algorithm is ciphertext CT = $\langle \{\{\tilde{C_{i,j}}\}_{1 \leq j \leq m_i}\}_{1 \leq i \leq n}$, $\hat{C}$, $\bar{C}$, $\check{C}$, C' $\rangle$

- **Match**(PK,SK_L,CT)→(true/false): A user performs the matching phase before going for decryption algorithm.

  First the user prepares the value of $\prod_{i=1}^{n} H_0(i \| v_{i,j})$. This value can be computed by user offline also. The user checks if his attributes L satisfies access policy T or not by checking if following equality holds true. User collects the relevant $\tilde{C}_{i,j}$ from each attribute category denoted with variable i. The value of j in $\tilde{C}_{i,j}$ denotes the attribute value in the i^{th} category which a receiver possesses.

$$C' = \frac{e(\hat{C}, D) \cdot e(\check{C}, \prod_{i=1}^{n} H(i \| v_{i,j}))}{e(\prod_{i=1}^{n} \tilde{C}_{i,j}, g_1^{\alpha}) e(\bar{C}, \bar{D})}$$

  If the equality does not hold true, the decryption procedure is aborted; otherwise, the decryption procedure is initiated.

Correctness:

The correctness of the matching phase is as follows.

$$\begin{aligned}
&\frac{e(\hat{C}, D)e(\check{C}, \displaystyle\prod_{i=1}^{n} H(i\|v_{i,j}))}{e(\prod_{i=1}^{n} \tilde{C}_{i,j}, g_1^{\alpha})e(\bar{C}, \bar{D})}\\
=&\frac{e(g_1^{st}, (g_2 \prod_{i=1}^{n} H(i||v_{i,j}))^{\alpha} \cdot g_2^{\rho})e(g_1^{t'\alpha}, \displaystyle\prod_{i=1}^{n} H(i\|v_{i,j}))}{e(\prod_{i=1}^{n} (g_2^{t_i} H(i||v_{i,j})^{st+t'\alpha}), g_1^{\alpha})e(g_2^{st\beta}, g_1^{\frac{\rho}{\beta}})}\\
=&e(g_1, g_2)^{(\alpha(s-1))t}\\
=&C'
\end{aligned}$$

5.1.4.2 Security Analysis

We show that the proposed scheme is secure in the IND-CP model. In particular, we prove the security of the scheme on basis of hardness of D-Linear Assumption and Decisional Diffie-Hellman Assumption. We prove the security of proposed scheme in two theorems. In first theorem we prove that unless a valid decryption key is available, the adversary is not able to find a valid match. Hence in the first theorem we impose a restriction that adversary will not get a valid secret key which can satisfy either of W_0^* and W_1^*. In the second theorem we prove that receiver anonymity is preserved in the modified scheme. We show that even if the $\mathcal{A}$ is able to get a valid key and find the match successfully, he can not find out the underlying access policy. That is, who else are the intended users for whom the match procedure returns true.

Our security model consists of a Challenger $\mathcal{C}$, a Simulator $\mathcal{S}$ and an Adversary $\mathcal{A}$. Suppose there exists a polynomial-time adversary $\mathcal{A}$, that can attack our scheme in the Random model with advantage ϵ.

Theorem 5.1. *If the $\mathcal{A}$ can break the proposed modified scheme in the random oracle model, then a simulator can be constructed to play the D - Linear game with a non-negligible advantage.*

Proof. We show that without a correct decryption key $\mathcal{A}$ is not able to compute any function of C' .If the $\mathcal{A}$ is able to succeed in doing so with non-negligible advantage ϵ_1, then we are able to design a simulator $\mathcal{S}$ that can play the D-Linear game with advantage $\frac{\epsilon_1}{2}$. We consider a challenger $\mathcal{C}$, a simulator $\mathcal{S}$ and a polynomial-time adversary $\mathcal{A}$. Suppose that $\mathcal{A}$ is able to distinguish a valid ciphertext from a random element with advantage $\epsilon_{dli}(l)$. We build $\mathcal{S}$ that can play the D-Linear game with advantage $\frac{\epsilon_{dli}(l)}{2}$. In the proof we are using a variant of D-Linear assumption which is equivalent to that defined in section - 2.6.2 and used in [47, 48]. The simulation proceeds as follows.

Let $\mathcal{C}$ set the groups G_1 and G_2 with an efficient bilinear map e and generator g. The $\mathcal{C}$ flips a fair binary coin μ, outside of $\mathcal{S}$'s view. If μ = 0, then $\mathcal{C}$ sets (g,Z_1, Z_2, Z_3, Z_4,Z) = (g, g^{z_1}, g^{z_2}, $g^{z_2 z_4}$, $g^{z_3+z_4}$, $g^{z_1 z_3}$), otherwise it sets g, Z_1, Z_2, Z_3, Z_4, Z) = (g, g^{z_1}, g^{z_2}, $g^{z_2 z_4}$, $g^{z_3+z_4}$, g^{z}) for values z_1, z_2, z_3, z_4 and z chosen randomly from $\mathbb{Z}_p$.

Setup: $\mathcal{S}$ takes the following values: $g_1 = g^{z_1}$, $g_2 = g^{z_2}$, $g_2^{\beta} = g^{\beta' z_1}$, where $\beta = \frac{\beta' z_1}{z_2}$ for some randomly chosen value β' from $\mathbb{Z}_p$. With the selection of random value α from $\mathbb{Z}_p$, g_1^{α} is calculated as $g^{z_1 \alpha}$. $H(x)$ is computed as $g^{z_1(1+H'(x))} = Z_1^{(1+H'(x))}$. H' is defined as a random oracle function which maps any random string from {0,1}* to an element of $\mathbb{Z}_p$. $\mathcal{S}$ announces the public key as $g_1 = Z_1$, $g_2 = g^{z_2}$, $g_1^{\alpha} = Z_1^{\alpha}$, $g_2^{\beta} = Z_1^{\beta'}$.

Phase 1: $\mathcal{A}$ issues a textitkth number of key generation queries to $\mathcal{S}$ for the set L_k of attributes. To generate the response, $\mathcal{S}$ picks a random value $\rho \in \mathbb{Z}_p$ and calculates the key components as.

$$\begin{aligned} D &= (g_2 \prod_{i=1}^{n} H(1\|i\|v_{i,j_i}))^{\alpha} \cdot g_2^{\rho} \\ &= g^{z_2 \alpha} \cdot g^{z_1 \alpha (n + \sum_{i=1}^{n} (H'(1\|i\|v_{i,j_i})))} \\ &= Z_2^{\alpha} \cdot Z_1^{\alpha (n + \sum_{i=1}^{n} (H'(1\|i\|v_{i,j_i})))} \\ \bar{D} &= g_1^{\frac{\rho}{\beta}} = g^{z_1 \frac{\rho z_2}{\beta' z_1}} \\ &= Z_2^{\frac{\rho}{\beta'}} \end{aligned}$$

Phase 2: $\mathcal{A}$ is allowed to run a polynomially bounded number of queries for secret keys, and ciphertext matching phases without violating the restrictions imposed during challenge phase.

Guess: $\mathcal{A}$ submits a guess ν' of ν. If $\nu' = \nu$, then $\mathcal{S}$ outputs $\mu = 1$ to indicate that it was given a valid D-Linear tuple; else, it outputs $\mu = 0$ to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about ν, in turn, $Pr[\nu \neq \nu' | \mu = 0] = \frac{1}{2}$. As $\mathcal{S}$ guesses μ'=0 when $\nu \neq \nu'$, $Pr[\mu = \mu' | \mu = 0] = \frac{1}{2}$. If $\mu = 1$, then $\mathcal{A}$ is able to view the valid encryption components with advantage $\epsilon_{dli}(l)$, a negligible quantity in security parameter in l. Therefore, $Pr[\nu = \nu' | \mu = 1] = \frac{1}{2} + \epsilon_{dli}(l)$. Similarly, $\mathcal{S}$ guesses μ'=1 when $\nu = \nu'$, in turn, $Pr[\mu' = \mu | \mu = 1] = \frac{1}{2} + \epsilon_{dli}(l)$. The overall advantage of the $\mathcal{S}$ in D-Linear game is $\frac{1}{2} \times Pr[\mu = \mu' | \mu = 0] + \frac{1}{2} \times Pr[\mu = \mu' | \mu = 1] - \frac{1}{2} = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times (\frac{1}{2} + \epsilon_{dli}(l)) - \frac{1}{2} = \frac{\epsilon_{dli}(l)}{2}$.

Therefore, if $\mathcal{A}$ has a non-negligible advantage $\epsilon_{dli}(l)$ in the above game then we can build a simulator (S) which can break the D-Linear problem with non-negligible quantity $\frac{\epsilon_{dli}(l)}{2}$, which is an intractable problem. Hence, proved. □

Theorem 5.2. *The proposed modified scheme provides receiver anonymity in IND-CP game, if the Discrete Logarithm(DL) assumption holds with a negligible advantage ϵ_{dl} and if the $H(.)$ is a collision resistant hash function.*

Proof. We assume that $\mathcal{A}$ has obtained the hash outputs of every attribute values in the system.

Setup: $\mathcal{C}$ computes and announces the public keys: g_1, g_2, g_1^{α}, and g_2^{β}.

Phase 1: $\mathcal{A}$ issues k^{th} number of attribute sets L_k of attribute values to get the secret keys for those attribute value sets.

Challenge: $\mathcal{A}$ submits two access policies T_0^* and T_1^* for which he wishes to be challenged upon, with the condition that for any set of attributes L_k submitted by $\mathcal{A}$ in Phase 1, $F(L_k,T_0^*) = F(L_k,T_1^*)$. That is, $\mathcal{A}$ is allowed to issue a valid secret key whose attributes can satisfy both the challenge access policy (T_0^* and T_1^*) or none of them. To make the differentiation between two policies let $\mathcal{A}$ has chosen the attribute λ $(1 \leq \lambda \leq n)$. In T_0^* and T_1^* for the attribute λ, $T_{0,\lambda}^* \neq T_{1,\lambda}^*$. There is at least one value $v_{\lambda,r}$ from the value set of attribute λ, such that $v_{\lambda,r} \notin T_{0,\lambda}^*$ and $v_{\lambda,r} \in T_{1,\lambda}^*$. Here, $1 \leq r \leq m_{\lambda}$. For rest of the attributes we assume that $T_{0,i}^*$= $T_{1,i}^*$, where $1 \leq i \leq n$ and $i \neq \lambda$.

Now $\mathcal{C}$ flips a random coin ν and submits the CT_ν

- If $\mathcal{A}$ has not retrieved a key which can satisfy either of the challenge access structure, then the game is as described in theorem 5.1.
- In the challenge ciphertext, the components which makes a differentiation between access policies T_0^* and T_1^* is $\tilde{C}_{\lambda,r}$.
    - If $\nu = 0$ then $\tilde{C}_{\lambda,r}$ is a random value. This is valid because $v_{\lambda,r}$ is not in $T_{0,\lambda}$ as per the definition.
    - If $\nu = 1$ then $\tilde{C}_{\lambda,r}$ is set as $g_2^{t_i} H(1\|\lambda\|v_{\lambda,r})^{st+t'}$.

$\mathcal{A}$ is given ciphertext $\langle\ \check{C}, \hat{C}, \bar{C}, C', \{\{\ \tilde{C}_{i,j}\ \}_{1\leq j\leq m_i}\}_{1\leq i\leq n}\ \rangle$.

Phase 2: $\mathcal{A}$ is allowed to run a polynomially bounded number of queries for secret keys and challenge access structure with the condition that he can only obtain the secret keys which can satisfy either both the challenge access structures or none of them.

Guess:$\mathcal{A}$ submits a guess ν' of ν. If $\nu' = \nu$, then $\mathcal{A}$ wins the game. To win the game $\mathcal{A}$ needs to discover whether the value of $\tilde{C}_{\lambda,r}$ is a correct ciphertext component or a random element. We will show that the advantage of $\mathcal{A}$ in making this decision is negligible.

To win the game, $\mathcal{A}$ tries to find the value of st and t' from $\hat{C} = g_1^{st}$ (or from $\bar{C} = g_2^{st\beta}$) and $\check{C} = g_1^{t'\alpha}$. ($g_1$,$g_2^{\beta}$ and g_1^{α} are issued as a part of public keys.) The advantage of $\mathcal{A}$ in retrieving the values of st and t' is equivalent to the advantage of breaking the Discrete Logarithm assumption, which is an intractable problem. In an alternative way to win the game, $\mathcal{A}$ tries to find a pair of values from the valueset of an attribute whose hash values of H function collide. $\mathcal{A}$ does so before committing the access policies T_0^* and T_1^*. Suppose that $\mathcal{A}$ has found such a pair of ($v_{\lambda,\eta}$,$v_{\lambda,r}$), that is, $H(1\|\lambda\|v_{\lambda,\eta}) = H(1\|\lambda\|v_{\lambda,r})$ then he can include the value of $v_{\lambda,\eta}$ for the attribute λ in both the challenge access policies T_0^* and T_1^*, but the value of $v_{\lambda,r}$ is included in only one of the access policies, say in T_1^*. Now at the time of making guess $\mathcal{A}$ compares the value of $\tilde{C}_{\lambda,\eta}$ with $\tilde{C}_{\lambda,r}$. If both are same then $\mathcal{A}$ gives the answer as $\nu' = 1$; else, $\nu' = 0$. The probability of winning the game is equivalent to the probability of finding two different values of one attribute which have the same hash values. Let there are at maximum m values for an attribute. Then the probability that any two values will have the same hash values is $O(\frac{m^2}{p})$. Considering the polynomial space m for an attribute value set and sufficient large size of p, the advantage of $\mathcal{A}$ is negligible. Therefore, the total advantage of $\mathcal{A}$ in this game is $\epsilon \leq \epsilon_{dl} + O(\frac{m^2}{p})$, which is negligible. □

5.1.4.3 Performance Analysis of The Proposed Scheme

The proposed scheme provides a matching operation construct for improving the performance of any existing AABE scheme which supports AND gate on multi-valued attributes. Previously this approach was suggested by Zhang *et al.* in [49]. As shown in chapter 5, the ciphertext components in Zhang *et al.*'s scheme which are used in matching phase are suffering from the security flaw. Therefore, we have provided the modified construction to solve the same purpose. The proposed modified scheme facilitates a receiver to find out whether he is the intended recipient or not with just n multiplication operations and three bilinear pairing operations. Here n denotes the number of attributes in the system. We have implemented the proposed scheme on a linux system with intel core-i3 processor running at 2.30 GHz and 3 GB RAM. Pairings are constructed on the curve $y^2 = x^3 + x$ over the field F_q for some prime q = 3 mod 4. The order p of the groups G_1 and G_2 is a prime number 160 bits, while the length of q is 512 bits. The resultant time required in matching phase operation is around 0.04 to 0.08 seconds with respect to total number of attributes values ranging from 10 to 100 .

We also provide the comparison of our proposed scheme parameters with that of Zhang *et al.*'s scheme. While comparing the proposed scheme with Zhang *et al.*'s scheme, we have found following results with respect to matching phase operation.

As from the table 5.1 we can see that there is significant decrease in the storage space for secret key and computation for matching operation on user side.

Parameters (Used for Match Operation)	Zhang *et al.*'s Scheme [49]	Proposed scheme
Number of User key components	n+2	2
Number of Cipher components	$m_i \cdot n + 3$	$m_i \cdot n + 3$
Number of Bilinear Mapping Operations	3	3
Number of Multiplication Operations	2·n	n

Table 5.1: Parameter Comparison between Matching operation of Zhang's Scheme and proposed scheme. Here n denotes the attribute categories in the system and m_i denotes the number of attribute values in i^{th} category ($1 \leq i \leq$ n).

5.1.5 Discussion

The proposed construction for Match operation can be merged with any existing AABE scheme to improve its decryption performance with the help of Match operation. However, instead of performing Match operation with cost-effective computation and then performing the costly decryption operation, an AABE scheme should be designed with cost-effective decryption operation. With this primary objective we have designed our next scheme which provides cost-effective decryption and message authenticity properties together.

5.2 The Proposed Privacy Preserving Signcryption Scheme

We present an anonymous signcryption scheme that provides sender privacy and receiver anonymity. The scheme supports "AND gate on multivalued attributes" access structure and identity-based signature as the building blocks. The identity itself contains the attributes of the sender. In the scheme, if a receiver is not able to decrypt the message because of insufficient access privileges, then he will not be able to determine who is the sender. When an authorized recipient decrypts a ciphertext, then he will be able to learn and verify who is the sender. Therefore, we have used the term sender privacy, which denotes that only authorized recipients can learn the sender identity. Receiver anonymity refers that even when an authorized user unsigncrypts a ciphertext, he can not learn the access policy and can not discover who else are the recipient of the same ciphertext.

5.2.1 Design Goals

The design goals for the proposed scheme are as follows.

Functional Goals

- The signcryption should be performed on data owner side effectively.
- The unsigncryption cost of the scheme on user side should be minimal.

Security Goals

- A user can not generate a valid signcryption of a message with a false identity.
- Only a receiver whose secret key satisfies the access policy of the ciphertext can unsigncrypt the ciphertext and learn the sender identity.
- After the successful decryption of a ciphertext the receiver will be able to verify the signature.
- An authorized receiver of a ciphertext cannot gain the details about the ciphertext policy more than the information that his attributes satisfy the access policy.

5.2.2 ID generation from Attributes

In the proposed scheme, each user is issued a signature key from a secondary identity. The secondary identity is generated from the attribute values possessed by the user as well as a unique identity of that user. As we discuss in the access policy structure, there are n attributes in the system. Let each attribute i has a valueset of size m_i and the person has a unique identity $\hat{id}$, then the ID from user is generated as follows:

$$ID = \{0,1\}^{\sum\limits_{i=1}^{n} m_i} || \hat{id}$$

The binary string of size $\sum_{i=1}^{n} m_i$ represents the attribute values possessed by the user. To represent the value of attribute i, m_i bits are used. Let the user possess j^{th} value of an attribute i, then from m_i bits used for attribute i, the j^{th} bit is set to 1 and rest of the m_i - 1 bits are set to 0. This is done for every m_i bits where $1 \leq i \leq n$.

Example: Let there are 3 attributes in a healthcare organization which has branches spread across 3 cities in the country : Role, Department, City. Each attribute has multi-value set. The valueset for Roll is of size 3 Clinical Staff, Administrative Staff, Patient; the Department has valueset of size 4 Accident, Emergency, Cardiology and Surgery; and the City has valueset of size 3 Baltimore, New Jersey, Houston. Now let David has registered himself as a patient in cardiology department in New Jersey then his attribute will be represented in form of string as 001010010. This string will be combined with identity string related to David (such as his name, birth date, his registration no. or any such info).

The ID generated in this way can help the receiver to identify the person along with the attribute values possessed by the sender.

5.2.3 Scheme Definition

Definition 17. *The scheme is defined with a 4-tuple (Setup, KeyGen, Signcrypt, Unsigncrypt) as follows.*

Setup(1^l)*: The AC inputs a security parameter l and outputs the master secret Key MSK and public parameters PK.*

KeyGen(MSK, ID, L)*: It is a randomized algorithm run by AC, that takes as input the master secret Key MSK along with user's unique identity ID and a set of attributes L of the user. It outputs two secret keys SK_{si} and SK_d. SK_{si} is generated from ID and used for signcryption operation of data. SK_d comprises all attributes in L and used for unsigncryption operation of the received ciphertext.*

Signcrypt(*PK, M, T, ID,* SK_{si})*: Signcrypt is a randomized algorithm run by the sender. It takes as input the system's public parameters PK, the message M to be encrypted, the access structure T, the sender's identity ID and the sender's secret key for signature* SK_s*. The output is a signed ciphertext CT.*

Unsigncrypt(*PK,CT,*SK_d)*: Unsigncrypt is a deterministic algorithm, where the user first performs the decryption operation on the ciphertext CT using his secret key* SK_d *and then performs verification procedure to check authenticity of message and sender identity.*

5.2.4 Security Model

For the proposed scheme we define the following goals of an adversary.

- The adversary can learn the plaintext data or sender identity from the ciphertext without having a valid secret key for unsigncryption of ciphertext.
- The adversary can retrieve the information about underlying access policy.
- The adversary can generate a valid signcrypted message with a spoof identity.

The security of the scheme has been proven secure in the model of "Indistinguishability in ciphertext policy and adaptively chosen ciphertext attack (IND-CP-CCA2)" and "Existential unforgeable in adaptive-predicate chosen plaintext attack (AP-EUF-CPA)", as defined in chapter 2. The formal definition of both the security models are given below.

AP-EUF-CPA Model:

Let the Γ denote the cryptographic scheme with the tuples $\langle$Setup, KeyGen, Signcrypt and Unsigncrypt$\rangle$.

```
$AP - EUF - CPA_{\Gamma}^{\mathcal{A}}(l)$

$(PK, MSK) \leftarrow_{\$} Setup(1^{l})$
$(m^{*}, ID^{*}, T^{*}, c^{*}) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{KeyGen+SGn+USc}}(1^{l}, PK, L, ID, T, M, C)$
If($ID^{*}$ = $ID$) for any query to $\mathcal{O}_{KeyGen}$ then return $\perp$
If($m^{*}$=$m$ AND $ID^{*}$=$ID$ AND $T^{*}$=$T$)
        for any query to $\mathcal{O}_{SGn}$ then return $\perp$
$(c_{b}) \leftarrow_{\$} \mathcal{O}_{SGn}(MPK, m^{*}, ID^{*}, T^{*})$
return $c^{*} = c$
```

Definition 18. *A signcryption scheme is AP-EUF-CPA secure, if the advantage of adversary $\mathcal{A}$ as defined below is negligible.*

$$\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\mathrm{ap-euf-cpa}} l = \tfrac{1}{2} - \Pr\left[1 \leftarrow AP - EUF - CPA_{\Gamma}^{\mathcal{A}}(l)\right]$$

IND-CP-CCA2 Model:

Let the Γ denote the cryptographic scheme with the tuples ⟨Setup, KeyGen, Signcrypt and Unsigncrypt⟩. In IND-CP-CCA2 model, the $\mathcal{A}$ is given access to the Oracle for KeyGen, Signcrypt and Unsigncrypt. It can retrieve a number of secret keys, chosen signcrypted messages and unsigncrypted texts from the chosen ciphertexts. In the challenge phase the $\mathcal{A}$ issues two pairs of message, ID and access policy as (m_0, ID_0, T_0) and (m_1, ID_1, T_1) where $|m_0| = |m_1|$ and $|ID_0| = |ID_1|$. A bit b is selected in random and accordingly the signcryption of m_b with respect to access policy T_b and sender identity ID_b is computed and given to $\mathcal{A}$. Once again $\mathcal{A}$ is given access to the KeyGen oracle and Unsigncrypt oracle. The restriction imposed on the $\mathcal{A}$ is that, he can retrieve SK from KeyGen Oracle which can satisfy either both the challenge access structure T_0^{*} and T_1^{*} or none of them. If $\mathcal{A}$ has retrieved a secret key SK which can satisfy both the challenge access structure then $m_0 = m_1$ and $ID_0 = ID_1$. To the unsigncrypt oracle, the adversary can submit any ciphertext other then the challenge ciphertext. At last the $\mathcal{A}$ issues a bit-value b'. The $\mathcal{A}$ wins the game if $b = b'$.

```
$IND - CP - CCA2_{\Gamma}^{\mathcal{A}}(l)$

$(PK, MSK) \leftarrow_{\$} Setup(1^l)$
$(m_0, ID_0, T_0^*)(m_1, ID_1, T_1^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{KeyGen+SGn+USc}}(1^l, PK, L, ID, M, C)$
If (F($L$,$T_0^*$) $\neq$ F($L$,$T_1^*$)) then return $\perp$
If (F($L$,$T_0^*$) = F($L$,$T_1^*$) = 1) then $m_0 = m_1$
$(c_b) \leftarrow_{\$} \mathcal{O}_{SGn}(MPK, m_b, T_b^*)$
$b' \leftarrow_{\$}$ $\mathcal{A}^{\mathcal{O}_{KeyGen+SGn+USc}}(1^l, PK, L, ID, M, C, c_b)$

    If (F($L$,$T_0^*$) $\neq$ F($L$,$T_1^*$)) then return $\perp$
    If ((F($L$,$T_0^*$) = F($L$,$T_1^*$) = 1) AND (($m_0 \neq m_1$) OR)($ID_0 \neq ID_1$))
            then return $\perp$
    If (C = $c_b$) then return $\perp$**return** $b'$
**return** $b' = b$
```

Definition 19. *A signcryption scheme is secure in IND-CP-CCA2 model, if the advantage of adversary $\mathcal{A}$ as defined below is negligible.*

$$\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\text{ind-cp-cca2}} l = \tfrac{1}{2} - \Pr\left[1 \leftarrow IND - CP - CCA2_{\Gamma}^{\mathcal{A}}(l)\right]$$

As per the definition of IND-CP-CCA2 model, the construction of scheme doesn't allow an adversary without a valid decryption key to learn anything from the ciphertext. The model also defines that with a valid decryption key, the adversary can unsigncrypt the ciphertext, but can not learn the access policy. This requirement states that an authenticate user after decrypting the ciphertext can learn that his attribute values are included in the access policy, but he can not identify which other attribute values are included in the access policy.

5.2.5 Detailed Construction

Setup(1^l→(MSK,PK): The AC selects two groups G_1 and G_2 of prime order p whose bit-length is l and a bilinear mapping function $e : G_1 \times G_1 \rightarrow G_2$. The AC chooses two random generators g_1 and g_2 from group G_1 and a secure hash function H_0:$\{0,1\}^* \rightarrow G_1$. The master secret key MSK is chosen as $\langle y, \vartheta \in_R \mathbb{Z}_p \rangle$. The corresponding set of public parameters also known as public key $PK = \langle g_1, g_2, g_1^y, g_2^{\vartheta} \rangle$

is published.

KeyGen(MSK, ID, L)$\rightarrow$(SK_{si},SK_d): Every user in the system gets two secret keys - SK_{si} for signcryption and SK_d for unsigncryption.

- The SK_{si} is computed as $\langle\ S = H_0(0\|ID)^y,\ \bar{S} = H_0(0\|ID)^{\frac{1}{y}}\ \rangle$.
- Let L=[L_1, L_2,$\cdots$, L_n] = [v_{1,j_1}, v_{2,j_2}, $\cdots$,v_{n,j_n}] be the attribute list for the user who requires a secret key. The AC picks a random value ρ and generates a user's secret key SK_d as follows.
  $D = (g_2 \prod_{i=1}^{n} H_0(1\|i\|v_{i,j_i}))^y \cdot g_2^{\rho}$, where L_i=v_{i,j_i}.
  $\bar{D} = g_1^{\frac{\rho}{\vartheta}}$.
  The user gets the unsigncryption key SK_d as $\langle D, \bar{D}\rangle$.

The user also calculates $\prod_{i=1}^{n} H_0(1\|i\|v_{i,j_i})$, which will be used in the unsigncryption algorithm.

Signcrypt(PK, M, T, ID, SK_{si})$\rightarrow$(CT): To signcrypt a message M, the sender selects five random numbers r, s, t, s' and t' from $\mathbb{Z}_p$. Now, the sender makes the n portions of r as r_i for $1 \leq i \leq n$ such that $\sum_{i=1}^{n} r_i$ =r and the n portions of t as t_i and s' as s'_i such that $\sum_{i=1}^{n} t_i$ =t and $\sum_{i=1}^{n} s'_i = s'$. Then, the sender computes two cipher components as

$$C_{s1} = e(g_1^y, g_2)^{(s-1)t} = e(g_1, g_2)^{y(s-1)t}$$
$$C_{s2} = g_1^{ys'}$$

The message M is wrapped with the sender identity ID and the two cipher components. The wrapped message is computed as $M' = M\|ID\|C_{s1}\|C_{s2}$. Now, the remaining cipher components are computed as follows:

$$\begin{aligned}
\check{C} &= M' e(g_1, g_2)^{y(s-1)t} e(g_1^y, H_0(0\|ID) \cdot H_0(M)^{s'}) \\
C_{sign} &= H_0(0\|ID)^{\frac{y+r}{st}} \\
\hat{C} &= g_1^{st} \\
\bar{C} &= g_2^{st\vartheta} \\
C' &= g_1^{t'y}
\end{aligned}$$

For all attribute values from each set T_i, the cipher components $\{\tilde{C}_{i,j}\}_{1\leq j\leq m_i}$ are generated as follows.

- If $v_{i,j} \in T_i$ then

$$\tilde{C}_{i,j} = H_0(M)^{-s'_i} g_2^{t_i} H_0(1\|i\|v_{i,j})^{st+t'} H_0(0\|ID)^{\frac{r_i}{y}}$$

- If $v_{i,j} \notin T_i$ then $\tilde{C}_{i,j}$ is a random value.

The final ciphertext is $CT = \langle\ \check{C}, C_{sign}, \hat{C}, \bar{C}, C', \{\{\ \tilde{C}_{i,j}\ \}_{1\leq j\leq m_i}\}_{1\leq i\leq n}\ \rangle$

Unsigncrypt(PK, CT, SK_d)→(M,ID): The Unsigncrypt algorithm consists of two procedures - *Decryption of ciphertext* and *Verification of sender identity*. The decryption procedure works as follows.

$$\begin{aligned}
C_v \quad &= \frac{e(\hat{C}, D \cdot C_{sign}) e(C', \prod_{i=1}^{n} H_0(1\|i\|v_{i,j}))}{e(\prod_{i=1}^{n} \tilde{C}_{i,j}, g_1^y) e(\bar{C}, \bar{D})} \\
&= e(g_1, g_2)^{y(s-1)t} e(g_1, H_0(0\|ID))^y e(g_1, H_0(M))^{ys'}
\end{aligned}$$

$$\begin{aligned}
\check{C}/C_v \quad &= \frac{M' \cdot e(g_1, g_2)^{y(s-1)t} e(g_1, H_0(0\|ID))^y e(g_1, H_0(M))^{ys'}}{e(g_1, g_2)^{(y(s-1))t} e(g_1, H_0(0\|ID))^y e(g_1, H_0(M))^{ys'}} \\
&= M'
\end{aligned}$$

The receiver extracts the cipher components $C_{s1} = e(g_1, g_2)^{y(s-1)t}$, $C_{s2} = g_1^{ys'}$, and ID from M'. The receiver now starts the verification procedure as follows.

$$\begin{aligned} R_1 \; &= \frac{C_v}{C_{s1}} = \frac{e(g_1, g_2)^{y(s-1)t} \cdot e(g_1, H_0(0\|ID)^y) \cdot e(g_1, H_0(M))^{ys'}}{e(g_1, g_2)^{y(s-1)t}} \\ &= e(g_1, H_0(0\|ID))^y e(g_1, H_0(M))^{ys'} \\ R_2 \; &= e(g_1^y, H_0(0\|ID)) \cdot e(C_{s2}, H_0(M)) \\ &= e(g_1, H_0(0\|ID))^y \cdot e(g_1, H_0(M))^{ys'} \end{aligned}$$

If R_1 and R_2 are equal then the verification succeeds and the sender identity is known to the receiver; else, it returns $\perp$.

5.3 Security Analysis

Our first theorem is to prove our claim that unless a valid decryption key is available, the adversary can not decrypt the ciphertext nor he can learn the access policy or sender information.

Theorem 5.3. *The proposed scheme is IND-CP-CCA2 secure under the D-Linear assumption.*

Proof. We consider a challenger $\mathcal{C}$, a simulator $\mathcal{S}$ and a polynomial-time adversary $\mathcal{A}$. Suppose that $\mathcal{A}$ is able to distinguish a valid ciphertext from a random element with advantage $\epsilon_{dli}(l)$. We build $\mathcal{S}$ that can play the D-Linear game with advantage $\frac{\epsilon_{dli}(l)}{2}$. In the proof we are using a variant of D-Linear assumption which is equivalent to that defined in section 2.6 and used in [47, 48]. The simulation proceeds as follows.

Let $\mathcal{C}$ set the groups G_1 and G_2 with an efficient bilinear map e and generator g. The $\mathcal{C}$ flips a fair binary coin μ, outside of $\mathcal{S}$'s view. If $\mu = 0$, then $\mathcal{C}$ sets (g,Z_1, Z_2, Z_3, Z_4,Z) = (g, g^{z_1}, g^{z_2}, $g^{z_2 z_4}$, $g^{z_3+z_4}$, $g^{z_1 z_3}$), otherwise it sets g, Z_1, Z_2, Z_3, Z_4, Z) = (g, g^{z_1}, g^{z_2}, $g^{z_2 z_4}$, $g^{z_3+z_4}$, g^z) for values z_1, z_2, z_3, z_4 and z chosen randomly from $\mathbb{Z}_p$.

Setup: $\mathcal{S}$ takes the following values: $g_1 = g^{z_1}$, $g_2 = g^{z_2}$, $g_2^{\vartheta} = g^{\vartheta' z_1}$, where $\vartheta = \frac{\vartheta' z_1}{z_2}$ for some randomly chosen value ϑ' from $\mathbb{Z}_p$. With the selection of random value y from $\mathbb{Z}_p$, g_1^y is calculated as $g^{z_1 y}$. $H_0(x)$ is computed as $g^{z_1(1+H_1(x))} = Z_1^{(1+H_1(x))}$. H_1 is defined any random oracle function which provides a random element from $\mathbb{Z}_p$ for any new non-repeated string from {0,1}*. It maintains a list of this input string and output element. If the input is repeated, then it repeats the output from this list. $\mathcal{S}$ announces the public key as $g_1 = Z_1$, $g_2 = g^{z_2}$, $g_1^y = Z_1^y$, $g_2^{\vartheta} = Z_1^{\vartheta'}$.

Phase 1: $\mathcal{A}$ issues a polynomially bounded number of queries to $\mathcal{S}$ and collects the following results in response of his queries.

- Whenever $\mathcal{A}$ makes its k^{th} key generation query for the set L_k of attributes, $\mathcal{S}$ picks a random value $\rho \in \mathbb{Z}_p$ and calculates the key components as.

$$\begin{aligned}
D &= (g_2 \prod_{i=1}^{n} H_0(1\|i\|v_{i,j_i}))^y \cdot g_2^{\rho} \\
&= g^{z_2 y} \cdot g^{z_1 y(n+\sum_{i=1}^{n}(H_1(1\|i\|v_{i,j_i})))} \\
&= Z_2^y \cdot Z_1^{y(n+\sum_{i=1}^{n}(H_1(1\|i\|v_{i,j_i})))} \\
\bar{D} &= g_1^{\frac{\rho}{\vartheta}} = g^{z_1 \frac{\rho z_2}{\vartheta' z_1}} \\
&= Z_2^{\frac{\rho}{\vartheta'}}
\end{aligned}$$

- In the result of query for signcryption key with respect to ID, $\mathcal{S}$ submits $H_0(0\|ID)^y = Z_1^{(1+H_1(0\|ID))y}$ and $H_0(0\|ID)^{1/y} = Z_1^{\frac{(1+H_1(0\|ID))}{y}}$.

- In response to the query for signcryption of messages M as per the access policies T and sender identity ID submitted by $\mathcal{A}$, $\mathcal{S}$ computes the ciphertext with the selection of the random numbers s, t, s' and r from $\mathbb{Z}_p$. The cipher components are generated with the public key parameters set up by $\mathcal{S}$.

- In response to the query for unsigncryption of CT with respect to attribute set L, $\mathcal{S}$ generates the secret keys for attributes included in set L. If the unsigncryption is successful, then $\mathcal{S}$ returns the unsigncrypted message M and sender identity ID. Else, $\mathcal{S}$ returns $\perp$ and aborts.

Challenge: The $\mathcal{A}$ outputs two pairs of message, ID and, Access Policy (M_0, ID_0, T_0^*) and (M_1,ID_1,T_1^*), on which he wishes to be challenged upon with restriction that for any input L submitted in Phase - 1, F(L, T_0^*) = F(L,T_1^*). Also, if for any key generated in Phase 1 on an attribute list L, $F(L,T_0^*) = F(L,T_1^*)$=1, then $M_0 = M_1$ and $ID_0 = ID_1$.

Now, $\mathcal{S}$ flips a random coin ν, and signcrypts M_ν as per sender identity ID_ν and access policy T_ν^*. $\mathcal{S}$ assumes st=z_3, t = z_4, and $t' = \frac{z_3+z_4}{z_1}$. The value of parameter r is assumed to be as $r'z_3$ - y with a random value r' chosen from $\mathbb{Z}_p$ and partitioned in n portions with each portion denoted as $r_i = (r'z_3 - y)/n$. Similarly, a random value s' is picked up from $\mathbb{Z}_p$ and divided in n portions, where each portion is defined as s'_i. For the values which are included in T_ν^*, $\mathcal{C}$ calculates $\tilde{C}_{i,j} = H_0(M_\nu)^{-s'_i} g_2^{t_i} H_0(1\|i\|v_{i,j})^{st+t'} H_0(0\|ID_\nu)^{\frac{r_i}{y}} = g^{-z_1(1+H_1(M_\nu))\frac{s'}{n}} \cdot g^{\frac{z_2 z_4}{n}} \cdot g^{z_1(1+H_1(1\|i\|v_{i,j}))(z_3+\frac{z_3+z_4}{z_1})} \cdot g^{z_1(1+(H_1(0\|ID_\nu)))(\frac{r'z_3-y}{ny})}$. This results in $\tilde{C}_{i,j} = Z_1^{(1+H_1(M_\nu))(\frac{-s'}{n})} \cdot Z_3^{\frac{1}{n}} \cdot Z^{(1+H_1(1\|i\|v_{i,j}))} \cdot Z_4^{(1+H_1(1\|i\|v_{i,j}))} \cdot Z^{\frac{r'(1+H_1(0\|ID_\nu))}{ny}} \cdot Z_1^{\frac{(1+H_1(0\|ID_\nu))}{n}}$. Now, for other attribute values which are not included in T_ν^*, $\tilde{C}_{i,j}$ are random values. The cipher components are computed by $\mathcal{S}$ as $\check{C} = M'_\nu \cdot e(g_1, g_2)^{y(s-1)t} \cdot e(g_1, H_0(0\|ID_\nu)^y) \cdot e(g_1, H_0(M_\nu))^{ys'} = M'_\nu \cdot \frac{e(Z,Z_2)^y}{e(Z_1,Z_3)^y} \cdot e(Z_1, Z_1)^{(1+H_1(0\|ID_\nu))y} \cdot e(Z_1, Z_1)^{(1+H_1(M_\nu))ys'}$, $\hat{C} = g_1^{st} = Z$, $\bar{C} = g_2^{st\vartheta} = Z^{\vartheta'}$, $C' = g_1^{t'y} = g^{(z_3+z_4)y} = Z_4^y$ and $C_{Sign} = H_0(0\|ID_\nu)^{\frac{(y+r)}{st}}$ =$g^{z_1\frac{(1+H_1(ID_\nu))(y+r'z_3-y)}{z_3}} = Z_1^{(1+H_1(ID_\nu))r'}$. Here, $M'_\nu = M_\nu\|C_{s1}\|Z_1^{ys'}\|ID_\nu$ ($C_{s1} = \frac{e(Z,Z_2)^y}{e(Z_1,Z_3)^y}$). The ciphertext is correct if $Z = g^{z_1 z_3}$. Else, it will be a random string. The ciphertext components $\{\{\tilde{C}_{i,j}\}_{\ 1\le j\le m_i}\}_{1\le i\le n}$, $\bar{C}$, $\hat{C}$, C', C_{Sign}, $\check{C}$ are given to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to run a polynomially bounded number of queries for secret keys, signcryption and unsigncryption of the messages without violating the conditions imposed in challenge phase. One more restriction included here is that $\mathcal{A}$ can not query for unsigncryption of CT_b to $\mathcal{S}$

Guess: $\mathcal{A}$ submits a guess ν' of ν. If $\nu' = \nu$, then $\mathcal{S}$ outputs $\mu = 1$ to indicate that it was given a valid D-Linear tuple; else, it outputs $\mu = 0$ to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about ν, in turn, $Pr[\nu \neq \nu' | \mu = 0] = \frac{1}{2}$. As $\mathcal{S}$ guesses μ'=0 when $\nu \neq \nu'$, $Pr[\mu = \mu' | \mu = 0] = \frac{1}{2}$. If μ = 1, then $\mathcal{A}$ is able to view the valid encryption components with advantage $\epsilon_{dli}(l)$, a negligible quantity in security parameter in l. Therefore, $Pr[\nu = \nu' | \mu = 1] = \frac{1}{2}$

+ $\epsilon_{dli}(l)$. Similarly, $\mathcal{S}$ guesses μ'=1 when $\nu = \nu'$, in turn, $Pr[\mu' = \mu|\mu = 1] = \frac{1}{2} + \epsilon_{dli}(l)$. The overall advantage of the $\mathcal{S}$ in D-Linear game is $\frac{1}{2} \times Pr[\mu = \mu'|\mu = 0]$ $+\frac{1}{2} \times Pr[\mu = \mu'|\mu = 1]$ - $\frac{1}{2}$ = $\frac{1}{2} \times \frac{1}{2}$ + $\frac{1}{2}\times(\frac{1}{2} + \epsilon_{dli}(l))$ - $\frac{1}{2}$ = $\frac{\epsilon_{dli}(l)}{2}$.
Therefore, if $\mathcal{A}$ has a non-negligible advantage $\epsilon_{dli}(l)$ in the above game then we can build a simulator (S) which can break the D-Linear problem with non-negligible quantity $\frac{\epsilon_{dli}(l)}{2}$, which is an intractable problem. Hence, proved. □

Our next theorem is to prove that the cipher components provide receiver anonymity. We show that even if the $\mathcal{A}$ gains a valid decryption key, he can decrypt the message and identify the sender, but can not find out the underlying access policy. This proves that a receiver decrypts the message with his secret key, but he is not able to determine which attribute values other than those possessed by the receiver are included in ciphertext access policy.

Theorem 5.4. *The proposed scheme provides receiver anonymity in IND-CP-CCA2 game, if the Discrete Logarithm(DL) assumption holds with a negligible advantage ϵ_{dl} and if the $H_0(.)$ is a collision resistant hash function.*

Proof. We assume that $\mathcal{A}$ has obtained the hash outputs of every attribute values in the system.
Setup: $\mathcal{C}$ computes and announces the public keys: g_1, g_2, g_1^y, and g_2^ϑ.
Phase 1: $\mathcal{A}$ issues a polynomially bounded number of queries to $\mathcal{S}$ and collects the following results in response of his queries.

- $\mathcal{A}$ makes its k^{th} key generation query for the set L_k of attributes

- $\mathcal{A}$ also gets response for the query of issuing signature key related to ID, as $\langle$ $H_0(0\|ID)^y$ and $H_0(0\|ID)^{\frac{1}{y}}$ $\rangle$.

- In response to the query for signcryption of messages M as per the access policies T (where $T \neq T_0^* \neq T_1^*$) and sender identity ID, submitted by $\mathcal{A}$, $\mathcal{C}$ computes the ciphertext with the selection of random numbers s, t and t' from Z_p.

- In response to the query for unsigncryption of CT with respect to attribute set L, $\mathcal{C}$ generates the secret keys for attributes included in set L. If the decryption and verification are finished successfully, then $\mathcal{C}$ returns the unsigncrypted message M and sender id ID. Else, $\mathcal{C}$ returns $\perp$ and aborts.

Challenge: $\mathcal{A}$ submits two message-ID pairs (M_0,ID_0) and (M_1,ID_1). $\mathcal{A}$ also submits two access policies T_0^* and T_1^* for which he wishes to be challenged upon, with the restriction that for any set of attributes L_k submitted during phase-1, $F(L_k,T_0^*) = F(L_k,T_1^*)$. That is, $\mathcal{A}$ is allowed to issue a valid decryption key for the set of attributes which can satisfy either both the challenge ciphertext or none of them. To make the differentiation between two policies let $\mathcal{A}$ has chosen the attribute λ ($1 \leq \lambda \leq n$). In T_0^* and T_1^* for the attribute λ, $T_{0,\lambda}^* \neq T_{1,\lambda}^*$. There is at least one value $v_{\lambda,r}$ from the value set of attribute λ, such that $v_{\lambda,r} \notin T_{0,\lambda}^*$ and $v_{\lambda,r} \in T_{1,\lambda}^*$. Here, $1 \leq r \leq m_\lambda$. For rest of the attributes we assume that $T_{0,i}^* = T_{1,i}^*$, where $1 \leq i \leq n$ and $i \neq \lambda$.
The $\mathcal{C}$ flips a random coin ν and submits the CT_ν for (M_ν,ID^*,T_ν^*)

- If $\mathcal{A}$ has retrieved a key for any queried L=[L_1, L_2, $\cdots$, L_n], such that $F(L,T_0^*) = F(L,T_1^*) = 1$, then $M_0 = M_1$ and $ID_0 = ID_1$.
- In case when $M_0 \neq M_1$, then the game is as described in Theorem 5.3. If $M_0 = M_1$ then only the ciphertext components which makes a differentiation between access policies T_0^* and T_1^* is $\tilde{C}_{\lambda,r}$.
  - If $\nu = 0$ then $\tilde{C}_{\lambda,r}$ is a random value. This is valid because $v_{\lambda,r}$ is not in $T_{0,\lambda}$ as per the definition.
  - If $\nu = 1$ then $\tilde{C}_{\lambda,r}$ is set as $H_0(M_\nu)^{-s_i'} g_2^{t_i} H_0(1\|\lambda\|v_{\lambda,r})^{st+t'}$.

$\mathcal{A}$ is given ciphertext $\langle$ $\check{C}$, C_{sign}, $\hat{C}$, $\bar{C}$, C', $\{\{$ $\tilde{C}_{i,j}$ $\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$ $\rangle$.

Phase 2: $\mathcal{A}$ is allowed to run a polynomially bounded number of queries for secret keys, signcryption and unsigncryption without violating the restrictions defined in challenge phase. Another restriction is that $\mathcal{A}$ can not issue the unsigncrypt queries for CT_b.

Guess:$\mathcal{A}$ submits a guess ν' of ν. If $\nu' = \nu$, then $\mathcal{A}$ wins the game. To win the game $\mathcal{A}$ needs to discover whether the value of $\tilde{C}_{\lambda,r}$ is a correct ciphertext component or a random element. We will show that the advantage of $\mathcal{A}$ in making this decision is negligible.

To win the game, $\mathcal{A}$ tries to find the value of st and t' from $\hat{C} = g_1^{st}$ (or from $\bar{C} = g_2^{st\vartheta}$) and $\check{C} = g_1^{t'y}$. (g_1,g_2^{ϑ} and g_1^{y} are issued as a part of public keys.) The advantage of $\mathcal{A}$ in retrieving the values of st and t' is equivalent to the advantage of breaking the Discrete Logarithm assumption, which is an intractable problem. In an alternative way to win the game, $\mathcal{A}$ tries to find a pair of values from the valueset of an attribute whose hash values of H function collide. $\mathcal{A}$ does so before committing the access policies T_0^* and T_1^*. Suppose that $\mathcal{A}$ has found such a pair of ($v_{\lambda,\eta}$,$v_{\lambda,r}$), that is, $H_0(1\|\lambda\|v_{\lambda,\eta}) = H_0(1\|\lambda\|v_{\lambda,r})$ then he can include the value of $v_{\lambda,\eta}$ for the attribute λ in both the challenge access policies T_0^* and T_1^*, but the value of $v_{\lambda,r}$ is included in only one of the access policies, say in T_1^*. Now at the time of making guess $\mathcal{A}$ compares the value of $\tilde{C}_{\lambda,\eta}$ with $\tilde{C}_{\lambda,r}$. If both are same then $\mathcal{A}$ gives the answer as $\nu' = 1$; else, $\nu' = 0$. The probability of winning the game is equivalent to the probability of finding two different values of one attribute which have the same hash values. Let there are at maximum m values for an attribute. Then the probability that any two values will have the same hash values is $\mathrm{O}(\frac{m^2}{p})$. Considering the polynomial space m for an attribute value set and sufficient large size of p, the advantage of $\mathcal{A}$ is negligible. Therefore, the total advantage of $\mathcal{A}$ in this game is $\epsilon \leq \epsilon_{dl} + \mathrm{O}(\frac{m^2}{p})$, which is negligible. □

Next we prove that our proposed scheme is existentially unforgeable against chosen plaintext attack in adaptive predicate (AP-EUF-CPA) model.

Theorem 5.5. *Let the function H_0:{0,1}$^* \rightarrow G_1$ is a secure Hash function and the order of G_1 is prime p with bitlength l. The adversary $\mathcal{A}$ makes a total κ >0 number of hash queries to H_0, then the advantage of winning the game described in AP-EUF-CPA model is $O(\frac{\kappa^2}{p})$.*

Proof. As per the game rules, the adversary $\mathcal{A}$ does not possess the signature key for ID^*. As a first option to win the game the $\mathcal{A}$ tries to generate valid signature key of ID^*. Because α is private, therefore, the $\mathcal{A}$ needs to compute $H_0(0\|ID^*)^\alpha$ and $H_0(0\|ID^*)^{\frac{1}{\alpha}}$ from the public key (g_1^α) and from the results of signature queries. Because the discrete logarithm problem is intractable, the advantage of $\mathcal{A}$ in computing the signature key himself without knowledge of α is ϵ_{dl}, which is negligible.

As an other option the $\mathcal{A}$ tries to break the collision property of function H_0. The proof will show that the advantage of $\mathcal{A}$ in this option is also negligible. As per the claim the model designed for game is adaptive. Here the $\mathcal{A}$ do not commit the message M^* and ID^* at the start up of game, to produce the forged ciphertext CT^* in the forgery phase. Instead of it, after gathering the query phase results, the $\mathcal{A}$ chooses a pair of (M^*,ID^*) and produces its signed ciphertext CT^* during the forgery phase. Remember that as per the game rules, the $\mathcal{A}$ has not gain the signature key for ID^* during the query phase, neither he has got the signed ciphertext of message M^* with respect to sender identity ID^*.

For simplicity we assume that the ciphertext access policy is same for all ciphertext queries and also it is same for the forged ciphertext CT^*. The $\mathcal{A}$ can gain a number of ciphertext CTs for pairs of (M,ID) during the query phase. These ciphertexts can be divided in three categories.

1. $M \neq M^*$ and $ID = ID^*$
2. $M = M^*$ and $ID \neq ID^*$
3. $M \neq M^*$ and $ID \neq ID^*$

Also the $\mathcal{A}$ gains signature key for a number of IDs where $ID \neq ID^*$. Based on these query results the $\mathcal{A}$ can play the game in two different ways.

Setup: The challenger gives l as the security parameter to run the Setup algorithm as described in the scheme and retrieves the master secret key MSK and public parameters PK. The public parameters PK are sent to the adversary.

Query Phase: The adversary $\mathcal{A}$ will issue following queries :

- $\mathcal{A}$ makes κ number of queries to function H to find a pair of messages :M_1 and M_2 such that $M_1 \neq M_2$ but $H_0(M_1) = H_0(M_2)$.

- $\mathcal{A}$ collects the decryption key components for attribute set L such that $F(L, T^*)$ = 1.

- $\mathcal{A}$ issue a query to generate a ciphertext for a pair (M_1,ID^*,T^*) and get the ciphertext CT_1. The CT_1 is computed with following results.

  - $C_{1,s1} = e(g_1, g_2)^{\alpha(s-1)t}$
  - $C_{1,s2} = g_1^{\alpha s'}$.
  - $\check{C}_1 = M_1' e(g_1, g_2)^{\alpha(s-1)t} e(g_1^{\alpha}, H_0(0\|ID) \cdot H_0(M)^{s'})$ $(M_1' = M_1||C_{1,s1}||C_{1,s2})$
  - $C_{1,sign} = H_0(0\|ID)^{\frac{\alpha+r}{st}}$
  - $\hat{C}_1 = g_1^{st}$
  - $\bar{C}_1 = g_2^{st\beta}$
  - $C_1' = g_1^{t'\alpha}$
  - For attribute values from each set T_i^* the following cipher components are generated.
    - If $v_{i,j} \in T_i^*$ then
      $\tilde{C_{1,i,j}} = H_0(M_1)^{-s_i'} g_2^{t_i} H_0(1\|i\|v_{i,j})^{st+t'} H_0(0\|ID)^{\frac{r_i}{\alpha}}$
    - If $v_{i,j} \notin T_i^*$ then $\tilde{C_{1,i,j}}$ is a random value.

The final ciphertext sent to $\mathcal{A}$ is $CT_1 = \langle\ \check{C}_1, C_{1,sign}, \hat{C}_1, \bar{C}_1, C_1', \{\{\ \tilde{C_{1,i,j}}\ \}_{1\leq j\leq m_i}\}_{1\leq i\leq n}\rangle$

Forgery: As the $\mathcal{A}$ has got the decryption key which can satisfy T^*, he decrypts the ciphertext CT_1 and extracts M_1,$C_{1,s1}$ and $C_{1,s2}$. Then he construct the message M_2' = $M_2||C_{1,s1}||C_{1,s2}$ and compute $\check{C}^* = M_2' \cdot C_{1,s1} \cdot e(g_1^{\alpha}, H_0(0\|ID)) \cdot e(C_{1,s2}, H_0(M_2))$. The ciphertext CT^* is a collection of $\langle$ $\check{C}^*$, C^*_{sign} = $C_{1,sign}$, $\hat{C}^*$ =$\hat{C}_1$, $\bar{C}^*$ = $\bar{C}_1$, $C^{*\prime}$ = C_1', $\{\{$ $\tilde{C^*_{i,j}}$ = $\tilde{C_{1,i,j}}$ $\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}$ $\rangle$ The $\mathcal{A}$ outputs (CT^*, M^* = M_2, T^*, ID^*) where CT^* is a ciphertext generated from M_2,T^*,ID^* and for which neither $\mathcal{A}$ has got the signature key related to $H_0(ID^*)$ nor he has received the ciphertext CT^* in query phase.

The adversary $\mathcal{A}$ wins the game with this approach if he can find a pair of two different messages M_1 and M_2 such that $H_0(M_1) = H_0(M_2)$. The size co-domain of function H_0 is p (because the order of group G_1 is prime number p with bitlength l.), so the probability that the $\mathcal{A}$ will find such a collision and win the game is $O(\frac{\kappa^2}{p})$.

Now in the next version of game the $\mathcal{A}$ tries to find a collision for IDs.

Setup: The challenger gives l as the security parameter to run the Setup algorithm and retrieves the master secret key MSK and public parameters PK. The public parameters PK are sent to the adversary.
Query Phase: The $\mathcal{A}$ will issue following queries:

- $\mathcal{A}$ makes κ number of queries to function H to find a pair of IDs :ID_1 and ID_2 such that $ID_1 \neq ID_2$ but $H_0(0\|ID_1) = H_0(0\|ID_2)$.

- $\mathcal{A}$ collects the signature key components for ID_1.

Forgery: $\mathcal{A}$ will choose $ID^* = ID_2$ for which he has not received the signature key component and produces a fraud signcrypted message for the pair (M^*, ID^* = ID_2). As $H_0(0\|ID_1) = H_0(0\|ID_2)$, the $\mathcal{A}$ can use the signature key component he received for ID_1 in query phase to produce the ciphertext of any message M^* signed with ID_2 during the forgery phase.

As in the previous game the advantage of adversary $\mathcal{A}$ in winning this game will be $O(\frac{\kappa^2}{2^l})$. While considering the $\mathcal{A}$ running in polynomial time and p as a sufficient large number this advantage will be negligible.

Hence, proved. □

5.4 Performance Analysis

To evaluate the performance of PASC as an ABE scheme with receiver anonymity, we have compared the performance of PASC with other AABE schemes [47, 48, 49] on a Linux system with Intel core-i3 processor running at 2.30 GHz and 3GB

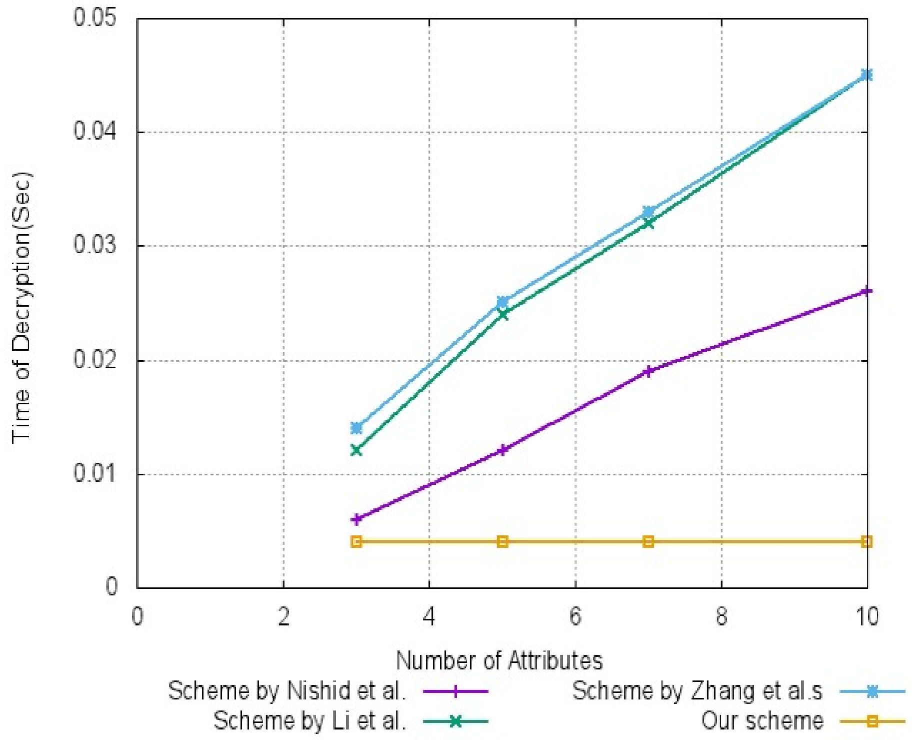


Figure 5.1: Comparison of Unsigncryption operation of proposed scheme with the decryption operation of existing AABE schemes.

Parameters		Nishide *et al.*'s scheme [47]	Li *et al.*'s scheme [48]	Zhang *et al.*'s scheme [49]	Our scheme
Decryption Operation Complexity#	No. of Bilinear Pairing Operation	O(n)	O(n)	O(n)	O(1)
	No. of Multiplication Operation	O(n)	O(n)	O(n)	O(n)

Table 5.2: Comparison of unsigncryption operation cost of our scheme with decryption operation cost of existing AABE schemes.(n = No. of attributes in the system)

RAM. We have used pbc cryptography library to perform bilinear pairing operations. Bilinear pairings are constructed on the curve $y^2 = x^3 + x$ over the field F_q for some prime q=3 mod 4. The order p of the groups G_1 and G_2 is a prime number of size 160 bits, where the length of q is 512 bits. The tables 5.2,5.3 and figure 5.1 shows the comparison of our scheme with the other AABE schemes [47, 48, 49]. In comparison we have not considered the scheme of [50], because it's

n,*m*	CipherText Size (kb)				Encryption Time* (Sec.)			
	[47]	[48]	[49]	Our Scheme	[47]	[48]	[49]	Our scheme
10,10	62	242	62	32	1.63	2.29	1.56	1.07
10,20	123	245	123	62	2.89	5.42	2.78	1.72
15,10	92	244	93	47	2.29	4.18	2.31	1.72
15,20	183	366	184	93	4.2	8.03	4.24	2.72
20,10	122	245	123	62	3.03	5.57	3.06	2.12
20,20	245	489	245	123	5.6	11.06	5.7	3.69

Table 5.3: Comparison of our Scheme with existing AABE schemes for ciphertext size and encryption time. (n = No.of attributes, m = Maximum size of valueset for an attribute, * for our scheme Encryption refers to signcryption)

access structure does not support multiple values of an attribute to be placed in the ciphertext access policy. Also, the construction of scheme in [50] is built upon the composite order bilinear group which provides less efficient pairing operation than that of prime order group. The comparison shows that the performance of PASC is better than the other schemes with similar objective and setup. The decryption operation of proposed scheme has been found cost-effective. We have not considered the parameters for the *Matching phase* of [49] while making the comparison, because the Matching phase of [49] suffers from the security flaws. It is easy to see from table 5.2 and figure 5.1 that the computational cost of the unsigncryption operation in our scheme is much less than the decryption operation of schemes [47, 48, 49]. In fact, our scheme provides constant unsigncryption cost irrespective of the number of attributes in the system, and the decryption operation cost of [47, 48, 49] increases linearly with the number of attributes. This is because the number of bilinear pairing operations are constant and number of multiplication operations increases linearly with the number of attributes. However, the operational cost of multiplication operation is negligible when compared to that of the bilinear pairing operation. There for, the decryption timing of our proposed scheme remains constant. In table 5.2 and 5.3 *n* refers to the number of attributes in the system and *m* denotes the maximum size of valueset for an attribute. While making comparison with existing AABE schemes, encryption time of our scheme refers to Signcryption time and the decryption operation refers to unsigncryption operation. In table 5.3 the comparison of our scheme is made

Schemes Parameters	[36]	[89]	[90]	[91]	[92]	[93]	[94]	PASC
Group Order	Prime	Prime	Prime	Prime	Composite	Prime	Prime	Prime
Access Structure	Threshold	Access Tree	Access Tree	Threshold	LSSS	LSSS	LSSS	AGM
Signature Key Size	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(1)
Decryption Key Size	O(n)	O(n)	O(n)	O(n)		O(n)	O(n)	O(1)
Ciphertext Size	O($N' + n_e$)	O($N' + n_e$)	O($N' + n_e$)	O(N')	O($N' + n_e$)	O($N' + n_e$)	O(N')	O(n)
Nu_{bp}	O(n_e) + O(n_d)	O($N_r + N_s$)	O($N_r + N_s$)	O(n_d)	Max(O(n_e),O(n_d))	O(n_e) + O(n_d)	O(n_d)	O(1)
Sender Attribute Privacy	No	No	No	No	Yes	No	Yes	Yes
Sender Accountability	No	No	No	Yes	No	No	No	Yes
Receiver Anonymity	No	No	No	No	No	No	No	Yes

Table 5.4: Comparison of our scheme PASC with the existing Attribute Based Signcryption schemes. Nu_{bp} denotes the number of bilinear pairing required for unsigncryption operation; n = Number of attributes possessed by user; N = total number of attribute values in the system; N_r = Number of nodes which are traversed by receiver in access tree for constructing the secret; N_s = Number of nodes which are traversed by receiver in access tree to verify the signature; n_e= number of attributes used by the encryptor to signcrypt the message; n_d=number of attributes used by the receiver to unsigncrypt the ciphertext

with [47, 48, 49] for the ciphertext size and encryption time. Both the ciphertext size and encryption time are affected by the number of attributes (n) and size of valueset for each attribute. In the table 5.4, we have compared the features of PASC with existing ABSC schemes. While making the comparison, for simplicity we only include the parameters which are related with either signer's attributes or receiver's attributes. In comparison, we have shown the complexity of bilinear pairing operations in unsigncryption operation of each ABSC schemes, because the unsigncryption operation cost is mainly affected by the number of bilinear pairing operations [104, 105]. Rest of the arithmetic operations such as the multiplication and exponentiation operations have negligible effect on unsigncryption operation timings. The analysis of parameters shown in Table 5.4 proves that the PASC scheme provides better security properties and optimized performance in compare to existing ABSC schemes.

Furthermore, we have compared our scheme with some existing multi-receiver IDSC schemes [101, 102] and shown the results in table 5.5. These two schemes provide the same functionality of single sender and multiple receivers as our scheme does. In the table 5.5 P, Mu, E, A and N_r notations refer to number of pairing Operations, number of multiplication operations, number of exponentiation Operation, number of addition Operations and number of receivers in ID-based signcryption schemes respectively.

As in the previous tables n denotes number of attributes in our scheme which is fixed during the setup algorithm of the scheme .The scheme of Ming *et al* [101] requires 5 pairing operations for unsigncryption of a message. The unsigncryption algorithm of [101] is not affected by number of receivers of a signcrypted text. However, their scheme doesn't provide receiver anonymity. The scheme by Pang *et al.* [102] provides receiver anonymity. However, in their scheme the cost of unsigncryption operation and ciphertext size linearly depends on the number of receivers of the ciphertext. Our scheme provides receiver anonymity. The unsigncryption operation in our scheme linearly depends on the number of attributes in the system. But the number of attributes are fixed during the setup of the system. This yields that once a system is established, the unsigncryption operation of our scheme requires a constant number of operations.

Schemes Parameters	Ming *et al.*'s scheme [101]	Pang*et al.*'s Scheme [102]	PASC
Unsigncryption Cost	$5P$	$3P + (N_r + 1)Mu + (N_r - 1)E + (N_r + 2)A$	$6P + (2n + 6)Mu$
Receiver Anonymity	No	Yes	Yes
Model	Standard Model	Standard Model	Random Oracle Model

Table 5.5: Comparison of our proposed scheme - PASC with the existing multi-recipient ID-based signcryption schemes

5.5 Conclusion

The proposed scheme is the signcryption scheme which has the building blocks of identity based signature and AABE. The identity is derived from user's attributes. The scheme provides digital signature, access control and confidentiality without compromising sender privacy and receiver anonymity. The scheme supports the accountability of unique sender identity. In the scheme, the signature can be verified only if user's attributes can satisfy the ciphertext access policy. Because of the feature of sender identity accountability along with the sender's attributes, the receiver user can learn the unique sender identity. The decryption cost is minimum and constant when compared with existing AABE schemes. The scheme is proven secure in the IND-CP-CCA2 and AP-EUF-CPA model. The implementation results of the scheme shows that the decryption procedure is efficient in comparison to other AABE schemes.

CHAPTER 6

Proxy Reencryption for Anonymous Attribute Based Encryption (PRE-AABE)

A searchable encryption facilitates a user to selectively retrieve the subset of documents, which are accessible to the user and which contains the keyword of user's interest. The user downloads these documents and decrypt them. However, in many practical scenarios, a user makes a search operation to find a document or collection of documents and forward them to other user. This is similar like a person who receives an email and he/she wants to forward that email to any other email-id. From perspective of cloud storage, there can be such scenarios, when user *Alice* wants to forward the encrypted data, which is encrypted with Alice's public key, to another user *Bob*. In a traditional way, *Alice* has to download the encrypted data from cloud storage, decrypt it and then after reencrypting the data with Bob's public key, upload the reencrypted data on cloud storage. This process is time-consuming and less efficient when the user is working with his handheld devices having limited storage and computation power. In such scenarios, proxy reencryption technique becomes beneficial to the user. The concept of proxy reencryption was introduced by Blaze *et al.* in [106]. In a system for proxy reencryption, a user provides his proxy reencryption key to the semi-trusted proxy and wants the proxy to perform the reencryption procedure. The reencryption procedure enables a user to delegate her access rights to any other user. For example when Alice wants to share the encrypted document which is accessible to her with Bob then as per this technique, Alice generates a proxy reencryption key and send it to the CS. The CS is now able to perform the reencryption of cipher document on behalf of Alice without learning the underlying plaintext. The output is a ciphertext, which is accessible to Bob.

The searching operation enables a user to select a subset of documents. The proxy reencryption techniques enables the user to forward these encrypted documents to other user without bearing the burden of downloading all those document and reencrypting them again on user side.

6.1 Background

Ciphertext-Policy Attribute-Based Proxy Reencryption (CP-ABPRE) is an extension of classical Proxy Reencryption techniques. In CP-ABPRE, the computation burden for updating the access policy of a ciphertext is transferred to a semi-trusted proxy such as CS. CP-ABPRE techniques allow a semi trusted proxy to update the access policy and accordingly ciphertext components of a ciphertext without uncovering the plaintext. Using this technique, a user is able to delegate his access privileges for a ciphertext to other. However, further security requirement expected for ABE techniques is to provide receiver anonymity. To apply the receiver anonymity during the proxy reencryption process, it is required to hide the access policy of ciphertext which is given as input to reencryption algorithm and access policy of the target ciphertext which is the output of reencryption algorithm.

Liang *et al.* have provided the Ciphertext Policy Attribute Based Proxy Reencryption (CP-ABPRE) scheme in [37], that enables the proxy to update the access policy associated with a ciphertext. They have proved their scheme selectively secure in standard model. Later in [107], Yu *et al.* have used the concept of proxy reencryption for implementing user revocation. In their scheme, whenever the user is revoked, the new master secret key components are updated. Accordingly the ciphertext components which are stored in cloud storage should also be updated. For this reason, the trusted third party generates the proxy rekeys and send it to the CS. The CS performs the reencryption of ciphertext using these proxy rekeys. In [108], Do *et al.* have addressed the issue of security against collusion of revoked user and proxy. They have constructed proxy reencryption for Key Policy Attribute Based Encryption (KP-ABE).

In [70], Liang *et al.* have proposed a CP-ABPRE scheme which is proved to be chosen ciphertext secure . The scheme proposed in [70] presents a CP-ABPRE scheme which provides both keyword search and proxy reencryption functionality. With keyword based search, the user is able to retrieve a subset of data and with reencryption key the user can instruct the cloud to perform the reencryption of those ciphertexts. In their scheme, the decryption operation requires constant number of bilinear pairing operations.

In [109], the Yang *et al.* have presented a ciphertext policy attribute based proxy reencryption scheme. In their scheme, the access policy is represented in form of Access Tree. Their idea is to provide a conditional reencryption. In their scheme, a ciphertext contains an embedded value ω, and the reencryption key contains the value ω'. The reencryption can only be performed on ciphertext where $\omega = \omega'$. Li *et al.* have provided a computationally efficient CP-ABPRE scheme in [110]. Their scheme is constructed using composite order bilinear groups and has been proved adaptively secure.

All these existing CPABPRE schemes fail to provide receiver anonymity because all of them requires the access policy to be attached with ciphertext in clear form.

In [111], Zhang *et al.* have introduced the concept of anonymous proxy reencryption. In their scheme the proxy reencryption task can be transferred to a proxy without compromising the data security. In their scheme, the access policy is represented in form of "AND gate on multi-valued attributes". The scheme in [111] requires an online proxy server within system premises. When a user wants to perform the reencryption, he generates the rekey and send it to the proxy server. The proxy server downloads the data from CS and then reencrypts it. The proxy server will take the approach of “Match-then-Decrypt", to avoid the reencryption overhead for the ciphertext whose access policies can not be satisfied with user’s attributes. Before reencryption, the proxy server performs the match operation to find out if the access policy of an encrypted ciphertext matches with the user's reencryption key. We have analyzed that their scheme is having some performance bottleneck and security issues as described below.

- The match operation requires only a subset of ciphertext components as per the user's attribute values. Therefore, user has to reveal his attribute information to the proxy server. This contradicts with the concept of receiver anonymity. The receiver anonymity demands that a valid recipient of a ciphertext is also not able to find out who else are the recipient of a ciphertext. While in the suggested approach of Match-then-reencrypt the user who issues the reencryption key, discloses his attribute information to the proxy server.

- The proxy server is on system side. Therefore, to perform the reencryption, the data should be downloaded from cloud storage to the proxy server. This increases the communication overhead.

- The proxy server must be powerful enough as the CS, because it has to satisfy the proxy reencryption queries of all users.

6.2 Proposed scheme

We devise a CP-ABPRE with the following features:

We have devised a proxy re-encryption scheme using ciphertext-policy anonymous attribute-based encryption, termed as PRE-AABE, that enables the cloud to perform the proxy reencryption without learning the plaintext contents or access policy within a ciphertext. Our scheme enables CS to perform proxy reencryption without compromising the receiver anonymity. In the scheme, the CS is able to perform the proxy reencryption of a ciphertext, but can not learn either of the underlined plaintext, access policy within a ciphertext, or user's attribute values hidden inside the proxy reencryption key.

The PRE-AABE scheme facilitates multi-hop reencryption. A Bob who receives a reencrytped ciphertext,can further reencrypt that ciphertext for Charlie.

This is possible in the scheme, because the reencryption algorithm of proposed scheme takes as input a ciphertext which can be an original or a reencrypted ciphertext. We have also added the feature of reencryption control. It allows a data owner or a data user to cease the further reencryption of a ciphertext.

To reduce the burden of high compute-intensive bilinear pairing operations in decryption algorithm, we have added the feature of partial decryption. The user provides his secret key in masked format to CS. The CS will perform the partial decryption of data and the partially decrypted data is returned to the user. Final decryption performed on user side includes a light-weight computation. The PRE-AABE is proven adaptively secure in the random oracle model. We have implemented the scheme to evaluate its performance.

6.2.1 Design Goals

The scheme aims to achieve the following goals.

Functional Goals.

- The scheme enables CS to perform reencryption of encrypted data without learning the receiver′ attributes.
- A receiver obtains the ciphertext from the CS in partially decrypted form and performs final decryption, which is less compute-intensive.

Security Goals

- The ciphertext hides the access policy of the ciphertext.
- The reencryption key as well as the partial decryption key hides the user′s attribute values.
- The CS is not able to learn the underlined plaintext data or the access policy attached with the ciphertext.

6.2.2 System Model

The entities which take part in the PRE-AABE are same as described in the section 2.3. In the proposed scheme the user is able to perform two tasks.

- A data user can issue the reencryption query to the CS, so that CS can update the access policy of the ciphertext and the ciphertext becomes accessible to another user whose attributes can satisfy the updated access policy.

- A data user can send his masked secret key to the CS. The CS can perform the partially decryption on the ciphertext and the partially decrypted ciphertext is returned to the user. The final decryption operation is done on user side.

A Cloud Server (CS) provides storage and computation services to the users of the system. On receiving a reencryption query from user, CS performs the reencryption of the ciphertext. In case of a request for partial decryption, the CS will do it and returns the partially decrypted ciphertext is returned to the user.

6.2.3 Scheme Definition

Definition 20. *Proposed scheme consists of six algorithms Setup, KeyGen, Encryption, ReKeyGen, ReEncrypt and Decryption, defined as follows.*

Setup(1^l) → *(MSK,PK): This algorithm is run by Attribute Center AC. It takes as input parameter a security parameter l and outputs the MSK and PK.*

KeyGen*(MSK,L)*→ *(SK): AC runs this algorithm for each user in the system. It takes as input the MSK along with user's attribute set L and outputs a secret key SK comprising components for all attributes in L.*

Encryption*(M,PK,T)* → *(CT): The Encryption algorithm is used for encrypting user's document M as per the access policy T using the system's public key PK. The algorithm outputs an encrypted document CT which is to be uploaded to the cloud.*

ReKeyGen*(PK, SK, T')*→*(RK): The algorithm takes as input a user's secret key SK, public key PK and an access policy for reencrypted ciphertext T'. The algorithm outputs a reencrypted key RK.*

ReEncrypt*(RK,CT)*→ *(CT'): It is run by the proxy to generate a reencrypted ciphertext CT^*. If CT has not gained any reencryption restriction, then the cloud server is able to perform reencryption, else it aborts.*

Decryption*(CT(or CT^*), SK)*→ *(M): The decryption algorithm is performed in two* *stages. The first stage of decryption which contains all costly bilinear operations is performed on cloud side and the second stage of decryption which is lightweight, is performed on user side.*

6.2.4 Security Model

The goals of adversary $\mathcal{A}$ for proposed schemes are listed below.

- The Adversary can retrieve the information about the plaintext from the ciphertext.
- The Adversary learns the access policy hidden inside the ciphertext access policy.

The proposed scheme is secure in IND-CP-CPA model as described in chapter 2. This model states that without a valid secret key, the $\mathcal{A}$ can not decrypt the ciphertext nor he can generated the rekey for performing reencryption. The formal description of IND-CP-CPA model is given here.

Let the Π denote the ABPRE scheme. In IND-CP-CPA model, the $\mathcal{A}$ is given access to the Oracle for KeyGen. It can retrieve a number of secret keys. In the challenge phase the $\mathcal{A}$ issues two pairs of message and access policy as (m_0, T_0) and (m_1, T_1) where $|m_0| = |m_1|$. A bit b is selected in random and accordingly the encryption of m_b with respect to access policy T_b is computed and given to $\mathcal{A}$. Once again $\mathcal{A}$ is given access to the KeyGen oracle. The restriction imposed on the $\mathcal{A}$ is that, he can retrieve SK from KeyGen Oracle which can satisfy either both the challenge access structure T_0^* and T_1^* or none of them. If $\mathcal{A}$ has retrieved a secret key SK which can satisfy both the challenge access structure then $m_0 = m_1$. At last the $\mathcal{A}$ issues a bit-value b'. The $\mathcal{A}$ wins the game if $b = b'$.

```
$IND - CP - CPA_{\Phi}^{\mathcal{A}}(l)$

$(PK, SK) \leftarrow_{\$} Setup(1^l)$
$(m_0, T_0^*)(m_1, T_1^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{KeyGen}}(1^l, PK, L)$
If (F($L$,$T_0^*$) $\neq$ F($L$,$T_1^*$)) then return $\perp$
If (F($L$,$T_0^*$) = F($L$,$T_1^*$) = 1) then $m_0 = m_1$
$(c_b) \leftarrow_{\$} \mathsf{Enc}(PK, m_b, T_b^*)$
$b' \leftarrow_{\$}$ $\mathcal{A}^{\mathcal{O}_{KeyGen}}(1^l, PK, L, c_b)$

        If (F($L$,$T_0^*$) $\neq$ F($L$,$T_1^*$)) then return $\perp$
        If ((F($L$,$T_0^*$) = F($L$,$T_1^*$) = 1) AND ($m_0 \neq m_1$))
                then return $\perp$
        return $b'$
return $b' = b$
```

Definition 21. *The proposed scheme is secure in IND-CP-CPA model, if the advantage of adversary $\mathcal{A}$ as defined below is negligible.*

$$\mathsf{Adv}_{\Phi,\mathcal{A}}^{\mathrm{ind-cp-cpa}} l = \tfrac{1}{2} - \Pr\left[1 \leftarrow IND - CP - CPA_{\Phi}^{\mathcal{A}}(l)\right]$$

In theorem 6.1, we have proved the security of PRE-AABE under DBDH assumption.

6.2.5 Detailed Construction

The construction of the proposed scheme is explained as follows.

Setup(1^l) $\rightarrow$ (MSK,PK): The AC chooses a security parameter l which determines key length, and performs the following steps to generate system keys and public parameters.

- Choose two multiplicative cyclic groups G_0 and G_1 with a prime order p (Value of l decides the length of p).

  Select g_1, g_2 as two generators of group G_1 and define a bilinear mapping e:$G_0 \times G_0 \rightarrow G_1$.

- Define a secure collision resistant hash function H_1: {0,1}* $\rightarrow \mathbb{Z}_p$.

- choose m+3 random elements $\{\alpha, \beta, \gamma, \{r_{i1}, r_{i2}, \cdots, r_{im_i}\}_{1\leq i\leq n}\}$ from $\mathbb{Z}_p$. These elements serves as the private key MSK of the system.

- publish the public key as $\langle\, g_1, g_2, e(g_1, g_2)^{\alpha}, g_2^{\frac{\alpha}{\beta}}, g_2^{\frac{\gamma}{\beta}}, g_1^{\frac{\gamma}{\alpha}}, \{g_2^{r_{i1}}, g_2^{r_{i2}}, \cdots, g_2^{r_{im_i}}\}_{1\leq i\leq n}$ $\rangle$. $g_2^{\frac{\gamma}{\beta}}, g_1^{\frac{\gamma}{\alpha}}$ are included in the construction only for proxy reencryption purpose.

- An existing CPA(Chosen Plaintext Attack)-secure Symmetric Key Encryption scheme (such as AES) is decided to be used for encryption algorithm. We denote this scheme as SKE in our further discussion.

Key Generation(MSK**,**L**)**$\rightarrow$ **(**SK**):** Each user in the system will get a secret key representing the attributes the user possesses. The AC chooses a random value r and generates the user's keys as follows.

- $D_0 = g_1^{r\beta}$.

- $\{D_{i1}{=}g_1^{(H_1(i\|v_{i,j})+r)\frac{\alpha}{r_{i1}}}, D_{i2}{=}g_1^{(H_1(i\|v_{i,j})^2+r)\frac{\alpha}{r_{i2}}},$ $\cdots, D_{im_i}{=}g_1^{(H_1(i\|v_{i,j})^{m_i}+r)\frac{\alpha}{r_{im_i}}}\}_{1\leq i\leq n}$ $(v_{i,j} \in L)$.

The output of the algorithm is the secret key $SK = \langle D_0, \{\{D_{ij}\}_{1\leq j\leq m_i}\}_{1\leq i\leq n}\rangle$

Encryption(M**,**PK**,**T**)** $\rightarrow$ **(**CT**):** Let $T = \{T_1, T_2, \cdots, T_n\}$ where T_i $\{1 \leq i \leq n\}$ is the set of values for an attribute i, which are permissible for decryption ($T_i \subseteq V_i$). When a sender wants to send a document M in encrypted form to a set of users with specific set of attributes, he generates the ciphertext for the document with the following steps.

- randomly pick a value K from G_2 which serves as an symmetric key to encrypt the document M with SKE scheme.

- compute $C_M \leftarrow SKE(M, K)$

- choose a random secret value s from $\mathbb{Z}_p$.

- randomly pick $s_1, s_2, \cdots, s_{n-1}$ from $\mathbb{Z}_p$ and calculates $s_n = s - \sum_{i=1}^{n-1} s_i$.

- For every attribute field i choose a'_i from $\mathbb{Z}_p$ for $1 \leq i \leq n$ and computes $f(x_i) = a'_i(x_i - H_1(i \| \hat{v}_{i,1}))(x_i - H_1(i \| \hat{v}_{i,2})) \cdots\ (x_i - H_1(i \| \hat{v}_{i,m_i})) + s_i$, where $\hat{v}_{i,j}$=$v_{i,j}$(j^{th} value of attribute i) if $v_{i,j} \in T_i$; else, it will be a random value. The resultant equation is

$$f(x_i) = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots + a_{im_i}x^{m_i} \tag{6.1}$$

  Summation of all coefficients except a_{i0} from all equations is denoted as $A_w = \sum_{i=1}^{n} (\sum_{j=1}^{m_i} a_{ij})$.

- Compute the encryption of K as
  $C_K = K \cdot e(g_1, g_2)^{\alpha(s - \sum_{i=1}^{n} a_i 0)}$
  $\hat{C} = g_2^{\frac{A_w \alpha}{\beta}}, C' = g_2^{\frac{A_w \gamma}{\beta}}, \{C_{i1} = \mathrm{g}_2^{a_{i1} r_{i1}}, C_{i2} = \mathrm{g}_2^{a_{i2} r_{i2}}, \cdots, C_{im_i} = \mathrm{g}_2^{a_{im_i} r_{im_i}}\}$
  for $1 \leq i \leq n$.

C' is included in the ciphertext only if the data owner wants to allow the reencryption of this ciphertext, else it will not be included. The algorithm returns $CT = \langle\ C_M, C_K, \hat{C}, C'\ \{\{C_{ij}\}\ _{1 \leq j \leq m_i}\}\ _{1 \leq i \leq n}\ \rangle$ as the output.

ReKeyGen(*PK*, *SK*, *T'*)→(*RK*) It is a randomized algorithm and run by a user. The algorithm takes as input the public key *PK*, a secret key *SK* and an access policy for reencryption T'. The output of the algorithm is a rekey *rk*, which is used by the *CS* to perform the reencryption of a ciphertext *CT* as per new access policy T'. The generation of *rk* involves following computation.

- select a random value K' from group G_1.

- Generate all encryption components $CT_{rk} = \langle\ C_{K'}, \hat{C}, C', \{\{C_{ij}\}\ _{1 \leq j \leq m_i}\}\ _{1 \leq i \leq n}\ \rangle$ with respect to access policy T' as shown in Encrypt algorithm.

- $D'_0 = D_0 \cdot g_1^{\frac{H_1(K')\gamma}{\alpha}} = g_1^{r\beta} \cdot g_1^{\frac{H_1(K')\gamma}{\alpha}}$

- $\{D'_{i1} = D_{i1} \cdot g_1^{H_1(K')} = g_1^{(H_1(i \| v_{i,j}) + r)\frac{\alpha}{r_{i1}}} \cdot g_1^{H_1(K')}$
  $D'_{i2} = D_{i2} \cdot g_1^{H_1(K')} = g_1^{(H_1(i \| v_{i,j})^2 + r)\frac{\alpha}{r_{i2}}} \cdot g_1^{H_1(K')}$
  $\cdots\ D'_{im_i} = D_{im_i} \cdot g_1^{H_1(K')} = g_1^{(H_1(i \| v_{i,j})^{m_i} + r)\frac{\alpha}{r_{im_i}}} \cdot g_1^{H_1(K')}\}\ _{1 \leq i \leq n}.$

The output of algorithm is $rk = \langle\, CT_{rk}, D'_0, \{D'_{i1}, D'_{i2}, \cdots, D'_{im_i}\}_{\,1\leq i\leq n}\,\rangle$.

ReEncrypt(RK**,**CT**)**$\rightarrow$ **(**CT'**)** After receiving a reencryption key rk, the CS performs following computation to perform the reencryption of a ciphertext CT_M. It first calculates following values:

- $R_{e1} = \prod_{i=1}^{n}(\prod_{j=1}^{m_i} e(C_{ij}, D'_{ij}))$
- $R_{e2} = e(\hat{C}, D'_0) = e(g_1, g_2)^{A_w \alpha r} e(g_1, g_2)^{H_1(K') A_w \frac{\gamma}{\beta}}$
- $C_1 = \frac{C_K \cdot R_{e2}}{R_{e1}} = Ke(g_1, g_2)^{A_w \frac{\gamma}{\beta} H_1(K') - A'_w H_1(K')}$
- $C'_1 = \frac{C'}{\prod_{i=1}^{n}(\prod_{j=1}^{m_i} C_{ij})}$ (Here C' is taken from CT_M)

The updated ciphertext CT' now includes C_M (from CT_M), C_1, C'_1 (from computational results) and $C_{K'}$, $\hat{C}$, C', $\{\{C_{ij}\}_{\,1\leq j\leq m_i}\}_{\,1\leq i\leq n}$ from rk. The value of C' from rk should be included only if the data user wants to grant the further reencryption of this ciphertext, else it will not be included. The output of the algorithm is a reencrypted ciphertext $CT' = \langle\, C_M, C_1, C'_1, C_{K'}, \hat{C}, C', \{\{C_{ij}\}_{\,1\leq j\leq m_i}\}_{\,1\leq i\leq n}\,\rangle$

Decrypt(CT**(or** CT^***),** SK**)**$\rightarrow$ **(**M**)**: The decryption operation is performed in two phase. In first phase the user sends his masked secret key SK' to the CS and CS will perform the partial decryption of the ciphertext CT. In second phase the user performs the final decryption computation.

- First Phase :

  The user generates SK' with his secret key SK and a random value ψ chosen from $\mathbb{Z}_p$as follows

  - $\hat{D}_0 = D_0^{\psi}$,
  - $\{\hat{D}_{i1} = D_{i1}^{\psi}, \hat{D}_{i2} = D_{i2}^{\psi}, \cdots, \hat{D}_{im_i} = D_{im_i}^{\psi}\}_{\,1\leq i\leq n}$

  The user submits SK' to cloud. Because of the value ψ, SK' can not reveal the user's attribute values. The cloud now performs the following computation to generate the partially decrypted ciphertext $\hat{CT}$

  - $R_{d1} = \prod_{i=1}^{n}(\prod_{j=1}^{m_i} e(C_{ij}, \hat{D}_{ij}))$=$e(g_1, g_2)^{(s - \sum_{i=1}^{n} a_i 0)\alpha\psi} \cdot e(g_1, g_2)^{A_w \alpha r \psi}$
  - $R_{d2} = e(\hat{C}, D_0) = e(g_1, g_2)^{A_w \alpha r \psi}$

- $R_d = \frac{R_{d2}}{R_{d1}}$

The partially decrypted ciphertext $\hat{CT}$is returned to the user. If the ciphertext is not reencrypted, then the CS will send $\hat{CT}$ = $\langle$ C_M, C_K, R_d $\rangle$ to the user. If the ciphertext is reencrypted, then the CS will send $\hat{CT}$ = $\langle$ C_M, $C_{K'}$, C_1, C_1', R_d $\rangle$. C_1 and C_1' are included because of proxy reencryption. With every further reencryption, these two components will be added.

- Second Phase :

  - If the ciphertext is not reencrypted, then user does the following computation to recover the plaintext M.
    - $K = C_K \cdot R_d^{\frac{1}{\psi}}$
    - Decrypt C_M with SKE using K.
  - If the ciphertext is reencrypted, then the user performs following computation.
    - $K' = C_{K'} \cdot R_d^{\frac{1}{\psi}}$
    - K= $C_1 e(g_1^{H_1(K')}, C_1')$
    - Decrypt M with SKE using K

  The second step of this computation will be repeated with as many times a ciphertext is further reencrypted. This shows that with every new level of reencryption, one bilinear pairing operation is added on user-side.

The calculation of R_{e1} and R_{d1} is elaborated below.

$$
\begin{aligned}
R_{e1} &= \prod_{i=1}^{n}\prod_{j=1}^{m_i} e(g_1^{(H_1(i\|v_i)^j+r)\frac{\alpha}{r_{ij}}} \cdot g_1^{H_1(K')}, g_2^{a'_{ij}r_{ij}}) \\
&= (g_1, g_2)^{(s-\sum_{i=1}^{n} a_{i0})\alpha} \cdot e(g_1, g_2)^{A_w \alpha r \psi} \cdot e(g_1, g_2)^{\sum_{i=1}^{n}\sum_{j=1}^{m_i} a_{ij} r_{ij} H_1(K')} \\
&= e(g_1, g_2)^{s-\sum_{i=1}^{n} a_{i0}\alpha} \cdot e(g_1, g_2)^{A_w \alpha r} \cdot e(g_1, g_2)^{\sum_{i=1}^{n}\sum_{j=1}^{m_i} A'_w H_1(K')}
\end{aligned}
$$

$$
\begin{aligned}
R_{d1} &= \prod_{i=1}^{n}\prod_{j=1}^{m_i} e(g_1^{(H_1(i\|v_i)^j+r)\frac{\alpha\psi}{r_{ij}}}, g_2^{a'_{ij}r_{ij}}) \\
&= (g_1, g_2)^{\sum_{i=1}^{n}(s-\sum_{i=1}^{n} a_{i0})\alpha\psi} \cdot e(g_1, g_2)^{A_w\alpha r\psi} \\
&= e(g_1, g_2)^{s-\sum_{i=1}^{n} a_{i0}\alpha\psi} \cdot e(g_1, g_2)^{A_w\alpha r\psi}
\end{aligned}
$$

6.3 Security Analysis

It is required that the encrypted message does not reveal any information about the ciphertext and underlying access policy to the $\mathcal{A}$. The scheme has been proven secure in the *Indistinguishability against ciphertext-policy and chosen plaintext attack (IND-CP-CPA)* model. In our scheme, unless a correct secret key is available, the ciphertext is indistinguishable from any other group element.

Theorem 6.1. *The PRE-AABE is adaptive secure in IND-CP-CPA model under the DBDH assumption.*

Proof. We prove that without a valid secret key, if the $\mathcal{A}$ is able to distinguish between the correct ciphertext and a random group element with non-negligible advantage, then we can build a simulator $\mathcal{S}$ that can break the DBDH problem with non-negligible advantage. The DBDH challenger sets the group G_1 and G_2. Then the challenger flips a binary coin μ outside of $\mathcal{S}$ view. If μ = 0 then the challenger sets *(g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g,g)^{abc})*. Else, the challenger sets *(g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g,g)^z)* for some random value $z \in \mathbb{Z}_p^*$. In the following game $\mathcal{S}$ plays the role of $\mathcal{C}$.

Setup: $\mathcal{S}$ assumes g_2 = B and g_1 = A. The remaining components of the masker key are chosen by $\mathcal{S}$ as in the original scheme. $\mathcal{S}$ calculates the PK with these chosen values and submits it to $\mathcal{A}$. A random oracle $\mathcal{O}_H$: {0,1}* $\rightarrow \mathbb{Z}_p^*$ is defined to simulate the hash function. $\mathcal{O}_H$ maintains a list of (request,response). Let us denote this list as LH. Whenever a query comes to compute $H(S)$ for some string

$S \in \{0,1\}^*$, $\mathcal{O}_H$ first makes a search in LH for any pair (S,h). Here h is a random element chosen from $\mathbb{Z}_p^*$. If any such pair exists in LH, then h is returned as result of $H(S)$, else an element $h \in_R \mathbb{Z}_p^*$ is picked up and send as response of $H(S)$. This newly generated pair ($H(S)$,h) is added in LH.

Phase 1: $\mathcal{A}$ issues adaptively generated queries to following oracles.

1. $\mathcal{O}_{KeyGen}$: $\mathcal{A}$ submits a list of attribute values L to the $\mathcal{C}$. $\mathcal{S}$ performs following computation to derive a secret key SK_L. {D_0 = $\mathrm{g}_1^{r\beta}$ = $A^{r\beta}$,{{D_{ij} = $g_1^{(H_0(i\|v_{i,j})^j+r)\frac{\alpha}{r_j}}$ = $A^{(H_0(i\|v_{i,j})^j+r)\frac{a}{r_j}}$ $\}_{1\leq j\leq m_i}$ $\}_{1\leq i\leq n}\}$. At the end he submits this key SK_L to $\mathcal{A}$.

2. $\mathcal{O}_{RekeyGen}$: $\mathcal{A}$ submits a list of attribute values L and an access policy T to the $\mathcal{C}$. $\mathcal{S}$ first gains a secret key from $\mathcal{O}_{KeyGen}$. Then as in the real scheme he generates the rekey $rk_{L\to T}$ from L and T. The reencryption key is submitted to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two pairs (M_0,T_0) and (M_1,T_1), where M_0 and M_1 are two equal length messages, and for any set of attribute values L submitted by $\mathcal{A}$ in Phase 1, F(L,T_0) = F(L,T_1) = 0. Here in challenge phase we consider that M_0 and M_1 are elements of group G_2 randomly chosen by $\mathcal{A}$. We do so to reduce the step of encrypting the message with a SKE scheme and then encrypting that symmetric key with our proposed construction. We assume that the SKE scheme chosen is a secure scheme and we wants to prove the security of our proposed construction. Consider c as the secret value used for encryption of keyword. The simulator $\mathcal{S}$ flips a coin $b \in \{0,1\}$. With the outputs obtained from oracles $\mathcal{O}_H$ the simulator $\mathcal{S}$ computes the challenge ciphertext. For $1 \leq i \leq n$-1 select a_i, s_i and build the equations for each attribute category as follows.

$$f(x_i) = a_i(x - H_0(i\|\hat{v}_{ii})) \cdots (x - H_0(i\|\hat{v}_{im_i})) + s_i \tag{6.2}$$
$$f(x_i) = a_{i0} + a_{i1}x + a_{i2}x^2 + \cdots a_{im_i}x^{m_i}$$

where in (6.2) $\hat{v_{ij}}$ = v_{ij} if $v_{ij} \in T_b$; else, if $v_{ij} \notin T_b$ then $\hat{v_{ij}}$ is some random value chosen from $\mathbb{Z}_p^*$ for $1 \leq j \leq m_i$. $\mathcal{S}$ computes C_{i1} = $B^{a_{i1}r_1}$ = $g_2^{a_{i1}r_1}$, C_{i2} = $B^{a_{i2}r_2}$ = $g_2^{a_{i2}r_2}$,

$\cdots$, C_{im_i} = $B^{a_{im_i}r_{m_i}}$ = $g_2^{a_{im_i}r_{m_i}}$ for $1 \leq i \leq n$-1. For the n^{th} attribute category choose a random value $a_n \in \mathbb{Z}_p^*$ and compute the following equation

$$\begin{aligned} f(x_i) = &\ a_n(x - H_0(\hat{v}_{ni})) \cdots (x - H_0(\hat{v}_{nm_n})) \\ = &\quad \acute{a}_{n0} + a_{n1}x + a_{n2}x^2 + \cdots a_{nm_n}x^{m_n} \end{aligned}$$

Note that $\hat{v_{nj}}$ = v_{nj} if $v_{nj} \in T_b$; else, $\hat{v_{nj}}$ is some random value chosen from $\mathbb{Z}_p^*$ for $1 \leq j \leq m_n$. Now, $\mathcal{S}$ computes $e(g_1,g_2)^{\acute{a}_{n0}\alpha}$, C_{n1} = $B^{a_{n1}r_1}$ = $g_2^{a_{n1}r_1}$, C_{n2} = $B^{a_{n2}r_2}$ =$g_2^{a_{n2}r_2}$, $\cdots$, C_{nm_n} = $B^{a_{nm_n}r_{m_n}}$ =$g_2^{a_{nm_n}r_{m_n}}$.

Compute $\hat{C}$=$B^{\frac{A_1\gamma}{\beta}}$ = $g_2^{\frac{A_w\gamma}{\beta}}$, C' = $B^{\frac{A_1\alpha}{\beta}}$ = $g_2^{\frac{A_w\alpha}{\beta}}$, where A_w = $\sum_{i=1}^{n} (\sum_{j=1}^{m_i} a_{ij})$. Compute C_{M_b} = $M_b \cdot \frac{Z^\alpha \cdot e(H_1(w_b),\prod_{i=1}^{n}\prod_{j=1}^{m_i} C_{ij})}{e(A,B)^{\sum a_{i0}\alpha}}$.

Now, $\mathcal{S}$ gives ciphertext CT_b = $\langle$ C_{M_b}, $\hat{C}$, C', and {C_{i1}, C_{i2}, $\cdots$, C_{im_i}} for $1 \leq i \leq n$ $\rangle$.

Phase 2: $\mathcal{A}$ repeats the queries for attribute values L, as it did in Phase 1 with the restrictions that for any input L, F(L,T_0) = F(L,T_1) = 0.

Guess: $\mathcal{A}$ outputs a guess b' of b. If b' = b, then $\mathcal{S}$ outputs μ=1 to indicate that it was given a valid DBDH-tuple, else, it outputs μ=0 to indicate that the ciphertext is a random element. Therefore, $\mathcal{A}$ gains no information about b, in turn, $Pr[b \neq b'|\mu = 0]$= $\frac{1}{2}$. As the simulator guesses μ'=0 when $b \neq b'$, $Pr[\mu = \mu'|\mu = 0]$ = $\frac{1}{2}$. If μ = 1, then the $\mathcal{A}$ is able to view a valid encryption of message with advantage $\epsilon_{dbdh}(l)$, a negligible quantity in security parameter l. Therefore, $Pr[b = b'|\mu = 1]$ = $\frac{1}{2}$ + $\epsilon_{dbdh}(l)$. Similarly, the simulator $\mathcal{S}$ guesses μ'=1 when b = b', in turn, $Pr[\mu' = \mu|\mu = 1]$ = $\frac{1}{2}$ + $\epsilon_{dbdh}(l)$. The overall advantage of the simulator in DBDH game is $\frac{1}{2} \times Pr[\mu = \mu'|\mu = 0]$ +$\frac{1}{2} \times Pr[\mu = \mu'|\mu = 1]$ - $\frac{1}{2}$ = $\frac{1}{2} \times \frac{1}{2}$ + $\frac{1}{2} \times$ ($\frac{1}{2}$ + $\epsilon_{dbdh}(l)$) - $\frac{1}{2}$ = $\frac{\epsilon_{dbdh}(l)}{2}$. Therefore, if the $\mathcal{A}$ has advantage $\epsilon_{dbdh}(l)$ in the above game instance, then we can build a simulator (S) which can break the DBDH problem with negligible quantity $\frac{\epsilon_{dbdh}(l)}{2}$. □

Theorem 6.1 proves that the ciphertext of a message does not disclose the message nor the underlying access policies.

6.4 Performance Analysis

For performance evaluation we have implemented the scheme PRE-AABE using Pairing Based Cryptography (PBC) library framework (version 0.5.14) on a Linux machine. Bilinear pairings operations are constructed using the curve $y^2 = x^3 + x$ over the field F_q for prime q=3 mod 4. The order of the groups G_1 and G_2 is a prime of size 160 bits and the length of q is 512 bits. We have evaluated the scheme with varying number of attributes and their various size of valuesets.

The user side operations such as Encrypt, RekeyGen and Final Decryption are tested on a machine with 2.30 GHz Intel-i5 Processor configuration. The Proxy Reencryption and Partial Decryption operations we have designed for CS. Therefore, we have run them on a google cloud computing instance with machine type *n1-standard-1*. The proxy reencryption time taken by cloud is shown in figure 6.1. The total of attribute values is the summation of number of values for each attribute in the system. Unlike the scheme of [111], our scheme involves all ciphertext components in the process of reencryption and there for our proposed scheme is able to achieve the receiver anonymity. Because of this reason, the reencryption cost increases linearly with the total number of attribute values.

The figure 6.2 shows the decrease in decryption computation overhead on user side. All the costly bilinear pairing operations are now done on CS side during partial decryption operation. To show the difference between computation cost of partial decryption and final decryption, we have plotted the graph shown in figure 6.2 using logarithmic scale. As like for the proxy reencryption operation, the time complexity of partial decryption also increases linearly with the total number of attribute values. However, the compelling computation power of CS can bear the computation cost of proxy reencryption and partial decryption operations. Figure 6.2 clearly shows that, the final decryption done on user side is very lightweight. It requires negligible time and small computing power on user side. This could be beneficial when user is accessing the data from his hand-held battery-driven devices.

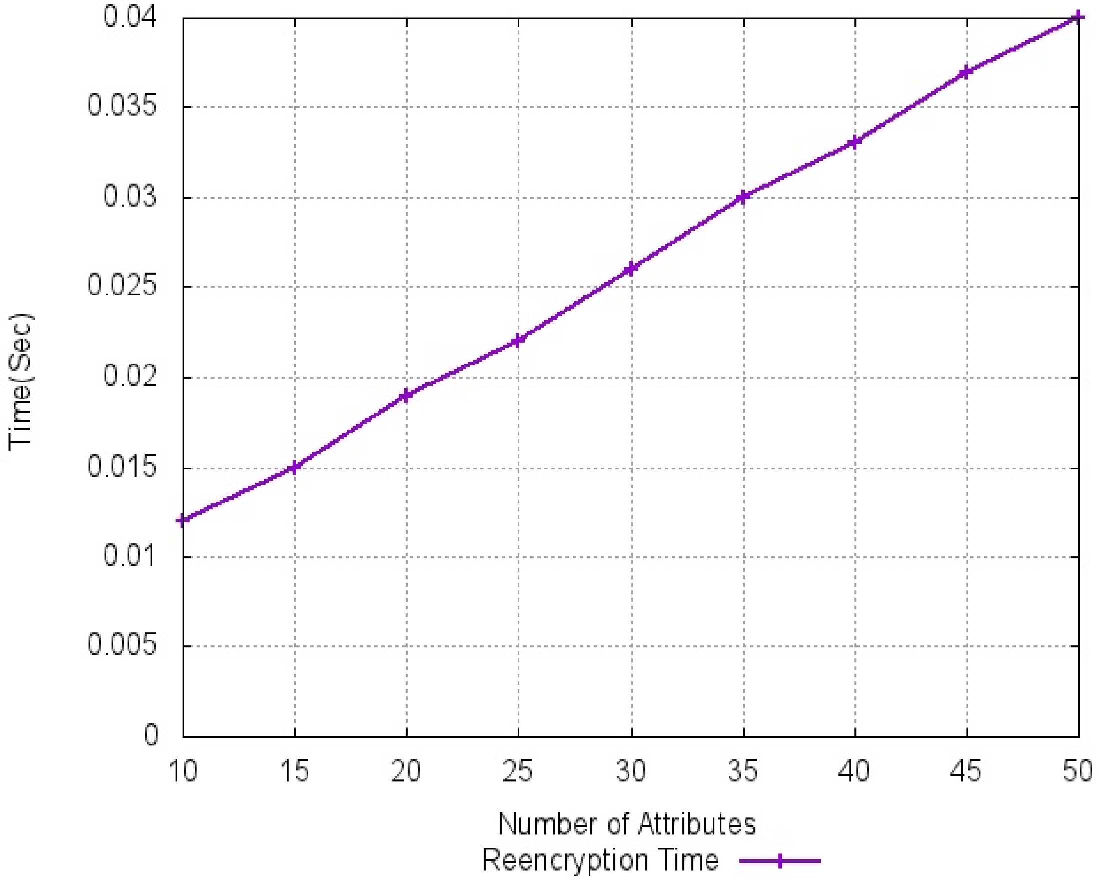


Figure 6.1: Reencryption Time Required by the *CS*

6.5 Conclusion

The proxy reencryption scheme helps in achieving data sharing at reduced computation cost. In case when a user wants to forward an encrypted document stored on public cloud storage to another user, then this technique reduces the computation burden of reencryption on user side. The proposed scheme is an attribute based proxy reencryption scheme, in which a *CS* acts as proxy and after getting a reencryption request from a user, the *CS* can update the access policy of the ciphertext. However, the *CS* is not able to learn the access policy before reencryption or the updated access policy after reencryption. The scheme enables a user to grant the access rights of an encrypted document to another user without compromising the receiver anonymity. The scheme supports multilevel reencryption, where an already shared document can be again reencrypted with

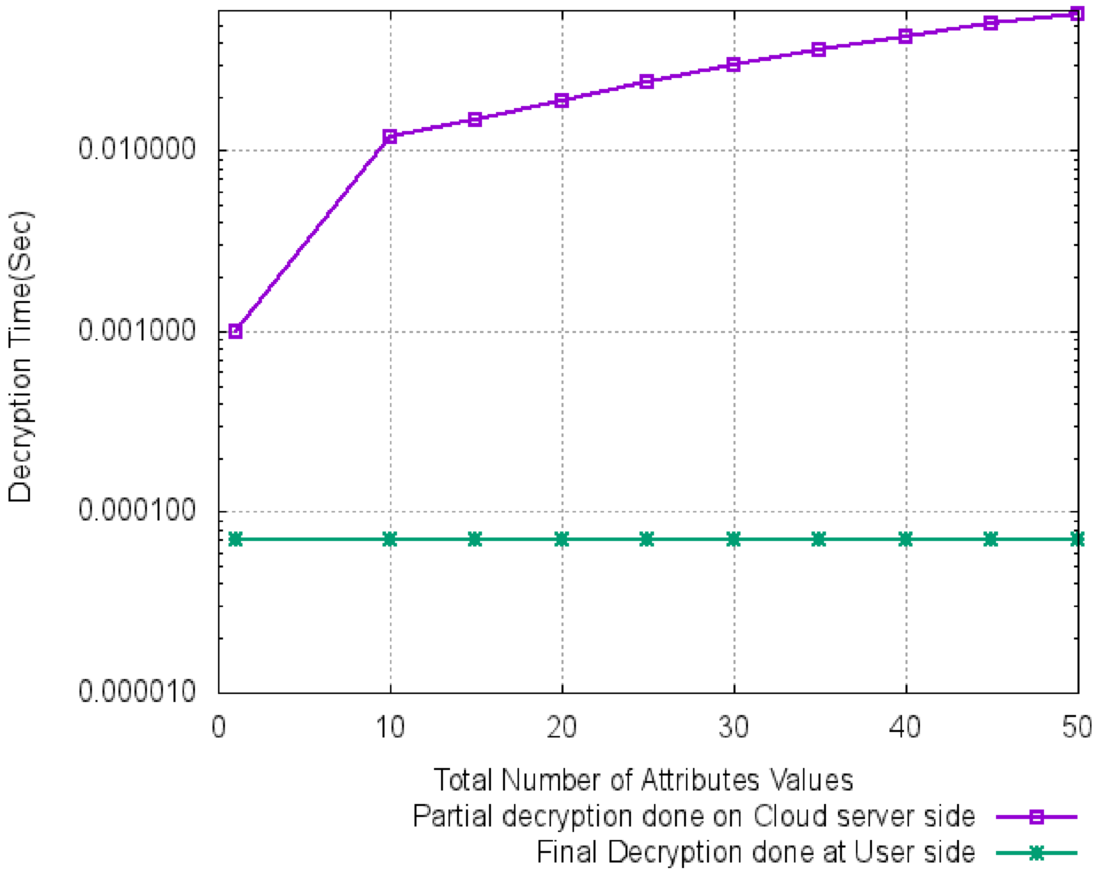


Figure 6.2: Comparison of Partial Decryption Time on *CS* side and Final Decryption Time on User Side

another access policy. In case when a data owner or data user wants to prohibit the further sharing of encrypted data, then proxy reencryption control mechanism is provided in the scheme, The scheme has been proven secure and its feasibility is checked by implementing the scheme. The performance results of the scheme shows that the reencryption time increases with the number of attributes. However, this computation burden is affordable for a *CS*. The operational cost of all user side operations remains constant irrespective of the number of attributes.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This report presents our work on privacy preserving public cloud storage services. Our work covers the cryptographic schemes built upon the attribute based encryption. Providing data confidentiality, access control and receiver anonymity together are primary focus of the proposed work. We have addressed the issue of secure data retrieval from cloud storage without compromising data privacy or receiver anonymity. The basic cryptographic primitive used in our schemes is Attribute based encryption, because it is a public key cryptography method that provides confidentiality along with fine-grained access control policy.

While describing the basics of attribute based encryption, we have presented the existing topologies for defining the access structure. We have chosen the form of AND gate on Multi-valued attributes for presenting the access policy of our proposed schemes. We have presented a detailed study on searchable encryption schemes. We have analyzed the existing schemes which facilitates search operation over encrypted data with fine-grained access control and presented a detailed comparison between them. Our study shown that the receiver anonymity is a salient property required for preserving data security, but only few searchable encryption schemes have addressed this. We have made the security and performance analysis of attribute based searchable encryption schemes that claims to provide the receiver anonymity and found that these schemes suffer from either the security flaws or from performance bottleneck issues.

To make an efficient and secure search operation over attribute based encrypted data with receiver privacy, we have proposed three new schemes. Our first scheme Data Owner based Searchable Encryption (DOSE) provides data owner based search with hidden access policy. A user is able to retrieve the documents from the cloud storage whose access policy is matched with user's attributes and which are uploaded by a specified data owner. The search operation does not reveal the data owner identity nor the user's attributes. The scheme does not support keyword based search, but it enables a cloud server to search the documents with just look-up operations. The absence of mathematical operations from search algorithm provides the fastest search timings. We have provided the security analysis of DOSE.

Our second scheme Receiver Anonymous Searchable Encryption (RASE) supports keyword based searching with hidden access policy. It allows an authorized user to retrieve only a subset of documents pertaining to his chosen keyword and satisfying his access rights from cloud server without revealing his attributes to the server. The scheme is constructed using multi-linear pairing. The RASE scheme uses the access policy in form of AND gate on multi-valued attribute and one value for each attribute is placed in the access policy of an encrypted index. After receiving a search query from user, the cloud server is able to search the data without compromising the data or receiver privacy. The RASE scheme requires the cloud server to perform only few and constant number of mathematical operations to conduct the search operation irrespective to the number of attributes in the system. This property makes the search operation of RASE constant. The security analysis of RASE proves the scheme secure against chosen keyword attack.

Our third scheme having the same objective of searchable encryption with receiver anonymity is Privacy preserving Searchable Encryption - PSE. Like the RASE scheme, it makes possible for an authorized user to retrieve only a subset of documents pertaining to his chosen keyword and satisfying his access rights from cloud server without revealing his attributes to the server. Unlike the RASE scheme, it allows more than one value for an attribute to be included in the access policy. We have made the security analysis of scheme and prove the scheme secure against the chosen keyword attack. The customized system model of PSE makes him secure against the File Injection Attack. The implementation of scheme is tested on google cloud instance. The results obtained from various algorithms are presented in chapter 4.

The searchable encryption facilitates a user to receive only a subset of documents. But for effective utilization of data, it is necessary that the user should be able to decrypt the document with minimal computation power. We have discussed the issue of decryption computation overhead in existing anonymous attribute based encryption schemes, and identified the functional requirement that the decryption computation of an anonymous attribute based encryption scheme should be as minimal as possible. Further we analyzed that providing data authentication is equally important as of preserving data confidentiality. Data authentication is necessary in cloud storage scenario, where multiple data owners upload their data and multiple users access them. Addressing both the performance and security requirements, we have devised an attribute based signcryption scheme that preserves the sender privacy and receiver anonymity. Our scheme Privacy preserving Attribute based Signcryption PASC facilitates signature and encryption operation in a single algorithm. In our scheme, the sender identity can be disclosed and verified only after a successful decryption of the ciphertext. The scheme is found secure in the IND-CP-CCA2 and AP-EUF-CPA model. The unsigncryption operation of PASC has minimum operational cost when compared with existing anonymous attribute based encryption schemes and attribute based signcryption schemes. The cost of unsigncryption operation is constant irrespective to the number of attributes in the system. The scheme also facilitates the sender accountability. Unlike the existing attribute based signcryption schemes, the signature key in PASC consists of user's attributes and a unique identity associated with that user. Therefore, the unsigncryption operation gives the receiver information about the sender's attributes as well as unique identity.

For ease of data sharing a receiver user should be able to share the data stored on a remote side server such as cloud storage with other users with minimal computation and communication overhead. In case when a data owner has shared his encrypted data with a user A and User A wants to forward that data to B, then proxy reencryption has been proved a useful crypto-primitive. For sharing the encrypted data with other user, the user A just has to send a reencryption key to the cloud server which enables the cloud server to recompute the ciphertext components, so that the data can be accessible to the other user B. This reencryption procedure is done by the proxy server, which in our case is the cloud server. It reduces the computation overhead of reencrypting the data on user side. We have analyzed the existing attribute based proxy reencryption schemes and found that the task of performing proxy reencryption with hidden access policy is a challenging job. We have studied one technique which has claimed to provide the proxy reencryption with hidden access policy. But our analysis shows that the scheme suffers from performance bottleneck issues and some security weaknesses. We have devised a novel technique of proxy reencryption that delegates the task of re-encrypting a ciphertext and there by updating the access policy of a ciphertext, without learning the access policy. The proposed scheme on proxy reencryption (PRE-AABE) facilitates proxy reencryption where a user sends the reencryption key generated from his secret key and the new access policy. Using the reencryption key the cloud server can perform the reencryption, but can not learn the access policy before reencryption or after reencryption . The PRE-AABE scheme is also featured with proxy re-encryption control mechanism, which helps a data owner or data user to prevent the further re-encryption of a ciphertext. To minimize the decryption overhead, the cloud server performs partial decryption of ciphertext using the server resources, without learning the receiver's attributes. This provides one more advantage of reducing the decryption cost significantly on user side. The PRE-AABE scheme is proven secure in IND-CP-CPA under BDH assumption. The experimental results show that the CP-ABPRE scheme is efficient and practical.

7.2 Future Work

Based on our research outcomes, the following two research problems can be taken up further in order to make make searchable encryption schemes more practical in real-world applications.

Performance Optimization of Searchable ABE: We have worked on Privacy preserving Searchable Encryption scheme that enables the data owner to place the multiple values for an attribute to be placed in the access policy. The scheme requires a massive number of bilinear pairing operations for conducting search operation. In future, a variant of bilinear pairing operation can be used to reduce the computation overhead.

User Revocation: In our proposed schemes the only way to provide user revocation is to reestablish the system parameters and to reissue the secret keys to each user. This approach is quire costly in an organization, where the users frequently enter and leave the system. Therefore, an efficient way for providing revocation feature can be identified in future work.

Bibliography

[1] Sosinsky, B. *Cloud computing bible*. Vol. 762. John Wiley & Sons, 2010.

[2] *The 17 biggest data breaches of the 21st century*. 2018. URL: https://www.csoonline.com/article/2130877/data-breach/the-biggest-data-breaches-of-the-21st-century.html (visited on 01/26/2018).

[3] Barron, C., Yu, H., and Zhan, J. "Cloud computing security case studies and research". In: *Proceedings of the World Congress on Engineering*. Vol. 2. 1. 2013, pp. 1–5.

[4] Brossard, D. *Coarse-grained vs. fine-grained access control - part I*. 2011. URL: https://www.webfarmr.eu/2011/05/coarse-grained-vs-fine-grained-access-control-part-i/ (visited on 05/28/2011).

[5] Sandhu, R. S. and Samarati, P. "Access control: principle and practice". In: *IEEE communications magazine* 32.9 (1994), pp. 40–48.

[6] Levy, H. M. *Capability-based computer systems*. Digital Press, 2014.

[7] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. "Role-based access control models". In: *Computer* 29.2 (1996), pp. 38–47.

[8] Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., and Samarati, P. "Over-encryption: management of access control evolution on outsourced data". In: *Proceedings of the 33rd international conference on Very large data bases*. VLDB endowment. 2007, pp. 123–134.

[9] Kallahalla, M., Riedel, E., Swaminathan, R., Wang, Q., and Fu, K. "Plutus: Scalable Secure File Sharing on Untrusted Storage." In: *Fast*. Vol. 3. 2003, pp. 29–42.

[10] Goh, E., Shacham, H., Modadugu, N., and Boneh, D. “SiRiUS: Securing Remote Untrusted Storage.” In: *NDSS*. Vol. 3. 2003, pp. 131–145.

[11] Ateniese, G., Fu, K., Green, M., and Hohenberger, S. “Improved proxy re-encryption schemes with applications to secure distributed storage”. In: *ACM Transactions on Information and System Security (TISSEC)* 9.1 (2006), pp. 1–30.

[12] Goyal, V., Pandey, O., Sahai, A., and Waters, B. “Attribute-based encryption for fine-grained access control of encrypted data”. In: *Proceedings of the 13th ACM conference on Computer and communications security*. Acm. 2006, pp. 89–98.

[13] Bethencourt, J., Sahai, A., and Waters, B. “Ciphertext-policy attribute-based encryption”. In: *Security and Privacy, 2007. SP’07. IEEE Symposium on*. IEEE. 2007, pp. 321–334.

[14] Naor, D., Naor, M., and Lotspiech, J. “Revocation and tracing schemes for stateless receivers”. In: *Annual International Cryptology Conference*. Springer. 2001, pp. 41–62.

[15] Boneh, D., Gentry, C., and Waters, B. “Collusion resistant broadcast encryption with short ciphertexts and private keys”. In: *Annual International Cryptology Conference*. Springer. 2005, pp. 258–275.

[16] Fiat, A. and Naor, M. “Broadcast encryption”. In: *Annual International Cryptology Conference*. Springer. 1993, pp. 480–491.

[17] Song, D., Shi, E., Fischer, I., and Shankar, U. “Cloud data protection for the masses”. In: *Computer* 45.1 (2012), pp. 39–45.

[18] Yu, S., Wang, C., Ren, K., and Lou, W. “Achieving secure, scalable, and fine-grained data access control in cloud computing”. In: *Infocom, 2010 proceedings IEEE*. Ieee. 2010, pp. 1–9.

[19] Ye, X. and Khoussainov, B. “Fine-grained access control for cloud computing”. In: *International Journal of Grid and Utility Computing* 4.2-3 (2013), pp. 160–168.

[20] Cheung, L. and Newport, C. “Provably secure ciphertext policy ABE”. In: *Proceedings of the 14th ACM conference on Computer and communications security*. ACM. 2007, pp. 456–465.

[21] Goyal, V., Jain, A., Pandey, O., and Sahai, A. “Bounded ciphertext policy attribute based encryption”. In: *Automata, languages and programming* (2008), pp. 579–591.

[22] Lewko, A. B., Okamoto, T., Sahai, A., Takashima, K., and Waters, B. “Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption.” In: *Eurocrypt*. Vol. 6110. Springer. 2010, pp. 62–91.

[23] Okamoto, T. and Takashima, K. “Fully secure functional encryption with general relations from the decisional linear assumption”. In: *Annual Cryptology Conference*. Springer. 2010, pp. 191–208.

[24] Ostrovsky, R., Sahai, A., and Waters, B. “Attribute-based encryption with non-monotonic access structures”. In: *Proceedings of the 14th ACM conference on Computer and communications security*. ACM. 2007, pp. 195–203.

[25] Waters, B. “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.” In: *Public Key Cryptography*. Vol. 6571. Springer. 2011, pp. 53–70.

[26] Yamada, S., Attrapadung, N., Hanaoka, G., and Kunihiro, N. “Generic constructions for chosen-ciphertext secure attribute based encryption”. In: *International Workshop on Public Key Cryptography*. Springer. 2011, pp. 71–89.

[27] Wang, Y., Wang, J., and Chen, X. “Secure searchable encryption: a survey”. In: *Journal of Communications and Information Networks* 1.4 (2016), pp. 52–65.

[28] Curtmola, R., Garay, J., Kamara, S., and Ostrovsky, R. “Searchable symmetric encryption: improved definitions and efficient constructions”. In: *Journal of Computer Security* 19.5 (2006), pp. 895–934.

[29] Boneh, D., Di Crescenzo, G., Ostrovsky, R., and Persiano, G. “Public key encryption with keyword search”. In: *International conference on the theory and applications of cryptographic techniques*. Springer. 2004, pp. 506–522.

[30] *Search, Store and Share easily*. https://www.4shared.com/.

[31] *MyMedWall - The Missing Link in Healthcare*. https://secure.mymedwall.com/phr/.

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

[33] Wang, C., Li, W., Li, Y., and Xu, X. "A Ciphertext-Policy Attribute-Based Encryption Scheme Supporting Keyword Search Function." In: *CSS*. Springer. 2013, pp. 377–386.

[34] ElGamal, T. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[35] Zheng, Y. "Digital signcryption or how to achieve cost (signature & encryption)« cost (signature)+ cost (encryption)". In: *Advances in Cryptology-Crypto'97* (1997), pp. 165–179.

[36] Gagné, M., Narayan, S., and Safavi-Naini, R. "Threshold Attribute-Based Signcryption." In: *SCN*. Vol. 6280. Springer. 2010, pp. 154–171.

[37] Liang, X., Cao, Z., Lin, H., and Shao, J. "Attribute based proxy re-encryption with delegating capabilities". In: *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*. ACM. 2009, pp. 276–286.

[38] Angelo, D. and Vincenzo, I. "jPBC: Java pairing based cryptography". In: *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*. IEEE, 2011, pp. 850–855. URL: \url{http://gas.dia.unisa.it/projects/jpbc/}.

[39] Zhang, Y., Katz, J., and Papamanthou, C. "All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption." In: *USENIX Security Symposium*. 2016, pp. 707–720.

[40] Chaudhari, P. and Das, M. L. “Privacy Preserving Signcryption Scheme”. In: *International Conference on Distributed Computing and Internet Technology*. Springer. 2017, pp. 196–209.

[41] Chaudhari, P., Das, M. L., and Dasgupta, D. “Privacy-Preserving Proxy Re-encryption with Fine-Grained Access Control”. In: *International Conference on Information Systems Security*. Springer. 2017, pp. 88–103.

[42] Sahai, A. and Waters, B. “Fuzzy identity-based encryption.” In: *Eurocrypt*. Vol. 3494. Springer. 2005, pp. 457–473.

[43] Odlyzko, A. M. “Public key cryptography”. In: *AT&T Technical Journal* 73.5 (1994), pp. 17–23.

[44] Meffert, D. “Bilinear pairings in cryptography”. In: *Master's thesis, Radboud Universiteit Nijmegen* (2009).

[45] Kapadia, A., Tsang, P. P., and Smith, S. W. “Attribute-Based Publishing with Hidden Credentials and Hidden Policies.” In: *NDSS*. Vol. 7. 2007, pp. 179–192.

[46] Yu, S., Ren, K., and Lou, W. “Attribute-based content distribution with hidden policy”. In: *Secure Network Protocols, 2008. NPSec 2008. 4th Workshop on*. IEEE. 2008, pp. 39–44.

[47] Nishide, T., Yoneyama, K., and Ohta, K. “Attribute-based encryption with partially hidden encryptor-specified access structures”. In: *International Conference on Applied Cryptography and Network Security*. Springer. 2008, pp. 111–129.

[48] Li, J., Ren, K., Zhu, B., and Wan, Z. “Privacy-Aware Attribute-Based Encryption with User Accountability.” In: *ISC*. Vol. 9. Springer. 2009, pp. 347–362.

[49] Zhang, Y., Chen, X., Li, J., Wong, D. S., and Li, H. “Anonymous attribute-based encryption supporting efficient decryption test”. In: *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM. 2013, pp. 511–516.

[50] Rao, Y. S. and Dutta, R. “Recipient anonymous ciphertext-policy attribute based encryption”. In: *International Conference on Information Systems Security*. Springer. 2013, pp. 329–344.

[51] Boneh, D. and Silverberg, A. “Applications of multilinear forms to cryptography”. In: *Contemporary Mathematics* 324.1 (2003), pp. 71–90.

[52] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., and Waters, B. “Candidate indistinguishability obfuscation and functional encryption for all circuits”. In: *SIAM Journal on Computing* 45.3 (2016), pp. 882–929.

[53] Miles, E., Sahai, A., and Zhandry, M. “Secure obfuscation in a weak multilinear map model: A simple construction secure against all known attacks.” In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 588.

[54] Garg, S., Gentry, C., Halevi, S., Sahai, A., and Waters, B. “Attribute-based encryption for circuits from multilinear maps”. In: *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 479–499.

[55] Garg, S., Gentry, C., Halevi, S., and Zhandry, Mark. “Fully Secure Attribute Based Encryption from Multilinear Maps.” In: *IACR Cryptology EPrint Archive* 2014 (2014), p. 622.

[56] Gorbunov, S., Vaikuntanathan, V., and Wee, H. “Attribute-based encryption for circuits”. In: *Journal of the ACM (JACM)* 62.6 (2015), p. 45.

[57] Cheon, J.H., Han, K., Lee, C., Ryu, H., and Stehlé, D. “Cryptanalysis of the multilinear map over the integers”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2015, pp. 3–12.

[58] Albrecht, M. and Davidson, A. *Are graded encoding scheme broken yet*. 2017. URL: https://malb.io/are-graded-encoding-schemes-broken-yet.html.

[59] Fernando, R., Rasmussen, P. MR, and Sahai, A. “Preventing CLT Zeroizing Attacks on Obfuscation.” In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 1070.

[60] Boneh, D., Lewi, K., and Wu, D. J. “Constraining pseudorandom functions privately”. In: *IACR International Workshop on Public Key Cryptography*. Springer. 2017, pp. 494–524.

[61] Huang, M. and Raskind, W. “A multilinear generalization of the Tate pairing”. In: *Contemporary Mathematics* 518 (2010), pp. 255–263.

[62] Kamara, S., Papamanthou, C., and Roeder, T. “Dynamic searchable symmetric encryption”. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM. 2012, pp. 965–976.

[63] Van Liesdonk, P., Sedghi, S., Doumen, J., Hartel, P., and Jonker, W. “Computationally efficient searchable symmetric encryption”. In: *Workshop on Secure Data Management*. Springer. 2010, pp. 87–100.

[64] Baek, J., Safavi-Naini, R., and Susilo, W. “Public key encryption with keyword search revisited”. In: *International conference on Computational Science and Its Applications*. Springer. 2008, pp. 1249–1259.

[65] Liu, C., Zhu, L., Wang, M., and Tan, Y. “Search pattern leakage in searchable encryption: Attacks and new construction”. In: *Information Sciences* 265 (2014), pp. 176–188.

[66] Cash, D., Grubbs, P., Perry, J., and Ristenpart, T. “Leakage-abuse attacks against searchable encryption”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2015, pp. 668–679.

[67] Jarecki, S., Jutla, C., Krawczyk, H., Rosu, M., and Steiner, M. “Outsourced symmetric private information retrieval”. In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM. 2013, pp. 875–888.

[68] Zheng, Q., Xu, S., and Ateniese, G. “VABKS: verifiable attribute-based keyword search over outsourced encrypted data”. In: *Infocom, 2014 proceedings IEEE*. IEEE. 2014, pp. 522–530.

[69] Liu, P., Wang, J., Ma, H., and Nie, H. "Efficient verifiable public key encryption with keyword search based on KP-ABE". In: *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2014 Ninth International Conference on*. IEEE. 2014, pp. 584–589.

[70] Liang, K. and Susilo, W. "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage". In: *IEEE Transactions on Information Forensics and Security* 10.9 (2015), pp. 1981–1992.

[71] Li, J. and Zhang, L. "Attribute-based keyword search and data access control in cloud". In: *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on*. IEEE. 2014, pp. 382–386.

[72] Dong, Q., Guan, Z., and Chen, Z. "Attribute-based keyword search efficiency enhancement via an online/offline approach". In: *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on*. IEEE. 2015, pp. 298–305.

[73] Hu, B., Liu, Q., Liu, X., Peng, T., Wang, G., and Wu, J. "DABKS: Dynamic attribute-based keyword search in cloud computing". In: *Communications (ICC), 2017 IEEE International Conference on*. IEEE. 2017, pp. 1–6.

[74] Shi, J., Lai, J., Li, Y., Deng, R. H., and Weng, J. "Authorized keyword search on encrypted data". In: *European Symposium on Research in Computer Security*. Springer. 2014, pp. 419–435.

[75] Koo, D., Hur, J., and Yoon, H. "Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage". In: *Computers & Electrical Engineering* 39.1 (2013), pp. 34–46.

[76] Wang, H., Dong, X., and Cao, Z. "Multi-value-Independent Ciphertext-Policy Attribute Based Encryption with Fast Keyword Search". In: *IEEE Transactions on Services Computing* (2017).

[77] Frikken, K., Atallah, M., and Li, J. "Attribute-based access control with hidden policies and hidden credentials". In: *IEEE Transactions on Computers* 55.10 (2006), pp. 1259–1270.

[78] Boneh, D. and Waters, B. “Conjunctive, subset, and range queries on encrypted data”. In: *Theory of Cryptography Conference*. Springer. 2007, pp. 535–554.

[79] Katz, J., Sahai, A., and Waters, B. “Predicate encryption supporting disjunctions, polynomial equations, and inner products”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2008, pp. 146–162.

[80] Chaudhari, P. and Das, M. L. “On the Security of a Searchable Anonymous Attribute Based Encryption”. In: *International Conference on Mathematics and Computing*. Springer. 2017, pp. 16–25.

[81] Guillevic, A. “Comparing the pairing efficiency over composite-order and prime-order elliptic curves”. In: *International Conference on Applied Cryptography and Network Security*. Springer. 2013, pp. 357–372.

[82] Coron, J., Lepoint, T., and Tibouchi, M. “Practical multilinear maps over the integers”. In: *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 476–493.

[83] Coron, J., Lepoint, T., and Tibouchi, M. “New multilinear maps over the integers”. In: *Annual Cryptology Conference*. Springer. 2015, pp. 267–286.

[84] De Caro, A. and Iovino, V. “jPBC: Java pairing based cryptography”. In: *Computers and communications (ISCC), 2011 IEEE Symposium on*. IEEE. 2011, pp. 850–855.

[85] Cohen, W. W. “Enron email dataset”. In: (2009).

[86] Lynn, B. et al. “Pbc: The pairing-based cryptography library”. In: *http://crypto. stanford. edu/pbc* (2011).

[87] Frank, A. and Asuncion, A. “UCI Machine Learning Repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California”. In: *School of information and computer science* 213 (2010).

[88] Chaudhari, P., Das, M. L., and Mathuria, A. "On anonymous attribute based encryption". In: *International Conference on Information Systems Security*. Springer. 2015, pp. 378–392.

[89] Wang, C. and Huang, J. "Attribute-based signcryption with ciphertext-policy and claim-predicate mechanism". In: *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*. IEEE. 2011, pp. 905–909.

[90] Emura, K., Miyaji, A., and Rahman, M. S. "Dynamic attribute-based signcryption without random oracles". In: *International Journal of Applied Cryptography* 2.3 (2012), pp. 199–211.

[91] Wei, J., Hu, X., and Liu, W. "Traceable attribute-based signcryption". In: *Security and Communication Networks* 7.12 (2014), pp. 2302–2317.

[92] Pandit, T., Pandey, S. K., and Barua, R. "Attribute-based signcryption: Signer privacy, strong unforgeability and ind-cca2 security in adaptive-predicates attack". In: *International Conference on Provable Security*. Springer. 2014, pp. 274–290.

[93] Liu, J., Huang, X., and Liu, J. K. "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption". In: *Future Generation Computer Systems* 52 (2015), pp. 67–76.

[94] Hong, H. and Sun, Z. "An efficient and secure attribute based signcryption scheme with LSSS access structure". In: *SpringerPlus* 5.1 (2016), p. 644.

[95] Malone-Lee, J. "Identity-Based Signcryption." In: *IACR Cryptology ePrint Archive* 2002 (2002), p. 98.

[96] Libert, B. and Quisquater, J. "A new identity based signcryption scheme from pairings". In: *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*. IEEE. 2003, pp. 155–158.

[97] Chen, L. and Malone-Lee, J. "Improved Identity-Based Signcryption." In: *Public Key Cryptography*. Vol. 3386. Springer. 2005, pp. 362–379.

[98] Barreto, P. S., Libert, B., McCullagh, N., and Quisquater, J. "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2005, pp. 515–532.

[99] Boyen, X. "Multipurpose identity-based signcryption". In: *Crypto*. Vol. 3. Springer. 2003, pp. 383–399.

[100] Duan, S. and Cao, Z. "Efficient and provably secure multi-receiver identity-based signcryption". In: *ACISP*. Vol. 6. Springer. 2006, pp. 195–206.

[101] Ming, Y., Zhao, X., and Wang, Y. "Multi-receiver Identity-Based Signcryption Scheme in the Standard Model." In: *ICICA (LNCS)*. Springer. 2011, pp. 487–494.

[102] Pang, L., Gao, L., Li, H., and Wang, Y. "Anonymous multi-receiver ID-based signcryption scheme". In: *IET Information Security* 9.3 (2015), pp. 194–201.

[103] Tan, C. "On the security of provably secure multi-receiver ID-based signcryption scheme". In: *IEICE transactions on fundamentals of electronics, communications and computer sciences* 91.7 (2008), pp. 1836–1838.

[104] Guillevic, A. and Vergnaud, D. "Algorithms for outsourcing pairing computation". In: *International Conference on Smart Card Research and Advanced Applications*. Springer. 2014, pp. 193–211.

[105] El Mrabet, N., Ionica, S., and Guillermin, G. *Pairing computation at 192 bits level security*.

[106] Blaze, M., Bleumer, G., and Strauss, M. "Divertible protocols and atomic proxy cryptography". In: *Advances in Cryptology—EUROCRYPT'98* (1998), pp. 127–144.

[107] Yu, S., Wang, C., Ren, K., and Lou, W. "Attribute based data sharing with attribute revocation". In: *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ACM. 2010, pp. 261–270.

[108] Do, J., Song, Y., and Park, N. “Attribute based proxy re-encryption for data confidentiality in cloud computing environments”. In: *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*. IEEE. 2011, pp. 248–251.

[109] Yang, Y., Zhu, H., Lu, H., Weng, J., Zhang, Y., and Choo, K. R. “Cloud based data sharing with fine-grained proxy re-encryption”. In: *Pervasive and Mobile computing* 28 (2016), pp. 122–134.

[110] Li, H. and Pang, L. “Efficient and Adaptively Secure Attribute-Based Proxy Reencryption Scheme”. In: *International Journal of Distributed Sensor Networks* 12.5 (2016), p. 5235714.

[111] Zhang, Y., Li, J., Chen, X., and Li, H. “Anonymous attribute-based proxy re-encryption for access control in cloud computing”. In: *Security and Communication Networks* 9.14 (2016), pp. 2397–2411.

Appendix 1: Publications

Journals

- Chaudhari, P., and Das, M. L., "Privacy Preserving Searchable Encryption with Fine-grained Access Control", Submitted to a Journal (Under review).
- Chaudhari, P., and Das, M. L., "KeySeE: Keyword Search with Receiver Anonymity in Attribute-based Searchable Encryption", Submitted to a Journal (Under review).

Conferences

- Chaudhari, P., Das, M. L., and Dasgupta, D., "Privacy-Preserving Proxy Reencryption with Fine-Grained Access Control". In: International Conference on Information Systems Security. Springer. 2017, pp. 88 - 103. (**Awarded as Best Paper**)
- Chaudhari, P. and Das, M. L., "A^{2}BSE: Anonymous attribute based searchable encryption". In Asia Security and Privacy (ISEASP), ISEA, IEEE. 2017, pp. 1 - 10.
- Chaudhari, P. and Das, M. L. "On the Security of a Searchable Anonymous Attribute Based Encryption". In: International Conference on Mathematics and Computing. Springer. 2017, pp. 16 - 25.
- Chaudhari, P. and Das, M. L. "Privacy Preserving Signcryption Scheme". In: International Conference on Distributed Computing and Internet Technology. Springer. 2017, pp. 196 - 209.

- Chaudhari, P., Das, M. L., and Mathuria, A. “On anonymous attribute based encryption”. In: International Conference on Information Systems Security. Springer. 2015, pp. 378 - 392