

Thesis for the Degree of Doctor of Philosophy

**OBLIVIOUS COMPUTATION IN PUBLIC CLOUD
FOR PRIVACY-AWARE ACCESS CONTROL
POLICIES AND DATA SEARCH**

Zeeshan Pervez

**Department of Computer Engineering
Graduate School
Kyung Hee University
Seoul, Korea**

Fall 2012

OBLIVIOUS COMPUTATION IN PUBLIC CLOUD FOR PRIVACY-AWARE ACCESS CONTROL POLICIES AND DATA SEARCH

Zeeshan Pervez

**Department of Computer Engineering
Graduate School
Kyung Hee University
Seoul, Korea**

Fall 2012

OBLIVIOUS COMPUTATION IN PUBLIC CLOUD FOR PRIVACY-AWARE ACCESS CONTROL POLICIES AND DATA SEARCH

by

Zeeshan Pervez

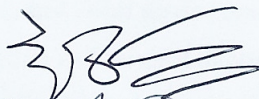
Supervised by

Prof. Sungyoung Lee, Ph.D.

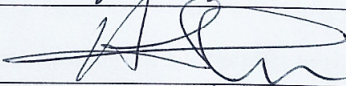
Submitted to the Department of Computer Engineering
and the Faculty of the Graduate School of
Kyung Hee University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Dissertation Committee:

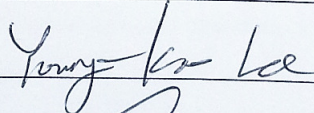
Prof. Choong Seon Hong, Ph.D.



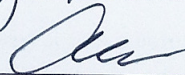
Prof. Eui-Nam Huh, Ph.D.



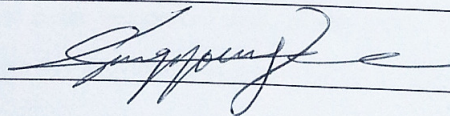
Prof. Young-Koo Lee, Ph.D.



Prof. Heejo Lee, Ph.D.



Prof. Sungyoung Lee, Ph.D.



Dedicated to my parents and brother along with his family.

This work is for, and because of you. I love you all.

Abstract

Cloud computing is an epitome of on-demand and scalable computing which provides computational power and storage capabilities on subscription basis, following the notion of pay-as-you-go. Numerous data storage services called *public cloud storage services* have emerged that changed the way we used to manage and interact with data outsourced to a public domain. With public cloud storage services, multiple subscribers can collaboratively work and share outsourced data without the concerns of data consistency, availability and reliability.

Since, public cloud storage services are provisioned by an untrusted cloud service provider that lies beyond the federated domain of subscribers, there is a great risk of privacy infringement when confidential data is outsourced to such services. Besides this, malicious subscribers can collude with cloud service provider to compromise privacy of the outsourced data along with the personal information of subscribers. This is because public cloud storage services enable subscribers to share their outsourced data with multiple subscribers each having its own access privileges.

To ensure data confidentiality in public cloud storage services, often encrypted data is outsourced to these services. It restrains untrusted cloud service provider and malicious subscribers to compromise privacy of the outsourced data and personal information of subscribers. Although encryption ensures data confidentiality; however, it fails to achieve fine-grained access control over outsourced data. It also limits the capability of a subscriber to search outsourced data by using conventional lookup queries. This is because relational operators cannot be applied to evaluate plain-text search queries for encrypted data.

Conventional procedures to enforce fine-grained access control rely on availability of a trusted entity that can govern and provision access to the outsourced data. Since, public cloud storage services are managed by untrusted cloud service provider, fine-grained access control cannot be

realized through access control policies. Besides this, access control policies also lose their practicality in public cloud storage services as these policies can be exploited by a untrusted cloud service provider to deduce confidential information about the outsourced data and subscriber's personal information.

Cloud service providers charge their subscribers according to amount of data accessed and network cost for data access requests. Thus, capability of a subscriber to search cloud storage plays a key role in maximizing the utility and minimizing the cost of managing data in public cloud storage services. However, existing methodologies to search encrypted data lack conformance to access control policies, and thus can be exploited by malicious subscribers to compromise privacy of the outsourced data.

To overcome the problems of privacy-aware authorized data access and search over encrypted data in untrusted domain, an oblivious computation called delegated private matching (DPM) is proposed in this dissertation. Unlike any of the conventional oblivious computation methodologies, DPM assists public cloud storage service's subscribers to utilize cloud infrastructure to process confidential information i.e., access control policy evaluation and searching encrypted data.

DPM provided the building block for the proposed Oblivious Access Control Policies (O-ACE) and Oblivious Term Matching (OTM). Both of these methodologies ensured privacy of the outsourced data as well as personal information of subscribers without relying on any trusted third party. O-ACE ensured that outsourced data is only accessible to authorized subscribers and malicious subscribers could not collude with cloud service provider. OTM realized a privacy-aware search over encrypted data by enabling authorized subscribers to define their search queries comprising of arbitrary number of search criteria.

Performance of proposed methodologies was analyzed on public cloud platform that provided computational and storage facilities i.e., Google App Engine. It highlighted the facts the proposed methodologies exerted amicable computational load on cloud infrastructure and could even be realized for low-end devices having limited compute resources. Security analysis showed that O-ACE and OTM were resilient to any conspired attack by malicious subscribers and cloud service provider. For unauthorized subscribers O-ACE and OTM generated randomized responses that did not reveal any information which can lead to potential loss of privacy.

Acknowledgement

First and foremost, my deepest gratitude to **Allah Almighty** for giving me the strength, courage, determination and perseverance to accomplish this milestone in my life. I am beholden for countless blessing bestowed upon me.

I would like to express my sincere gratitude to Professor Sungyoung Lee for providing me the opportunity to work with him. I could never have been able to achieve this milestone without your unceasing guidance, support and encouragement. I am grateful to my dissertation evaluation committee for their insight comments and valuable suggestions during the dissertation defense. It really helped me in elevating the quality of this dissertation.

I am very much thankful to the entire Pakistani community in Kyung Hee University for their time and support. I am highly obliged to Dr. Asad Masood Khattak, Dr. Muhammad Shoaib Siddiqui, Mr. Wajahat Ali Khan, Dr. Adil Mehmood Khan, Mr. Ozair Idrees Khan, Mr. Jalal Ahmad, Mr. Muhammad Hammed Siddiqui, Mr. Muhammad Fahim, Mr. Bilal Amin, Mr. Muhammad Waqas Nawaz, and Mr. Kifayat Ullah Khan for their continuous support and for making my stay in Korea, wonderful and memorable. I would also like to thank all of my Korean labmates for their help.

No form of thanks would suffice for my parents and brother along with his family for their unconditional love and countless prayers. I owe every bit of my success to you all. I would like to extend my gratitude to my cousins Mr. Arslan Ali and Mr. Furqan Ali for helping out my parents' in every aspect of life, in my absence. I am also obliged to Mr. Emad Tahir for talking to me from time to time and for his encouraging remarks.

Thank you everyone! and *Stay blessed*.

Suwon, December 2012

Zeeshan PERVEZ

Table of Contents

Abstract	i
Acknowledgment	iii
Table of Contents	iv
List of Figures	ix
List of Tables	xi
Chapter 1 Introduction	1
1.1 Overview	1
1.2 Motivation	3
1.2.1 Public Cloud Storage Services	3
1.2.2 Data Privacy Issues in Public Cloud Storage	6
1.2.2.1 Untrusted Cloud Service Provider	7
1.2.2.2 Lack of Control Over Cloud Infrastructure	8
1.2.2.3 Malicious Subscribers	9
1.2.2.4 Limitations of Existing Methodologies	9
1.3 Problem Statement	10
1.3.1 Limitations of Conventional Access Control Enforcement in Untrusted Domain	12
1.3.1.1 Reliance on Cloud Service Provider	15
1.3.1.2 Reliance on Trusted Third Party	15

1.3.1.3	Credential Leakage and Privacy Infringement	15
1.3.1.4	Malicious Subscribers	16
1.3.2	Limitations of Conventional Encrypted Data Search in Untrusted Domain	16
1.3.2.1	Limited Number of Trapdoors and Availability of Data Owner	16
1.3.2.2	Reliance on Trusted Third Party	17
1.3.2.3	Lack of Conformance with Access Control Policy Enforcement	17
1.4	Contributions	17
1.4.1	Oblivious Access Control Policies	18
1.4.2	Oblivious Data Search	21
1.5	Structure of Dissertation	23
Chapter 2	Related Work	25
2.1	Privacy	25
2.2	Data Privacy	26
2.3	Cloud Storage Service and Data Privacy	27
2.3.1	Authorized Data Access in Cloud Storage Services	27
2.3.1.1	Access Control Enforcement by Trusted Third Party	28
2.3.1.2	Access Control Enforcement with Attribute based Encryption .	30
2.3.1.3	Access Control Enforcement by Data Owner	31
2.3.2	Limitations of Conventional Access Control Enforcement within Cloud Storage Services	32
2.3.3	Privacy-aware Search in Cloud Storage Services	36
2.3.3.1	Trapdoor-based Search Queries for Encrypted Data	36
2.3.3.2	Index-based Search Queries for Confidential Data	37
2.3.4	Limitations of Existing Methodologies to Search Encrypted Data Out- sourced to Cloud Storage Services	39
Chapter 3	Preliminaries	41
3.1	Homomorphic Encryption	41
3.1.1	Key Generation	41

3.1.2	Encryption	42
3.1.3	Decryption	43
3.1.4	Homomorphic Operation	43
3.2	Private Matching	44
3.3	Proxy Re-Encryption (PRE)	46
3.3.1	Key Generation	46
3.3.2	Encryption	46
3.3.3	Re-Encryption	46
3.3.4	Decryption	47
Chapter 4	Delegated Private Matching	48
4.1	Availability Requirement for Involved Entities	48
4.2	Oblivious Private Matching in Untrusted Domain	49
4.3	Security Analysis	50
4.3.1	Malicious Client	53
4.3.2	Malicious Validator	53
Chapter 5	Oblivious Access Control Policy Evaluation	54
5.1	Introduction	54
5.2	Models, Design Goals and Assumptions	54
5.2.1	Conceptual Model	55
5.2.2	Security Model	55
5.2.3	System Design Goals	56
5.2.4	Assumption and Notations	57
5.3	Application Scenario and Abstract Idea of Oblivious Access Control Policy Evaluation	59
5.4	Enforcing Oblivious Access Control Policy for Cloud-based Data Sharing	60
5.4.1	Initialization	63
5.4.2	Data Outsourcing	63
5.4.3	File Access	64

5.5	Complexity Analysis	65
5.5.1	Policy Modeling	66
5.5.2	Data Access Request	66
5.5.3	Access Control Evaluation	67
5.5.4	Mandatory Value Recovery	67
5.6	Security Analysis	67
5.6.1	Malicious Cloud Server	68
5.6.2	Malicious Clients	69
5.6.3	Malicious Identity Provider	69
5.7	Implementation	70
5.8	Evaluation	71
5.8.1	Phase 1: Performance Analysis of Access Control Policy Evaluation on Google AppEngine	72
5.8.2	Phase 2: Performance Analysis of Data Owner and Client Components .	72
5.9	Discussion	77
5.10	Summary	79
Chapter 6	Oblivious Data Search in Cloud Storage	81
6.1	Introduction	81
6.2	Models, Design Goals, and Assumptions	81
6.2.1	Conceptual Model	82
6.2.2	Security Model	82
6.2.3	System Design Goals	83
6.3	Application scenario and abstract idea for searching encrypted data in cloud storage	84
6.4	Searching cloud storage with oblivious term matching (OTM)	85
6.4.1	Setup	85
6.4.2	Data Outsourcing	88
6.4.3	Query Generation	89
6.4.4	Searching	89
6.4.5	Response Extraction	90

6.5	Complexity Analysis	91
6.5.1	Index Outsourcing	91
6.5.2	Query Generation	92
6.5.3	Query Modeling	92
6.5.4	Query Evaluation	92
6.5.5	Oblivious Result Processing	93
6.5.6	Result Extraction	93
6.6	Security Analysis	93
6.6.1	Malicious Cloud Server	94
6.6.2	Malicious Subscriber	95
6.6.3	Malicious Third Party	95
6.7	Implementation	96
6.8	Evaluation	97
6.8.1	Inverted Index and Search Criteria Generation and Processing	98
6.8.2	Oblivious Query Modelling and Response Extraction	98
6.8.3	Oblivious Query Evaluation on Google App Engine	101
6.9	Enhancing Encrypted Search Data with Post-Processing	106
6.10	Discussion	108
6.11	Summary	111
Chapter 7	Conclusion and Future Directions	113
7.1	Conclusion	113
7.1.1	Access Control Enforcement in Cloud Storage Services	114
7.1.2	Encrypted Data Search for Cloud Storage Services	114
7.2	Future Directions	115
	Bibliography	116
Appendix A	List of Publications	127

List of Figures

1.1	Worldwide forecast for personal cloud subscriptions (in millions)	4
1.2	Public cloud storage services - privacy and storage capacity	5
1.3	Data sharing, synchronization and collaborative services with cloud storage . . .	6
1.4	Potential loss of privacy with access control policies	13
1.5	Potential loss of privacy with search over encrypted data	14
1.6	Extending private matching to delegated private matching - abstract view	19
1.7	Oblivious access control policy evaluation	20
1.8	Oblivious data search on encrypted outsourced data	22
2.1	Access control enforcement by relying on trusted third party or services	29
2.2	Access control enforcement with attribute based encryption	30
2.3	Access control enforcement by relying on data owner's availability	31
2.4	Data encrypted with attribute based encryption	35
2.5	Attribute based encryption secret key	35
2.6	Trapdoor-based encrypted data search for cloud storage services.	37
2.7	Index-based data search for cloud storage services.	38
3.1	Private matching: entity interaction and time-line.	45
4.1	Delegate private matching: entity interaction and time-line.	51
5.1	Oblivious access control policy evaluation (O-ACE) for cloud-based data sharing.	61
5.2	Exchange of private values between the owner, cloud server and user during O-ACE.	62

5.3	Computational time and cost of access control policy evaluation on Google App Engine.	73
5.4	Computational time required to model the access control policy.	74
5.5	Computational time required to process the identity attributes.	75
5.6	Computational time required to recovery mandatory values.	76
6.1	Abstract model of privacy aware oblivious data search in a cloud storage.	86
6.2	Inverted index generation and indexed term encryption time.	99
6.3	Search criteria encryption and decryption time.	100
6.4	Query modelling, oblivious query generation encryption and response extraction time.	102
6.5	Oblivious query evaluation time, cloud server response time and estimated execution cost for 1000 requests.	103
6.6	Computation time (ms) required to evaluate oblivious term matching on F1, F2 and F4 Frontend instances of Google App Engine.	104
6.7	Response time (ms) of oblivious term matching on F1, F2 and F4 Frontend instances of Google App Engine.	105
6.8	CPU Time (ms) of oblivious term matching on F1, F2 and F4 Frontend instances of Google App Engine.	105
6.9	Extended functionalities - oblivious term matching conceptual model.	107

List of Tables

2.1	Limitations of conventional access control methodologies.	34
2.2	Limitations of conventional methodologies to search untrusted storage services. .	40
5.1	Notations used in the descriptive detail of O-ACE.	58
5.2	O-ACE computational complexity and estimation of transmitted values.	66
6.1	Notations used in the descriptive detail of OTM.	87
6.2	OTM computational complexity and estimation of transmitted values.	91

1.1 Overview

We are living through Post-PC era in which computational power and storage facilities are available as limitless subscription based services and are not confined to local machines or dedicated servers [1, 2]. Software delivery models have evolved too, and are no longer bound to perform their functionalities on specific machines on which they are installed [3]. For individuals Post-PC era offers subscription based computing; whereas, for organizations it presents an opportunity to adopt notion of economies of scale when formulating strategies that directly or indirectly relate to information technology [4]. Besides the usage model, this new style of computing greatly affects the management of information technology resources. Individuals as well as organizations can focus on their core competencies and can evade the hassle of maintaining their information technology infrastructure according to their computational needs [5].

This new computing paradigm of on-demand computing is called *Cloud Computing* [6]. Emergence of virtualization technologies, availability of high-speed Internet and adoption of Service Oriented Architecture (SOA) are the enabling trends that derive cloud computing [7, 8]. Benefits provided by this on-demand, massive, and scalable computing facility can be primarily lumped into one category - cost [9–11]. Annually Gartner publishes a list of organizational strategic technologies; for the past two years, it has rated cloud computing as one of the core technologies, which organizations must consider during their strategic planning for the next ten years [12]. With the emergence of cloud computing, whole new business models and services have been developed which were either considered impractical or had considerably low return on investment due to their intrinsic requirements of massive information technology infrastructure.

According to the deployment of information technology infrastructure, and resource config-

uration that provision on-demand computing, cloud computing can be divided into four distinct deployment models i.e., public, private, hybrid and community cloud [13]. The main factors that carve out distinction between these models are the ownership, management and operation of underlying cloud infrastructure (computational power, storage capacity, network facility and software services) by Cloud Service Provider(s) (CSP), and configuration in which resources are interconnected with each other [14, 15]. Regardless of the responsibility of a CSP, and configuration of resources the central theme of cloud computing remains the same in each deployment model i.e., on-demand availability of virtualized resources / services that can be consumed over the Internet on subscription basis, adhering to the notion of pay-as-you-use [16].

In public cloud, cloud infrastructure is owned, managed, and operated by a CSP. Availability of services provisioned by a public cloud is guaranteed by the CSP, and often described in service contracts. Multiple subscribers can subscribe to these services without interfering each other service domain i.e., processing, storage, or application logic. In case of private cloud, services are provisioned for exclusive use only, and are intended to be consumed by a single subscriber i.e., an individual user or an organization. In private cloud, cloud infrastructure can be owned, managed, and operated by a subscriber, or by a third party. Hybrid cloud is a combination of two or more cloud infrastructures (public or private) to serve a common interest. The synergy among the clouds is of assistive nature and meant to complement each other's functionality. Cloud infrastructures are owned, managed, and operated by individual CSPs; however, they are bound together by a standard or proprietary technology that leverages the common interest. In community cloud, cloud infrastructure is provisioned to a specific community of subscribers belonging to multiple organizations. The community of subscribers serves the shared interest of respective organizations. Cloud infrastructure in community cloud can be owned, managed, and operated by a single, multiple organizations that form the community, or by a third party.

Fundamentally, cloud computing can provision services in three distinctive deployment models i.e., Infrastructure-as-a-Service (IaaS), Platform-as-a-service (PaaS), and Software-as-a-service (SaaS) [17, 18]. IaaS provides bare compute resources (computational power, storage capacity) to its subscribers according to their computational and persistence requirements. In IaaS subscribers can choose operating system, storage and provisioning applications of their own

choice; whereas, CSP is responsible for the availability of cloud infrastructure. PaaS leverages its subscribers with computing environment on which subscribers build custom application specific to their business needs. Underlying cloud infrastructure is managed by a CSP. Tools, software packages, libraries, and services specific to PaaS are provided by the CSP to harness cloud infrastructure during different phases of application life-cycle i.e., development, deployment and provisioning. SaaS provides software solutions that replace in-house applications managed locally by subscribers. It enables subscribers to choose functionalities or service modules suitable to their needs. Through SaaS, CSP provisions a single service to multiple subscribers, each customized to meet specific needs of a subscriber.

Generally, services provisioned by various deployment models of cloud computing, possess the characteristics of on-demand availability, broad accessibility, resource pooling, rapid elasticity, and service metering [19]. The essence of cloud services is accessibility over heterogeneous platforms, enabling subscribers to consume them over various devices (e.g., smartphones, tablets, and workstations), and over the Internet. These services are developed and provisioned as multi-tenant services, enabling CSP to develop service once and provision it to multiple subscribers. The economics of scale comes with rapid elasticity. Subscribers can instantaneously scale services to meet their computational requirements. They can also measure their services usage, and can formulate their scaling strategy accordingly.

1.2 Motivation

1.2.1 Public Cloud Storage Services

With the amount of digital data doubling almost every eighteen months, cloud storage provides a cost-effective solution to deal with the ever-increasing demand of storage facility [20]. It provides raw storage as a service on subscription basis, which can be scaled-up or scaled-down instantly according to the requirements. CSPs operate large data centers, and provision virtualized pool of storage as object store or data store. These stores are accessible via application programming interface, which leverages application developers to build custom applications (web, desktop, smartphone services) to unleash the potential of on-demand and seamless availability of

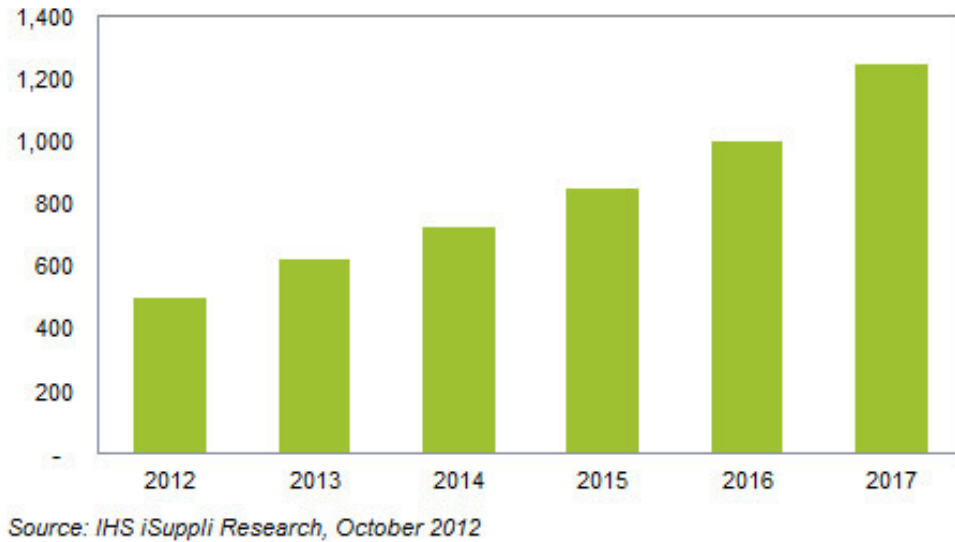


Figure 1.1: Worldwide forecast for personal cloud subscriptions (in millions)

virtually unlimited integrated storage capacity.

According to Gartner, by year 2016 cloud storage subscribers will be outsourcing 36% of their digital content to cloud storage services [21]. Another study by International Data Corporation, has projected that by year 2015 the combined volume of IT spending on public and private cloud storage will exceed 20\$ billion worldwide [22]. According to market intelligence firm iHS iSuppli, until October 2012 the total number of paid and free cloud storage subscribers is around 375 million, and will likely reach 500 million by the end of the year [23]. CSPs are expected to engage 625 million active subscribers in 2013. Figure 1.1 shows the projected number of public cloud storage subscribers for the next five years. Figure 1.2 illustrates the pricing and storage capacity of public cloud storage services [24–30] available at [31].

The advent of cloud storage has brought forth some interesting applications and services, and has certainly changed the way we manage and interact with the outsourced data [32]. Data backup, synchronization, sharing and collaborative services are the most prevalent services which avail benefits of cloud storage [33]. Subscribers of these services, including both individuals and enterprises, can keep their data for much longer time without the concerns of reliability and availability of the outsourced data. At very primitive level, these services untether the data from any specific medium, platform or location and provision it in a consistent manner across all devices and views.

Name	Price per month (\$)	Storage Capacity (GB)
 SugarSync	4.17	30
 mozy	4.45	50
 just cloud .com	4.49	75
 SPIDEROAK	8.33	100
 Google Drive	2.49	25
 Microsoft SkyDrive	0.83	27
 Dropbox	9.99	50

Figure 1.2: Public cloud storage services - privacy and storage capacity

Figure 1.3 shows the conceptual model of clouds storage services that enable data owner to share, collaborate and synchronize outsourced data with other subscribers.

Data backup services persist data on cloud storage, and take away the hassle of managing and scheduling data backup jobs, and maintaining data archives. Synchronization services work similar to data backup services; however, these services serve a specialized need of data consistency among different subscribers and devices. Instead of scheduled data backup, these services only transmit changed data contents i.e. addition, deletion and modification of data contents. This ensures that changes are reflected across all views (i.e., devices and subscribers) immediately and minimum amount of data is transferred to accommodate changes. Sharing and collaborative services take cloud storage to a next level of subscribers' engagement. Subscribers can share data among themselves and can update it in a collaborative manner i.e., add, delete, update data contents, accept or reject changes, post comments. Changes are reflected in real-time across all collaborating subscribers and the underlying framework that powers sharing and collaborative services

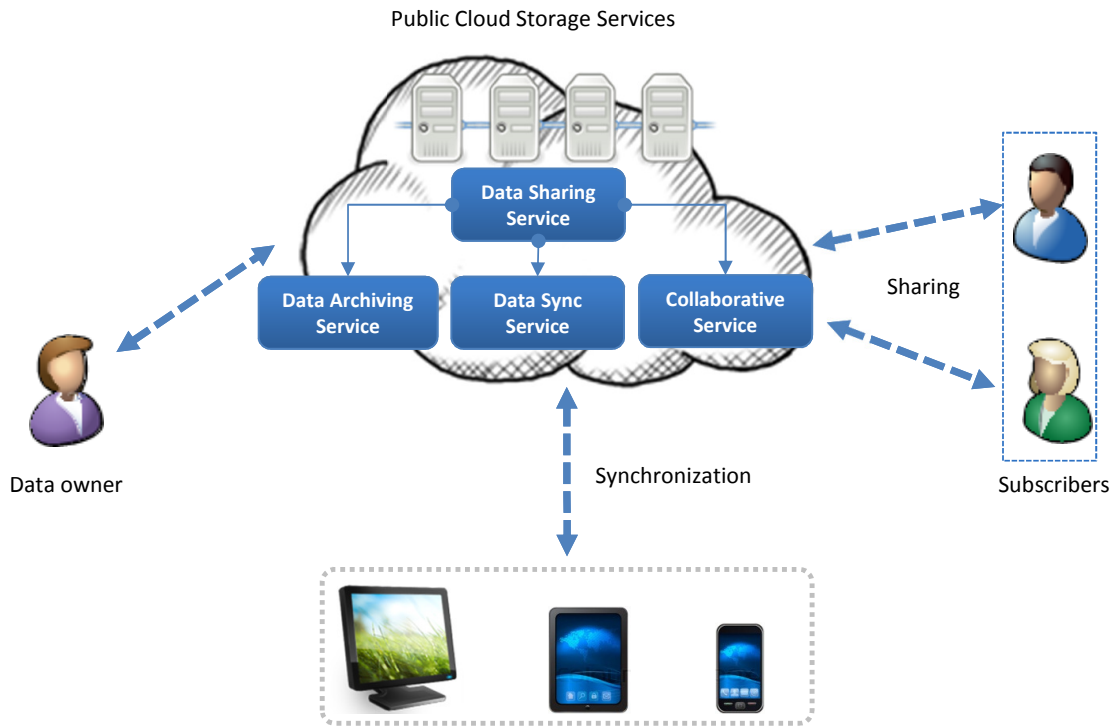


Figure 1.3: Data sharing, synchronization and collaborative services with cloud storage

resolves conflicts seamlessly without any human intervention.

1.2.2 Data Privacy Issues in Public Cloud Storage

Public cloud storage is an online storage facility, which can be accessed through Internet. Around the ecosystem of public cloud storage, myriad services have been developed which process, persist and provision the data of their subscribers. Public cloud storage is a closed system, and often very limited information about the cloud infrastructure is available to its subscribers [34]. Consequently, subscribers are unaware about the security measures adopted by a CSP, how often they are evaluated, and how well these security measures conform to standards and government regulations [35–38]. CSP are bound to provide services according to service contracts. These service must be provisioned according to service level agreement signed between CSP and its subscribers.

Since, CSP provisions the underlying cloud infrastructure that powers cloud storage services,

there is a great risk of privacy infringement when data is outsourced to these services [39–42]. This is because cloud infrastructure lies beyond the federated domain of a subscriber, and often CSP is least trusted when confidential or personal data (e.g., medical records, bank statements, family photo album etc.) is outsourced to these services. A malicious and curious CSP can compromise privacy of a subscriber by learning confidential as well as personal information from the outsourced data. Thus, not only outsourced data is prone to privacy attacks, personal information can be compromised if cloud storage services are carelessly used. CSP provisioning public cloud storage services can exploit encryption and key exchange algorithms providing weak cryptographic guarantees, inadequate enforcement of access control policies, and capability of a subscriber to share data with other subscribers.

Most of the privacy concerns that arise with the adoption of cloud storage services are mainly because CSP is considered as an untrusted entity. In the following we present privacy concerns that are directly related to lack of control over cloud infrastructure and security measures that govern access to the outsourced data persisted by cloud storage services.

1.2.2.1 Untrusted Cloud Service Provider

Information technology service providers having financial and technical resources to own, manage and operate massive data centers provision cloud storage services. These service providers make their business by leasing out cloud infrastructure to their subscribers. In the context of cloud storage services subscribers utilize these resources to manage their data - collaborate, share, and achieve data contents. Nevertheless, the biggest hurdle in the adoption of cloud storage services is lack of trust on service providers [43–46]. This is because, often internal details about the operation and management of cloud infrastructure is not available to subscribers.

Often CSPs utilize proprietary software and hardware, revealing information regarding the underlying cloud infrastructure can jeopardize their business. Competitors can exploit this information to formulate cost effective subscription models to attract more subscribers. However, this lack of information, greatly affects the trustworthiness of a CSP, from security and privacy point of view. For a subscriber it is nontrivial to ensure that CSP is following security and privacy standards and all other necessary steps are taken to prevent potential loss of data as well as personal

privacy.

United States of America is spending 15.2% of its GDP on Healthcare [47]. To reduce operational expenses of information technology resources the government is encouraging business in healthcare to adopt cloud computing. However, according to United States Department of Health and Human Services (HHS), 62% of privacy breaches are mainly because of Business Associates (BA) [48], [49]. BAs assist healthcare service providers in their day-to-day and business activities. In case, if BA is a CSP provisioning public cloud storage services, it can exploit its capabilities (i.e., process, persist and provisioning of the outsourced) to compromise privacy of the outsourced medical records. Thus, while adopting public cloud storage services, the level of trust that subscribers have on CSP is greatly influenced by the sensitivity of the outsourced data. While dealing with confidential data (i.e., medical records, financial statements), it is obvious that subscribers cannot trust CSP, as CSP may be driven by malicious intent to compromise privacy of the outsourced data and subscriber as well.

1.2.2.2 Lack of Control Over Cloud Infrastructure

Management and operation of data centers is a business secret, and cloud service providers do not share any information related to their underlying cloud infrastructure. Consequently, subscribers cannot determine whether current security and privacy practices adopted by a cloud service provider are adequate, according to industry standards, and conform to government regulations and policies. In cloud storage services, subscribers have no control over the underlying cloud infrastructure as these services are provisioned beyond the federated domain of subscribers [13, 34, 50–52].

In most of cloud storage services, subscribers do not know the exact physical location of their outsourced data. However, in United States of America government regulations (i.e., Health Insurance Portability and Accountability Act - HIPAA) prevent medical records to be transmitted or persisted outside their jurisdiction [53]. Thus, if a hospital or healthcare service provider subscribes to a cloud storage service, it is his responsibility to ensure exact physical location of the outsourced medical records. Cloud storage services are provisioned as virtualized pool of storage. Often these pools are replicated to ensure uninterrupted provisioning in case of disaster or data

center down time. Clearly, it is nontrivial (in most cases not allowed), for a subscriber to track the outsourced data, as CSP is exclusively responsible for the management of cloud infrastructure.

1.2.2.3 Malicious Subscribers

Although cloud storage services are provisioned by untrusted CSP; however, CSP is not the only entity that is capable of compromising privacy of the outsourced data. Especially, in data sharing and collaborate services, malicious subscribers are equally capable of compromising privacy of the outsourced data [39,54]. They either can collude with untrusted CSP, or can exploit vulnerabilities of cloud infrastructure to gain access to the outsourced data. Even if malicious subscribers do not manage to gain access to the outsourced data, it can still learn confidential information that can lead to potential loss of privacy, data as well as personal information. By executing unauthorized search queries, malicious subscribers can learn presence or absence of particular keywords.

Often encrypted data is outsourced to cloud storage services to ensure data confidentiality. Encryption restrains CSP from learning any information about the outsourced data. However, it greatly decreases utility of cloud storage services, as availability of trusted third parties would be required to distribute data decryption keys to authorized users.

1.2.2.4 Limitations of Existing Methodologies

Public cloud storage services introduce a unique challenge within the domain of untrusted storage systems. It is unique in a sense that all of the involved entities (i.e., cloud service provider, subscribers seeking access to the outsourced data) can behave maliciously. CSP, provisioning the outsourced data can assist unauthorized subscribers to gain illicit data access or learn confidential information that can lead to potential loss of privacy. Whereas, subscribers can exploit data sharing and collaborative functionalities of a cloud storage service, complimented with malicious intent of CSP, to compromise privacy of the outsourced data.

Existing privacy and security measures either consider storage service provider as a trustable entity or rely on trusted third party and availability of data owner to govern data access [55]. In public cloud storage services, subscribers cannot rely on CSP to enforce access control policies, ensuring that only authorized subscribers manage to gain access to the outsourced data - lack of

trust on CSP and lack of control over the cloud infrastructure. Reliance on trusted third party would be a nontrivial task when collaborating subscribers belong from different domain of trust e.g., financial institutes and insurance company, or data is shared between subscribers having different regulations for trustable entities e.g., healthcare service provider and automobile manufacture. Whereas, rely on data owner to govern data access would affect the data sharing and collaborating functionalities as data owner would have to ensure its availability to allow or deny access of the subscribers.

Besides this, existing methodologies mainly focus on data confidentiality and key exchange algorithms. This is because encryption ensures that CSP and unauthorized subscribers cannot learn any information about the outsourced data. Whereas, key exchange algorithms guarantees secure dissemination of data encryption keys to authorized subscribers. However, these methodologies fail to enforce fine-grained access control over the outsourced data. Few of the methodologies tried to fuse access control policy enforcement with data decryption process; however, the process of access control enforcement can itself be compromised by malicious CSPs, enabling them to infer confidential information about the subscriber and outsourced data.

1.3 Problem Statement

Public cloud storage services provide untrusted storage facility that lies beyond the federated domain of its subscribers. Since, it is owned, managed, and operated by CSP, subscribers have limited and often no control over the data outsourced to a public cloud storage service. Nevertheless, its adoption is driven by the need of economic of scale. Subscribers can subscribe to virtually unlimited storage facility on subscription basis that can be scaled according to their usage requirements. With data outsourced to public cloud storage, subscribers do not need to concern about its availability, all issues related to data management (i.e., recovery, replication, synchronization, conflict resolution, auditability etc.) are seamlessly handled by the underlying framework that realizes cloud storage services.

Myriad services have been developed which utilize public cloud storage to process, persist and provision outsourced data. These services have greatly affected the way we manage and share our digital contents across different geographical locations, on varied devices and among groups of

people. Although, user experience from data interaction's point of view has been greatly influenced by public cloud storage services, however there is a great concern of privacy infringement (i.e., data and as well as personal) associated with the adoption of services provided by untrusted CSP. There are various factors, which give rise to these concerns. Lack of trust on CSP, multi-tenancy and malicious intents of subscribers and closed nature of cloud infrastructure hinder the adoption of public cloud storage services for persisting confidential and personal information.

Since, the prime focus of this thesis is on privacy issues related to public cloud storage services; we consider that outsourced data contain personal as well as confidential information. To ensure data confidentiality data is encrypted locally, and then outsourced to a cloud storage service provisioned by an untrusted CSP. We also consider that CSP is interested in learning or deducing information related to outsourced data and data owner (i.e., subscribers who outsources data to cloud storage). The motives of CSP could be evil or driven by the need to increase user experience i.e., target advertisement, service recommendation etc. CSP can also collude with other malicious subscribers to learn confidential and personal information of a particular subscriber. In the following, we present two distinct scenarios of public cloud usage along with potential loss of data privacy as well as personal information. These scenarios depict typical usage model of a cloud service using public cloud infrastructure as persistence and provisioning layer.

Suppose Queen's Hospital in downtown Seoul area is a very busy hospital. On daily basis hundreds of patients have their appointments with medical doctors. Various clinical tests are carried out and medicines are prescribed to patients, resulting in huge amount of data sharing between medical doctors and patients. To cope with the ever-increasing demand of storage capacity and computational capabilities, hospital management has opted for a public cloud storage service provided by Eve. Since, hospital deals with confidential as well as personal data, encrypted data is outsourced to the Eve's cloud storage. To ensure timely exchange of medical reports, daily symptoms, and recommendation between patients and medical doctors, Queen's hospital has developed a service called MP-Connect. It utilizes Eve's public cloud as data and compute layer.

In the following, we present two scenarios illustrating the capabilities of subscribers (i.e., medical doctor and patient) to share data with other subscribers. These scenarios illustrate the work-flows of online health management systems that enable medical doctors and patients to ex-

change clinical information with each other. It also enables medical doctors to collaborate with each other in order to deal with critical cases.

Scenario 01: data sharing and access control policies Alice is a medical doctor working in Queen's Hospital, specialized in diabetes mellitus. Bob is a diabetic patient, and consulting Alice on regular basis. Both Alice and Bob exchange clinical information on MP-Connect. Since, Alice is dealing with multiple patients; she defines access control policies for each of her patient and provisions access to each patient accordingly. Bob only wants to share his medical reports with Alice. He defines access control policy to enable Alice to access his medical report and clinical feedback.

Scenario 02: data sharing and searching capabilities Similar to the previous scenario, Alice is a medical doctor at Queen's Hospital. For critical cases, she seeks advice from Mallory, who happens to be a senior medical doctor working in Queen's. Both Alice and Mallory share data i.e., medical reports, clinical symptoms and feedback, of patients showing critical symptoms. Whenever Alice wants to get opinion on a particular case, she grants access to Mallory on related data. Mallory can search shared data for particular keywords and can access it according to her access privileges.

Scenarios presented above can be exploited by a CSP and malicious subscribers to compromise privacy of a subscriber, if access control policies and data searching functionalities are realized with conventional privacy preserving methodologies. This is because, CSP is an untrusted entity and access control policy enforcement and data searching functionality have a tendency to reveal confidential and personal information that can be used to compromise privacy of a subscriber and outsourced data as well. Figure 1.4 and 1.5 illustrate the possibilities of privacy infringement with aforementioned scenarios.

1.3.1 Limitations of Conventional Access Control Enforcement in Untrusted Domain

Conventional access control enforcement methodologies are not designed to govern data access in an untrusted domain [56, 57]. In the following we generally present limitation of existing methodologies within the context of cloud storage services. These limitations will be discussed at

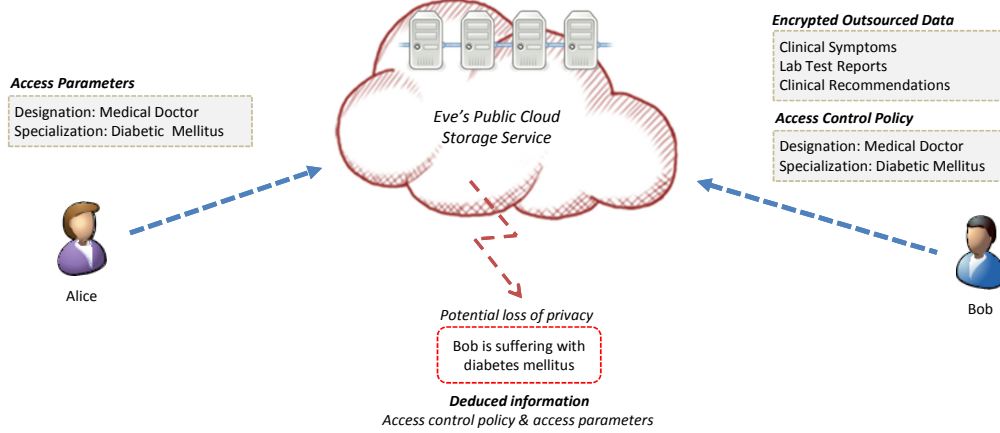


Figure 1.4: Potential loss of privacy with access control policies

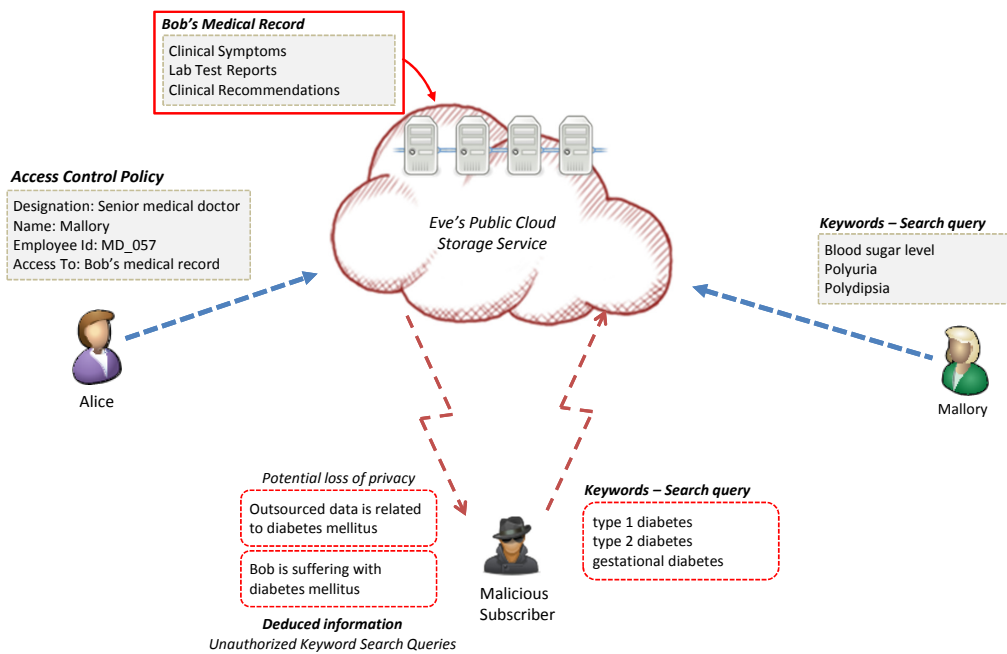


Figure 1.5: Potential loss of privacy with search over encrypted data

length in the subsequent chapter.

1.3.1.1 Reliance on Cloud Service Provider

Untrusted CSP provisions public cloud storage services. Enabling CSP to enforce access control policy would inevitably compromise privacy, as it can behave maliciously and can assist unauthorized subscribers to gain access to the outsourced data. Beside this, CSP can monitor access pattern of subscribers and deduce confidential and personal information about the outsourced data and subscriber as well e.g., medical records of a patient accessed by different medical doctors, cloud service provider can deduce information related to disease of a patient etc.

1.3.1.2 Reliance on Trusted Third Party

Since, CSP cannot be trusted to ensure privacy of confidential and personal information, the most obvious solution is to rely on trusted third party. In the context of public cloud storage services, trusted third party ensures that subscribers manage to gain access to the outsourced data according to their access privilege - by distributing appropriate data encryption keys. Introducing trusted third party greatly reduces utility of a public cloud storage service. The fundamental concept of public cloud storage service is to scale accordingly to computational and storage requirements. In this context, the capabilities of a trusted third party must follow the same notion of scalability, with number of subscribers and frequency of data access requirement.

1.3.1.3 Credential Leakage and Privacy Infringement

In most of cases, access privileges of a subscriber are associated with its credentials. In existing access control policy enforcement methodologies, credentials are revealed to policy evaluator. If CSP is equipped with access control policy evaluation and enforcement, it could lead to potential loss of privacy. In scenario illustrated in Figure 1.4, if Eve can learn that Alice specializes in diabetes mellitus then it can be deduced that patients visiting her are most likely to be diabetics; consequently it is a privacy infringement on Bob's personal information.

1.3.1.4 Malicious Subscribers

Public cloud storage services follow the notion of multi-tenancy - multiple subscribers share same cloud infrastructure. In public cloud storage services, malicious subscribers can collude with each other or even with CSP to compromise privacy of the outsourced data and data owner as well. Malicious subscribers can work together to gain access to the outsourced data by combining their access privileges - cloud service provider can assist them by revealing information related to access control policy i.e., set of valid access parameters that can successfully bypass access control policy evaluation process.

1.3.2 Limitations of Conventional Encrypted Data Search in Untrusted Domain

Search over encrypted data enables data searching capabilities without the need to decrypt concealed data contents. Although it ensures data privacy within untrusted domain; however, when utilized within the context of data sharing and collaborate services it greatly affects the utility of underlying services. In the following, we present limitations of conventional methodologies to search encrypted data outsourced to cloud storage services - detailed discussion is presented in subsequent chapter.

1.3.2.1 Limited Number of Trapdoors and Availability of Data Owner

Since, encrypted data is outsourced to public cloud storage services to ensure data confidentiality, conventional searching methodologies are not applicable to search data contents in encrypted form. At very primitive level encryption scrambles data contents with some auxiliary information (i.e., data encryption key), thus conventional term matching methodologies are not applicable to identity similarities between search query and encrypted data. To search encrypted data, trapdoors are defined which are then used to search for particular keywords. Although, searching encrypted data with trapdoor ensures data privacy; however, it limits the searching capability of a subscriber to a limited number of trapdoors. Normally trapdoors are distributed by the entity that encrypts the data, thus its availability must be assured for the distribution of valid trapdoors to legitimate and authorized subscribers.

1.3.2.2 Reliance on Trusted Third Party

Indexing outsourced data, and then executing search query on index ensures that encrypted data can be searched without using trapdoors. Since, index contains information about the outsourced data in plain text format, it must be stored by a trusted third party. Persisting index at trusted location not only added complexity in managing data updates it also greatly affects the utility of a cloud storage services. Since, public cloud storage provides the abstraction of unlimited storage facility that is scalable according to requirement; trusted third party must also conform to the same notion of scalability.

1.3.2.3 Lack of Conformance with Access Control Policy Enforcement

Enabling search over outsourced data can lead to potential loss of data privacy. Unauthorized subscribers can query outsourced data and deduce confidential as well as personal information. In the scenario presented in Figure 1.5, Mallory can query Alice's cloud space without her consent and deduce information related to patient's by learning presence or absence of particular keywords. To ensure privacy when enabling data search over cloud storage services, searching queries must conform to access control policies. However, as presented above CSP is an untrusted entity and trusted third party greatly affects the utility of cloud storage services, conformance of search query to access control policy cannot be realized by using conventional methodologies.

1.4 Contributions

In this dissertation, a private matching protocol is presented that enable cloud storage service's subscribers to delegate computational and persistence capabilities to an untrusted CSP with privacy considerations. The proposed delegated private matching protocol overcomes the aforementioned limitations of public cloud storage services. Delegated private matching protocol is an extension of private matching protocol (see Figure 1.6(a)). It enables entities to delegate their value matching capabilities to an untrusted entity (i.e., Cloud Service Provider) without losing privacy of the involved data (see Figure 1.6(b)). It uses cryptographic primitives to prevent potential loss of data privacy even if CSP and subscribers behave maliciously. It also restrains entities with malicious

intents to deduce confidential as well as personal information about the data owner. Figure 1.6(b) illustrates the delegated private matching protocol.

Based on delegated private matching protocol, oblivious access control policies (Chapter 5) and oblivious data search (Chapter 6) are realized. Both of the proposed schemes utilize cloud infrastructure for computational and persistence capabilities. In the following, we present key contributions of these systems with respect to public cloud storage services, which enables data owner to share data with authorized subscribers.

1.4.1 Oblivious Access Control Policies

Oblivious access control enforcement governs data access within public cloud storage services. With oblivious access control policies, we make the following contributions in the area of cloud-based data sharing services:

- oblivious evaluation of access control policies by a cloud service provider
- realization of access control policy framework without relying on trusted third party
- processing of encrypted access control policies and access attributes within the domain of cloud service provider
- ensure privacy of outsourced data and personal information of a subscriber during access control policy evaluation

Figure 1.7 illustrates the enforcement of access control policy in public cloud storage services. It utilizes public cloud infrastructure to obliviously evaluate access control policy without relying on any trusted third party. Encrypted access control policy and access parameters ensure that malicious subscribers and untrusted CSP cannot deduce any information about the outsourced data and data owner.

The highlights of proposed oblivious access control policy evaluation framework is its amicable computational load on cloud infrastructure. We implemented the proposed framework in Java and deployed on Google App Engine. The execution cost of access control policy evaluation remained between 0.01 ~ 0.30 dollars per 1000 requests, for access control policies consisted of 2 to 10 distinct access parameters.

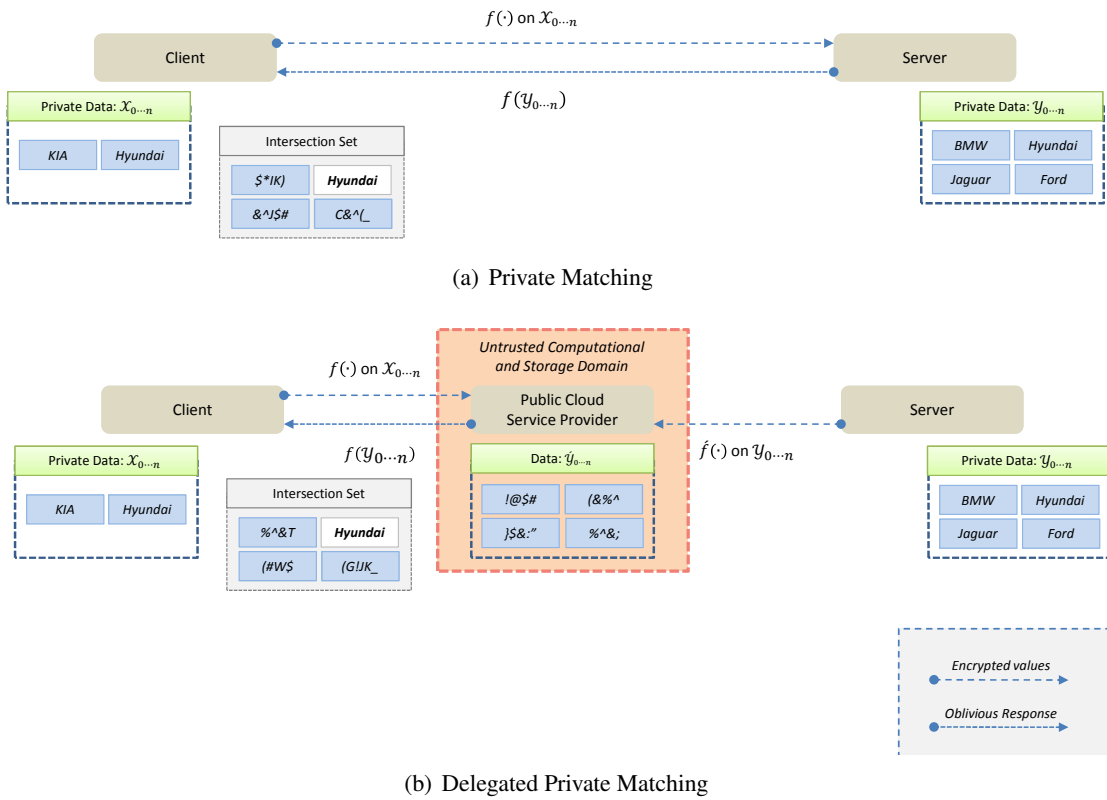


Figure 1.6: Extending private matching to delegated private matching - abstract view

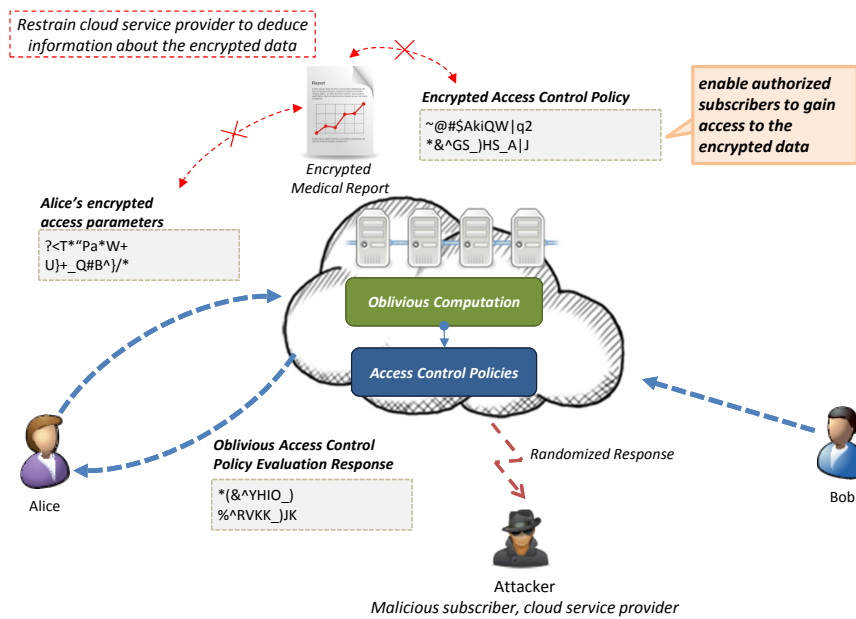


Figure 1.7: Oblivious access control policy evaluation

1.4.2 Oblivious Data Search

Oblivious data search permits subscribers to search public cloud storage while ensuring privacy of the outsourced data. The proposed methodology to search public cloud storage makes the following contributions in the area of cloud storage services having shared data contents.

- oblivious execution of search queries on a public cloud infrastructure
- conformance of search queries with access control policies
- enhanced capabilities of subscribers of search cloud storage without the need of trapdoors - index-based data search
- single persistence of index data structure for seamless data and index updates

Figure 1.8 shows the proposed methodology to search encrypted outsourced data. It realizes index-based data search over encrypted outsourced data. Authorized subscribers can search cloud storage without the need to exchange trapdoors with the data owner. Encrypted search queries are evaluated by untrusted CSP. Oblivious evaluation of search queries ensures that CSP cannot learn any information that it can be exploited to compromise privacy of the outsourced data and data owner as well.

Efficacy of the proposed oblivious data search was tested on Google App Engine. Our evaluation results showed that computational cost of search query execution remained within the range of $0.035 \sim 1.098$ dollar per 1000 requests, for queries comprising of 2 to 14 distinct search criteria.

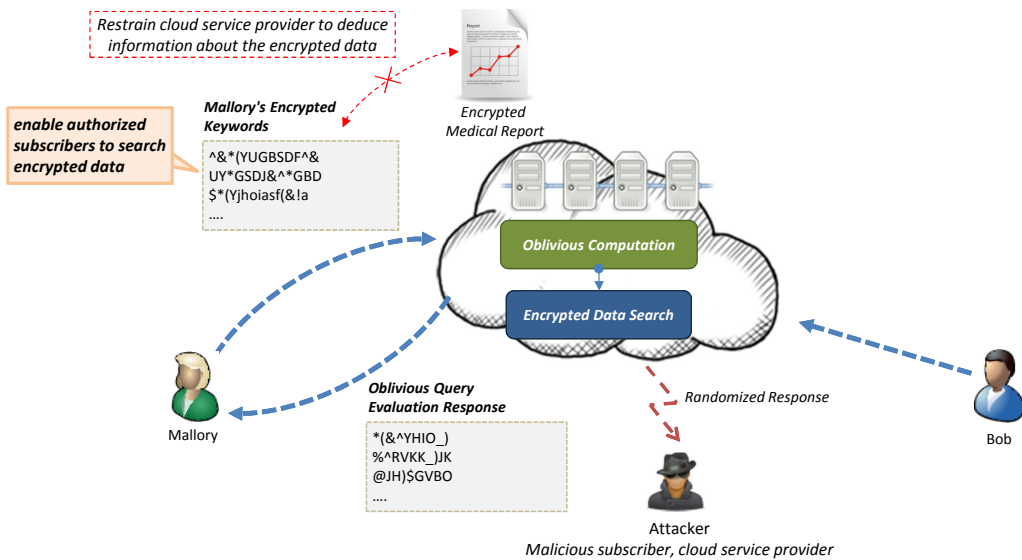


Figure 1.8: Oblivious data search on encrypted outsourced data

1.5 Structure of Dissertation

This dissertation is organized in to following chapters.

- **Chapter 1 - Introduction:** Chapter 1 explains the privacy issues of public cloud storage services. It illustrates that existing methodologies lack realism in an environment where both CSP and subscribers can behave maliciously. Potential privacy breaches are presented as service scenarios - illustrating the fact, the conventional procedures to ensure privacy do not provide withstand attacks launched by malicious subscribers or even by untrusted CSP.
- **Chapter 2 - Related Work:** Chapter 2 discusses the related work encompassing the area of privacy-aware data sharing and searching in untrusted storage services. Existing systems that enforce access control policies in public cloud storage services are discussed at length. In addition, methodologies to search encrypted data are also presented along with their limitations to maximize utilization of public cloud infrastructure.
- **Chapter 3 - Preliminaries:** Chapter 3 presents preliminaries that are used in this dissertation to ensure privacy in public cloud storage services. Techniques and protocols discussed in this chapter provide the building blocks for our proposed privacy-aware access control policies and data searching methodologies.
- **Chapter 4 - Delegated Private Matching:** Chapter 4 presents the extension of private matching protocol called delegated private matching protocol. In this chapter we argue the existing private matching protocol cannot be adopted by untrusted domain, we then illustrate its modified version that conforms to public cloud configurations. Security analysis of delegated private matching is also presented in this chapter.
- **Chapter 5 - Oblivious Access Control Policy Evaluation:** Chapter 5 explains the proposed access control policy evaluation framework for public cloud storage services. Implementation details and its deployment model for public cloud storage services are also presented. Evaluation results are discussed at length illustrating its efficacy to govern data access in untrusted domain.

- **Chapter 6 - Oblivious Data Search in Cloud Storage:** Chapter 6 presents privacy-aware data searching methodology for public cloud storage services. It presents functional details of provisioning oblivious data searching capabilities to authorized subscribers.
- **Chapter 7 - Conclusion and future directions:** Chapter 7 concludes the dissertation along with the future directions.

In this chapter, we specify the scope and boundaries of research carried out in this dissertation. We mainly focus on the problems of access control enforcement and privacy-aware data searching within untrusted domain. This chapter presents an overview of related work. We begin with the definition of privacy in different domains. Then we present information privacy in untrusted domain, and examine some of the existing prominent attempts to prevent disclosure of personal or confidential information unwilling. After that, we elaborate information privacy issues in untrusted storage system.

2.1 Privacy

The concept of privacy is not new, different philosophers have defined it differently with the evolution of human society - although the word *privacy* was never used specifically, but its concept existed long before. Nevertheless, the overall notion of privacy remains the same.

Aristotle's distinction between the public sphere of politics and political activity, the polis, and the private or domestic sphere of the family, the oikos, as two distinct spheres of life, is a classic reference to a private domain [58].

Most recently Avner Levin et. al. describe privacy as:

Control over personal information [59].

Predominately the notion of privacy is based on control and autonomy [60]. Arthur R. Miller describes privacy as:

An individual's ability to control the circulation of information relating to himself [61].

Professor Alan Westin at Columbia University describe privacy as:

The claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others [61].

Margaret Mead along with several other anthropologists have illustrate the concept of privacy protection in various cultures as:

through concealment, seclusion or by restricting access to secret ceremonies.

Thus, the concept of privacy centered on the willingness to reveal personal information and ability to control its access. It has been around ever since we started thinking about segregation of information with respect to one's inclination or need to share it with others.

2.2 Data Privacy

The same notion of privacy discussed earlier applies to the current era of digitization, enabling us to share information instantly and conveniently. At the same time, it has made it even difficult to govern diverse source of information across different domains. We have come a long way in evaluation of digital age, from desktop computer to web-server, from hand-held telephony devices to sensory enable smartphones. With each technological advancement we made, the need of privacy protection also retrofitted, due to the possibilities and capabilities to an adversary to compromise privacy of an individual and information associated with it.

Data privacy within the context of current information age can be defined as [62]:

Individuals claim that data about themselves should not be automatically available to other individuals and organizations, and that, even where data is possessed by another party, the individual must be able to exercise a substantial degree of control over that data and its use. This is sometimes referred to as "data privacy" and "information privacy" .

Information Privacy is the interest an individual has in controlling, or at least significantly influencing, the handling of data about themselves.

Each domain has its own set of standards to persist and provision data. However, the notion privacy across diverse domains remain same - adversary should not be able to deduce any information related to data owner and data should be accessible to authorized users. Considering the scenario discussed in Section 1.3, healthcare record of a patient should only be accessible to a doctor, patient is visiting. The notion of data privacy is not limited to data accessibility. It also restrain adversary to deduce any confidential and personal information from the data itself. Thus, the scenario illustrated Section 1.3, restrict an unauthorized subscriber to search encrypted data and deduce information that can lead to potential loss of personal and data privacy.

In the current information age, data privacy is all about one's capability to govern data access. It defines capability of a user to allow or deny access to a particular user, whilst ensuring that only authorized users can access data. In addition, it prevents unauthorized user to use deductive reasoning to compromise privacy of a user. From a data owner point of view, control and ownership of data and governance of data accessibility are key capabilities that derive the notion of data privacy.

2.3 Cloud Storage Service and Data Privacy

In the following, we present most recent and relevant work within the area of authorized data access and encrypted data search for cloud storage services. As emphasized in previous chapter both of these capabilities greatly affects the utility of a cloud storage service from privacy-aware data sharing point of view.

2.3.1 Authorized Data Access in Cloud Storage Services

Access control policy ensures that every access to a system can be governed and only authorized users are able to gain access to resources (i.e., data, memory, network etc.) [63]. In cloud storage services, access control policy ensures authorized data access when data is shared with multiple subscribers. We examine efficacy of existing access control policy frameworks in the context of

cloud-based data sharing services.

Primarily within untrusted domain of CSP, access control policy can be realized by three different methodologies i.e., relying on trusted third party, using attribute based encryption and ensuring data owner's availability to evaluate access control policies. In the following, we present related work utilizing these methodologies.

2.3.1.1 Access Control Enforcement by Trusted Third Party

One of the most prevalent methodologies to enforce access control policy within untrusted domain of CSP is by relying on trusted third party. Encrypted data is outsourced to cloud storage service; whereas, data decryption is persisted on trusted third party along with the access control policy. To obtain data decryption key subscriber's access attributes must conform to access control policy. Once subscriber has the data decryption key it can access encrypted data for which it has obtain decryption key. Figure 2.1 illustrates the conceptual model of governing data access with trusted third party. Enforcement of access control policy with key management services shares the same analogy - methodologies focusing on such services are discussed in the following.

FADE [64] is a secure overlay cloud storage system based on policy-based assured file deletion. It is designed to share outsourced data in an untrusted domain and to assuredly delete it once the need of sharing is over. To ensure data confidentiality and authorized data access, data encryption key is used to conceal the outsourced data, and control keys are used to encrypt the data encryption key. Concealed outsourced data and data encryption key are outsourced to a cloud storage; whereas, control keys are managed by a key manager. Key manager is also responsible to maintain and evaluate access control policies. Whenever a policy is revoked appropriate control key is deleted, restraining access to the outsourced data. FADE delegates the task of policy evaluation to a key manager.

TrustStore [65] is an Amazon S3 based storage service, enabling storage subscribers to outsource their confidential data to Storage Service Provider (SSP), with data confidentiality and integrity considerations. It utilizes a Key Management Service Provider (KMSP) to generate and distribute decryption keys, besides this KMSP also takes over the responsibility of subscriber authentication. TrustStore segregates data into two components; data-fragments: dividing data into

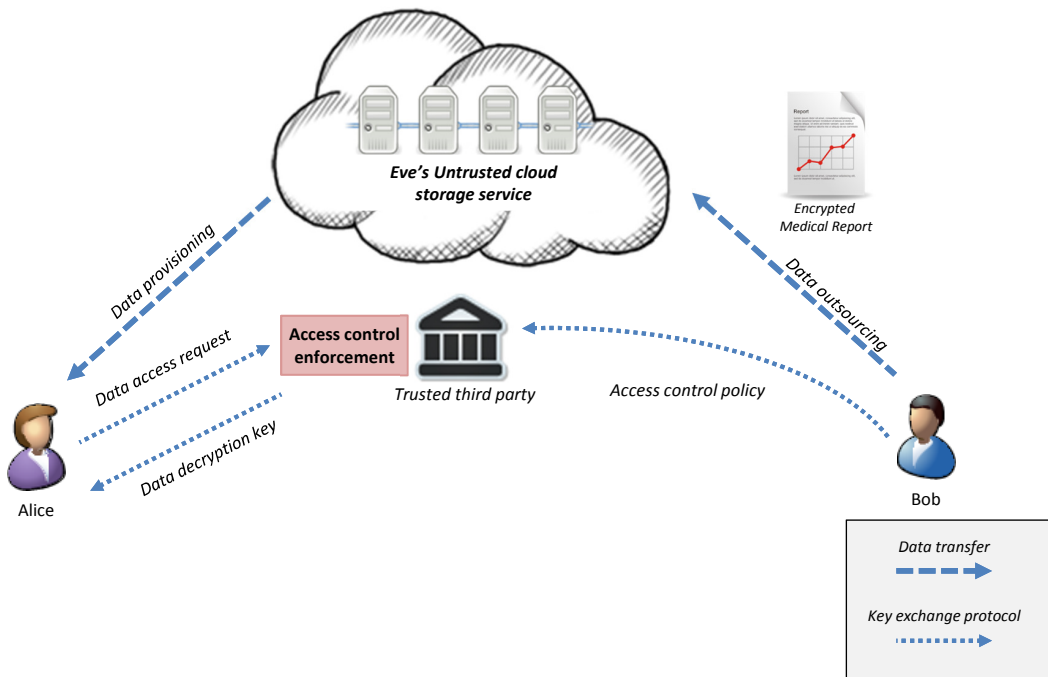


Figure 2.1: Access control enforcement by relying on trusted third party or services

equal sized fragments encrypted with distinct encryption keys, and meta-data-object: containing information about encryption keys and data fragments. Data-fragments are outsourced to storage provider and meta-data-object is stored at KMSP. TrustStore is based on an assumption that SSP and KMSP do have any knowledge about each other, thus privacy of data is ensured.

CRUST [66] is a cryptographic remote storage system, which avoids the use of public key encryption for the purpose of speed and efficiency in cryptographic operations. It assumes the availability of a trusted agent (*i.e.*, *trusted third party*) responsible for key management. For each new authorized subscriber, trusted agent generates an encryption key by using system master key stored locally on it. CRUST maintains file in blocks each encrypted with a separate key. Apart from that, it utilizes lazy re-encryption strategy which ensure that only the updated block of a file is re-encrypted to avoid unnecessary cryptographic operations.

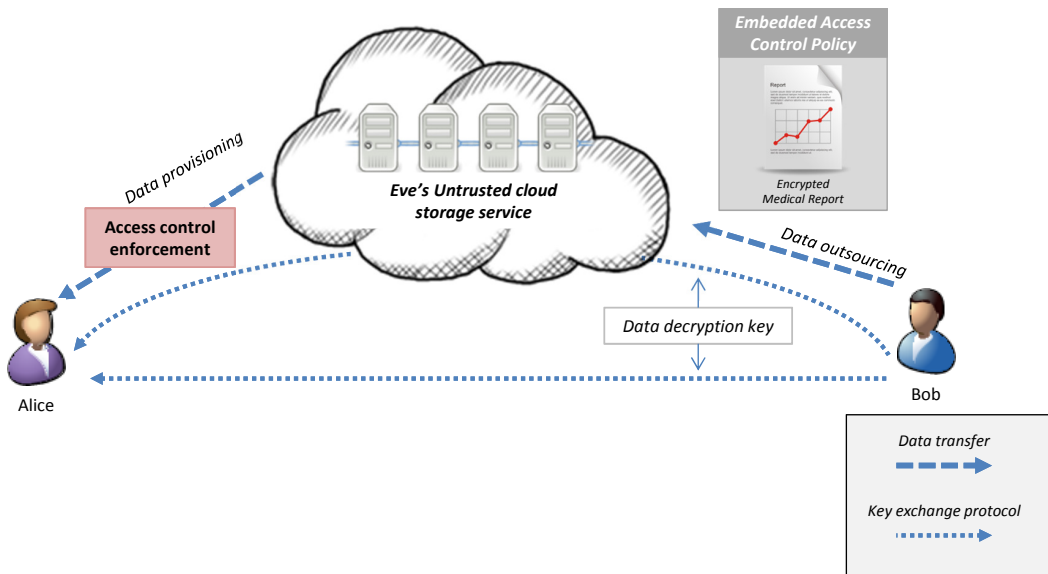


Figure 2.2: Access control enforcement with attribute based encryption

2.3.1.2 Access Control Enforcement with Attribute based Encryption

Attribute based encryption [67] is a relatively new form of access control enforcement. It associates data decryption policy with the encrypted data. Data decryption key (user's secret) is equipped with set of attributes that confirm of decryption policy. Thus, subscriber having required set of attributes associated with its data decryption key can decrypt the data. Data owner generates the data decryption key and distributes it to authorized subscribers. Decryption key can be distributed either by direct interaction or by encrypting it with authorized subscriber's public key and outsourcing it to an untrusted key management service. Figure 2.2 illustrates the abstract model of cloud-based data sharing in which outsourced data is encrypted with attribute based encryption.

Cryptographic Cloud Storage [68] is a cloud-based data sharing system designed to outsource enterprise data storage. It consists of three core components i.e., Data Processor (DP), Data Verifier (DV), and Credential Generator (CG). DP encrypts the outsourced data, DV verifies the data integrity at cloud storage, and CG generates decryption key for the users with whom data owner wants to share the outsourced data. Decryption keys are generated according to access control policy and user access privileges. Cryptographic Cloud Storage achieves fine-grained access control by encrypting the decryption keys with Attributes Based Encryption (ABE) [67].

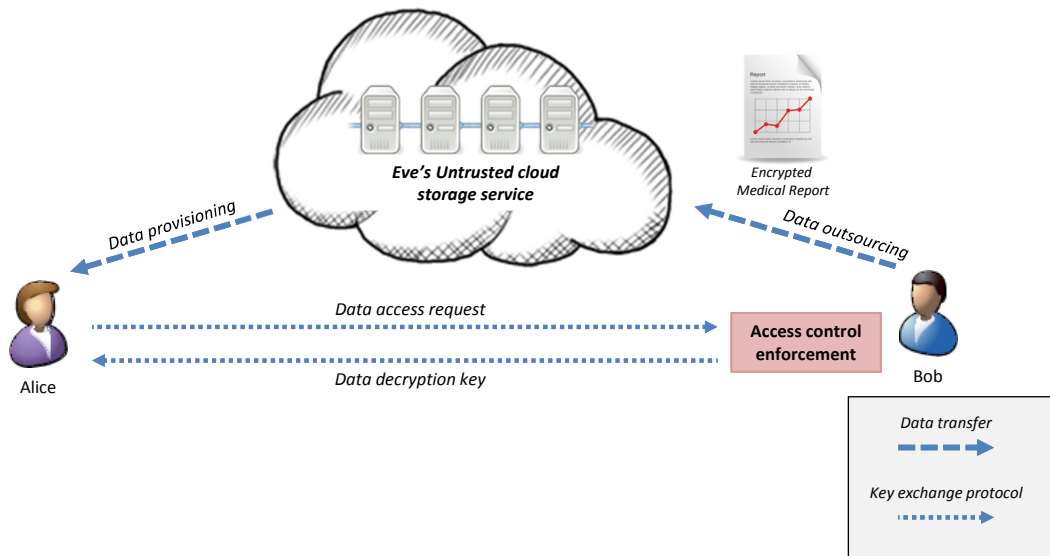


Figure 2.3: Access control enforcement by relying on data owner's availability

The most recent work addressing privacy issues in cloud storage is carried out by Shucheng Yu et al. [69]. They proposed a cloud storage system based on key-policy attribute-based encryption (KP-ABE) which utilizes lazy re-encryption along with proxy re-encryption for seamless data sharing. Access privileges are specified in user-secret key; whereas, outsourced data is encrypted with access attributes. In order to gain access to outsourced data subscriber's access privileges must be conform to attributes with which data is encrypted. Storage provider distributes the proxy re-encrypted secret key to authorized subscribers.

2.3.1.3 Access Control Enforcement by Data Owner

The simplest form of access control enforcement is to rely on data owner to distribute data decryption key to authorized subscribers. Data owner encrypts the outsourced data with appropriate encryption algorithm. It then outsources the encrypted data to a cloud storage service. Subscriber seeking access to encrypted data engages in a key exchange protocol with the data owner. Data owner evaluates the access control policy and provides valid data decryption key or access token delineating that subscriber is authorized to access/decrypt the outsourced data. Figure 2.3 shows a cloud-based data sharing system, in which data owner evaluates the access control policy.

Weichao Wang et al., [70] proposed a cloud-based data sharing system for massively large data. In order to achieve fine-grained access control, data is divided into multiple data blocks (D_1, D_2, \dots, D_n) , each encrypted with a distinctive block encryption key. These keys are managed by the data owner in a binary tree with the possibility to derive valid block encryption key from the parent node (non-leaf node). Data owner himself issues the security token along with a secret value to authorized subscribers. Security token is utilized by the storage provider to ensure that request is initiated by an authorized subscriber. Whereas, secret value is utilized by a subscriber to derive block decryption key. Access privileges of a subscriber are evaluated by the data owner himself, after which appropriate secret value is revealed.

Remote untrusted storage system closely resembles to cloud storage systems specifically in terms of storage service provisioning. There are numerous systems which tends to ensure data privacy in remote storage systems.

Plutus [71] is a cryptographic file storage system that enables secure file sharing on untrusted servers. Plutus greatly reduces the number of encryption keys that need to be exchanged with legitimate users, by aggregating files of similar access patterns into file-groups. Individual files are encrypted with file-block keys, which are further encrypted with a file-lockbox key. Instead of disseminating the file-block keys, file-lockbox key is handed over to legitimate users, by the data owner.

SiRiUS [72] is a secure file system designed to layer over an insecure network and a P2P file system. It works by maintaining an access structure in a meta-data file (md). Each entry in md contains the file encryption key $F EK$ and signature key $F SK$, encrypted with user's public key. Legitimate users can decrypt the respective entry in md by using their private key.

2.3.2 Limitations of Conventional Access Control Enforcement within Cloud Storage Services

Although methodologies discussed earlier ensure authorized data accessibility within untrusted domain; however, those methodologies fails to maximize utilization of cloud resources. Additionally most of these methodologies do not consider access control policy as confidential information - thus leveraging malicious CSP to deduce information that can lead to potential loss of data and

personal privacy. Table 2.1 lists the limitations of most relevant cloud storage services which enable data sharing.

Attribute-based encryption implicitly enforces access control policy. However, data encrypted under attributed based encrypted can reveal confidential information to the CSP. As illustrated in Section 1.3, access control policies can assist malicious CSP to deduce confidential information about the data and data owner as well. Figure 2.4 and 2.5 show the data encrypted with attribute based encryption and user's secret key respectively. It is evident that malicious CSP can easily learn access control policy under which data is encrypted. Similarly, if secret key of a subscriber is compromised adversary can learn access attributes and then can collude with CSP to compromise privacy of the outsourced data that conform to compromised attributes.

Although attribute based encryption ensures authorized data access, however it fails to conceal confidential information (i.e., access control policy and access attribute). Thus, for untrusted cloud storage services, attribute based encryption can leverage CSP to deduce confidential information even if it does not have access to valid data decryption key.

Table 2.1: Limitations of conventional access control methodologies.

Related work	Access control enforcement	Limitations
Cloud-based data sharing system for massively large data [70]. Large data files are divided into multiple parts - each encrypted with different key.	Keys are managed by data owner in a binary tree structure. Security tokens are issued by data owner and validated by cloud storage provider.	<ul style="list-style-type: none"> • Availability the data owner • Reliance on untrusted cloud service provider
FADE [64] is a secure cloud storage system. It is designed to share outsourced data in an untrusted domain and to assuredly delete it once the need of sharing is over.	Data encryption key encrypts the outsourced data. Control keys encrypts the data encryption key. Control keys are managed by a key manager.	<ul style="list-style-type: none"> • Delegation of data governance to key manager • Poor utilization of cloud resources
TrustStore [65] is an Amazon S3 based storage service. It manages data as data-fragments and meta-data. Data-fragments are persisted at Storage Service Provider (SSP), whereas meta-object is managed by Key Management Service Provider (KMSP).	Utilizes a KMSP to generate and distribute decryption keys. KMSP and SSP are independent entities and do not know each other.	<ul style="list-style-type: none"> • Delegation of data governance to key manager • Impracticable assumption
Cryptographic Cloud Storage to outsource enterprise data [68]. Data Processor encrypts the outsourced data. Data Verifier verifies the data integrity at cloud storage. Credential Generator generates and manages credential of the users.	Utilizes Attribute Based Encryption (ABE). Data owner generates and disseminates ABE secret key to authorized users.	<ul style="list-style-type: none"> • Availability of data owner • ABE reveals information about access control policy
SiRiUS [72], Plutus [71], and CRUST [66] are remote storage systems.	Utilizes asymmetric encryption to ensure authorized data access to the outsourced data.	<ul style="list-style-type: none"> • Poor utilization of cloud resources

yîk<84>iĒD.©Z<91>k^<9d><8e>o(è^A^RzB^Ge&rôDéé:Ë^L<8d><8e>. ^CaĂ^UX<9c><90>B> <8e>yHŭ
ëeqCøY^C^A-<93><83>^AV_bĚ<9b>9Ç^ODSfXtăy`l^Kæ^?oşİ,(Ø4<8e>_ĖS^A^ŷ^S
g~^|Ů2c[±ù<ê@ê@D^ê@ê@D^ê@ê@ê@ê@ê@ê@Initechê@ê@ê@<80><90><89>y^M^H<8a>G^<85
>3^N<92>^VjQ2ôœN^X^SŦ<80>CòĖu3iā&^9e>^?
xE`_k\<92>M<92>©.*@n<8a>^G^Wó<8f><9d>^R<9e><95>y^W#^A^?^?#Éç>^XQp@^CĬ.Ĕđ@_Ĕ<95>^M^?
@_,^[^D#^Yŷ;Ă^F4<9d>@ex
%Pk^UD^2@ē^L<@>Ÿl[uô&^PD<80>niĒhıppx^@ê@ê@<80>^Vü^êvx<87>_<9f>^Srú»ÉéEB<9a>,d<91>^A<
84><91>Ă@^Z^V4R^ô_f^V© ðYĂ^úŷİ^<95><99>Pw^E^?
u@^WionK%Ă<92>'ûrZR<8c>Ô^F<89>Ø^V<9e>^LYİ^<8c>,<81>ěz>Iā4^Lōi<9b>Ĕ^n^÷^^Eó<98>^LS<
9e>p^T|
±H<9a>Tô-ñ^H<9d><84>|ö+py) ^P[V^Zi<96>Ø^@ê@ê@ê@ê@ê@ê@Lease_Soft@ê@ê@ê@<80>@ñ=4B^B
<90>^AGô&<84>^E<93>_<94>Nè^PÀp^?^?^VcG^@|
^<8f>Øn<8e>ëĖÜ<92>Ů3#^Wėh_i^S^?;yâEuĀðòàx^M^inò!
ÒŮ^Kŷ»Eabô&Âtc<97>^V[<9f>ð^EI<8e>ô&êGi^YDu5Ă<99><^&ũ
^f<9d>9B<8c>kfô<97><9d>puĒ&L^<97>L^MD:<89>Ĕ^Wô&NİĒê@ê@ê@<80>#Aw^|Iy<8e><8b>Ů^-9±µUñ
^X<8f>Ă<8a><9c>ôñ|Çi<8c>^Z=Źi^n<8c>^RY<84><85>J^X/<93>J3<97>°^ [ăbh<9b>£Fôix
°^_<84>U3p|lrĂ^X<89>^[Ĕ<97>Ba'' t;+z <84>^C^Vê4<8f>
%<86>^FONŮ<86>P<9a>^Uß2;P=Ĕ^F8W<82>>^ToD.4^_Uôē^Λ^3<8c>U^T^!
#ê@ê@ê@A^@ê@ê@Software_Developmentê@ê@ê@<80>^p^|E0ē9ə(wŮôĖŮJt<89>Uc^?
Ă^Pp'ůáil7Ā\<93>^ŷv<9b><82>O^]k<8f>B^F&ẽ%;[,<80>Ōi^+â&Ă^qŷ^FM_H0!f2|
|ÖĐó| ^HbĂ2<82>^X<84>ă±.ĂVx4~2C##k;KEĂ|^ZôK /<9c>^_j^VĔb
<86><83>>0<87>^PŞ<9a>!|ă<88>.^Ñê@ê@ê@<80>^A^BOĂôz^,Ů<9d>ũ^|r<98>^\\n+<8f>==w<93>426çŮ
°<86>3^u0^H^?:^VAD<9b>^<97>1^@W^c91>w6^Zôñg[Ăy^GG^Prô&ĂŮ0ĖZY|ăTs2!°<8f>Êmr<8f> ŮZ
ăiyñ<80>1ĂĖ°<95>ô&B=Àn<90>^Heŧũ;^CăxŷŮ ^K 3|Ů<8a>,%ŲZ
i^@ê@ê@A^@ê@ê@Team_Leader^@ê@ê@<80>T^<8e>@<99>^Bq|ũ^a^2^uù<93>Ă<8c>^2t)^E<85>9ăŎ<
89>3<8e>^L@<90>^FEN^E;ø^P|
<9a>[%<82>^GITf|<9f>^4_k<8a>Rkn3R#^E^EAz<91>\$<9c>7z^KS|Ĭd<93><81>ø<80>úi^3 W3<ø
|^G I<93><83>^N^?4&82>ăñ|^Cê^PôTç|RV^Xpn<91><96>;5?Áú=WÓ<9a>@»(^ê@ê@ê@<80>n

Figure 2.4: Data encrypted with attribute based encryption

^VÍ<Z>8>A^]ýBæ\[^[ôîRH^\'I\WóTçÜ<8b>ø^C<82>^G^H4ýF^íKK^<8a><80>\^é\ [<90>^S<82><95>
<89>~L/E_ÛÄ/5ø^_5f^R~^Ú<8e>èüà<96>ð#P+O^°<99>»è^ùî.TðI,±Z;_úD!N<9f>Ä_<8b>-
È<8d>)ôqîÊOK<84>ðÊ6Ä~¥Sot^è^@^@I^nitech^@^@^@<80>9~Äî-
=+i4)P<8a><84>^ÈwëYÄîð%qqH^W^Ä<88>9^S^DStù7Ê<90>^>[<8c><95>e<97>'.9Eî^N<99>uÄô\$U!
ððäRð%>)èE(ágù<^ArôÍf<97>^G;Ñsu:Jø^Ts_æ
Un^S=^4ý6+^[\^mW\|<9c>0ÇÄE<9a>hoøpHYÇ÷æ<9a>^'ð<ø<^@^@<80>9Ñî^iá7ú1<85>^A<8b>^Z<94>OQ»
_û]JL^<89>»xi^A^HU^[\afó.Û-d^E^î]I<94>B6^_U<8d>ç^Z^ZDÜF<96>^>|<97>Ñ
ä_k_y^LjÄW^<94>^çYëÄ_+8<96>I^tð:qÄ<80>|Ê<8b><9d>d^D^Oð^W7^DÈÄ<91><81><9f>^X<84><8a
>üä<8d><9d>°<9d>h<^<8f><9b>^Bðç^WrG^<^Lease_Soft^@^@^@<80>+ä|<88>^B)i<84>
Êw<87>8^ð;^Íq<85>^A<88>ðP^Lê^ð,t^\'^Aðö^;V<82>P-H-
M^D^sh^<83><8A<89>;nRîBðîZy_<8a>^C<81>ÛS<82>^*Nqu(<^@;Ê^Äö<q\$9^{|<93>^Äö^'S<87>,MùK^
;cu<8c>P^YôMÄR\$È^vffîôYGêðö(H-ð<80>^E^S<8b><9d>D^D^Oð^I^G<95>à<97>u_é<91>
EßJ^N<84><8e>;ð^>|<84>üÄeð|è.w_o~O>P_ä%~ý|(<9e>^}|^Ä<91><8f>|ÜK^FÜC'^B1È<81>É{^ÖB%£
_ùuY<83><85><8f>^A?~
j<80>U<8f>Xð<8a><9f>-
â^T;{^+uü<^<8b>ðÄ<85>Ä|
iá^Software_Development^@^@^@<80>E<9f>N:~^<8a><ðq<8d>^_8D<9b><92>*].^NG<9d>×9^T<
8e>g1\$^Xðð1^,e<9a><99>?_.÷iú<93>^L|T~áo<87><8a>^\'Lýq<9a>8^0;9^\'B%?VÜÇÆ&!
f0Q^[6úî^[\Y^ÑjçbÈ^AúE^\'<98>/C÷+|s|mÄ|3^_ë|^2g_<96>îçá0ðz^FD8»^EÈ^@^@<80>S^Uðx^_
^Rðð<8b>^>|dçîEüðîèV/6<95>^6^A^L<93>_ëS<8b>îçhîð<è+^+÷8<85>âçÄH<80>ù<91>ñ~^P2^Pä^D
S
^K1r<8c>ðÊä<9f>^Úá^^[^U|.ç/CÜq<99>^ð:~L^?WäY<8f><86>îÄ^8a>Ä
^S<8c>~>|<87>,fð<9c><88>^'ð[ð%ôäèÜÄU<88>1äçTeam_Leader^@^@^@<80>p<85>0[ÄÜ
^t:~>91>0[~Le<96>]gè^d^'<8e>^@Z%@@^Nî#p^U^B4^*|
<9e><89>àç^K;^A^RäðäèÑ<99>^2XËYÄæä<8d><è~^O^<97>i)"<85>,"CIÎßu^#Ä)æp^@&;<90><8a>^'eO^~
Y<8a>ðÄ<99>ð%<8d>8^\'ð-È<97>è3<82>î^C3Ü<85>^üY_<86>-
Sç^D<92>i;,<9b>V^è<^@<80>^R<95>QR<85>~ä<8d>=<85>^@çV^B<9b>tá^|)î7<96>Ñ#!
äBMPZ<93><MöHîykp^H^Q_îæð<9c><96>^R<9f>Ü<84>uud<99><8c>R^k<84>P<9d>^Hü|wî«

Figure 2.5: Attribute based encryption secret key

2.3.3 Privacy-aware Search in Cloud Storage Services

Data search enables subscribers to access relevant data to avoid unnecessary access requests and bandwidth consumption. Searching capabilities in cloud storage services are of paramount importance as CSPs bill their subscribers on the amount of data accessed and network usage. Considering the lack of trust on CSPs, data searching capabilities can be exploited which lead to privacy infringement.

Primarily, encrypted data within cloud storage services can be searched either by using trapdoor-based search queries or by utilizing index data persisted as secure location. In the following, we present existing methodologies encompassing trapdoor-based and index-based data search within cloud and untrusted storage services. Their limitations within the context of effective utilization of cloud infrastructure are also presented.

2.3.3.1 Trapdoor-based Search Queries for Encrypted Data

One of the most prevalent privacy-aware searching methodology is trapdoor-based search over encrypted data. It enables to define one-way hash functions that are believed to difficult to compute without special input "called trapdoor". Entity that encrypts the data defines a trapdoor for a particular keyword, and leverages authorized users to search encrypted data by using defined trapdoor. Figure 2.6 illustrates the conceptual model of search over encrypted data (symmetric and asymmetric encryption) . Trapdoors can be transmitted to authorized subscribers either by direct communication or through trusted third party.

Searchable Symmetric Key Cryptography (SKC) was first proposed by Song et. al. [73], making it possible to search encrypted data, by using trapdoors defined for a particular keyword. Based on SKC various schemes have been developed which utilize it to search encrypted index, instead of the encrypted data [74], [75], [76]. Followed by SKC, first practical searchable Public Key Cryptography (PKC) was proposed by Boneh et. al. [77]. PKC enable untrusted server to perform search over encrypted data concealed with a public key, without the need to reveal actual decryption key (private key). Both, SKC and PKC utilize trapdoors to execute search queries.

Li et. al. in [78] proposed Authorized Private Keyword Search (APKS) on encrypted Personal Health Records (PHR) by using Hierarchical Predicate Encryption (HPE). In their construction

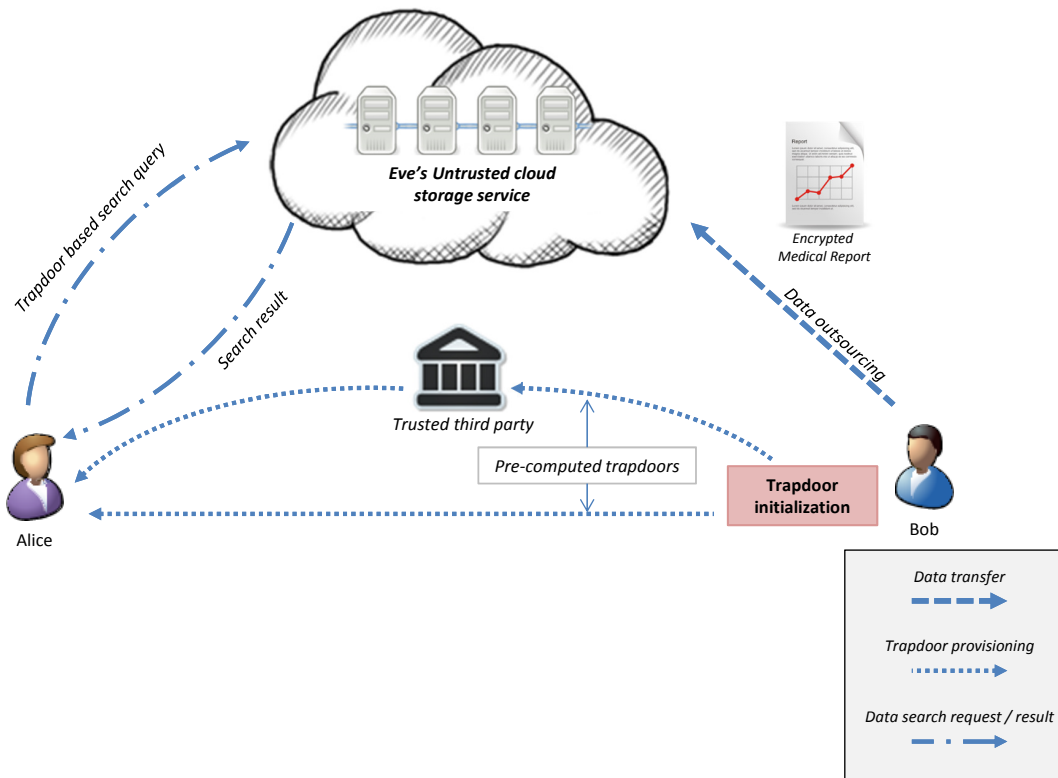


Figure 2.6: Trapdoor-based encrypted data search for cloud storage services.

of privacy-aware search, a Trusted Third Party (TTP) was responsible for distributing capabilities (trapdoors). Authorized subscribers obtain capabilities from a TTP, according to their access privileges and then submit trapdoors to a CSP. Likewise, Wang et. al. [79] proposed a secure ranked search over encrypted data, for data residing within the untrusted domain of a CSP. However, their proposed system only support single keywords based search queries. Similarly, CS2 [80] provides Symmetric Searchable Encryption (SSE) with Search Authentication (SA). CS2 utilizes inverted index to search encrypted data, along with dynamic data updates.

2.3.3.2 Index-based Search Queries for Confidential Data

Index-based data search in an altogether different way than trapdoor-based queries to search data within untrusted domain. Standard lookup queries can be executed on index data persisted in a secure location (i.e, trusted third party services or within federated domain of a data owner);

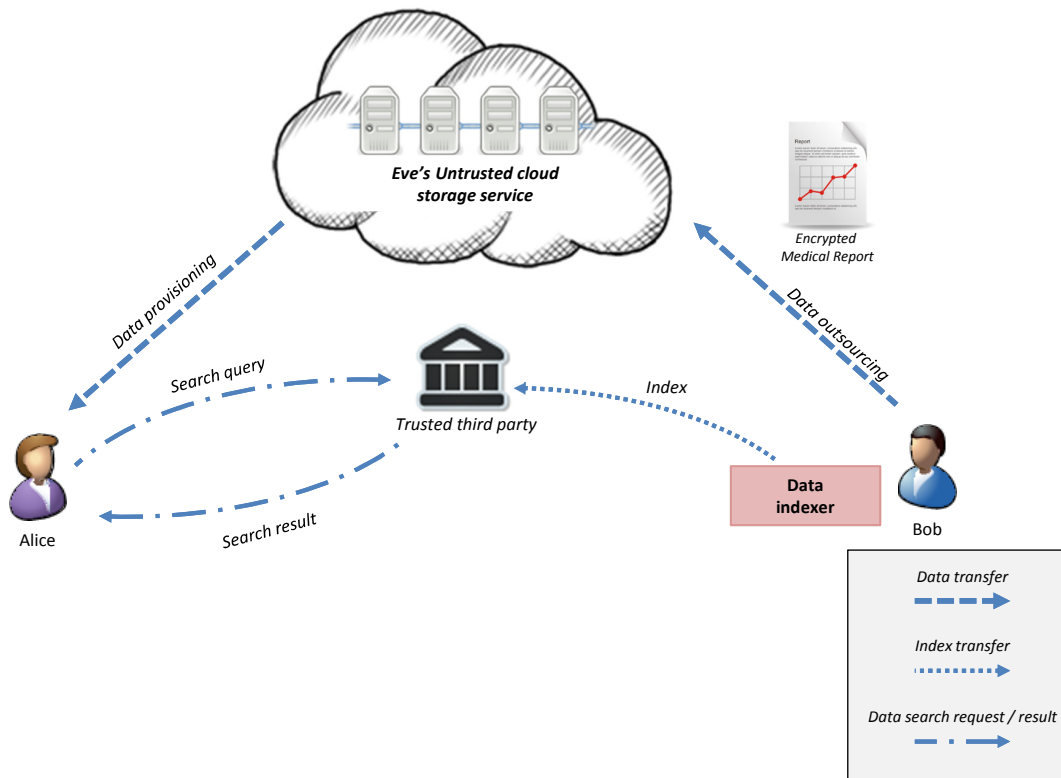


Figure 2.7: Index-based data search for cloud storage services.

whereas, encrypted data outsourced to an untrusted storage service e.g., cloud or remote storage service. Index-based data search provides more efficient and useful search results, as indexed data can be equipped with auxiliary information (i.e., inverted index, centralized index for related data) that can enhance searching capability of a user. Figure 2.7 shows the conceptual model of index-based data search for cloud-based storage services. In the following, we present searching methodologies, which utilize index-based data search to retrieve relevant document stored in a cloud or remote storage services.

Enterprise search products, like Google search appliance [81] and Windows enterprise search [82] provide document indexing functionality. Enterprises can use these products to query document repositories within their data center and over cloud storage as well. These products create a single enterprise wide centralized search-able index. Search queries are executed on indexed data and search results are filtered out according to access privileges of a user. As these systems enforce access control policies at query time, they require search services to be hosted within the

federated domain of an enterprise.

2.3.4 Limitations of Existing Methodologies to Search Encrypted Data Outsourced to Cloud Storage Services

Search over encrypted data enables privacy-aware searching capabilities within untrusted domain. However, it remains limited in its functionality, as searching capabilities (i.e., trapdoors) need to be transmitted to authorized users. In case of index-based data search, index need to persisted and processed at secure location as it can reveal data contents to an adversary. Thus, existing methodologies either provide limited searching capabilities or require additional computational resources.

Research concluded in [83] has shown that by carefully modeling search queries malicious users can learn valuable information from a centralized index, even if they do not have access to the data residing within federated domain. Thus, a malicious user can utilize index-based data search to submit unauthorized search queries to compromise privacy of the data persisted within trusted domain. Table 2.2 presents limitation of conventional methodologies to search encrypted data within the context of cloud-based storage services which provide data sharing facility to their subscribers.

Table 2.2: Limitations of conventional methodologies to search untrusted storage services.

Related work	Data search	Limitations
Searchable symmetric key cryptography (SKC) [73], Privacy-preserving queries on encrypted data [74].	Trapdoors based cryptography. Utilizes untrusted storage provider to execute search query.	<ul style="list-style-type: none"> • Limited searching capabilities - search queries are confined to trapdoors. • Availability of data owner.
Searchable public key cryptography (PKC) based on the concept of asymmetric encryption [77].	Trapdoors based cryptography. Utilizes untrusted storage provider to execute search query.	<ul style="list-style-type: none"> • Limited searching capabilities - search queries are confined to trapdoors. • Availability of data owner.
Authorized Private Keyword Search (APKS) on personal health record [78].	Trapdoor-based cryptography. Utilizes untrusted storage provider to execute search query. Trusted third party responsible for distributing trapdoors.	<ul style="list-style-type: none"> • Limited searching capabilities - search queries are confined to trapdoors. • Reliance on trusted third party for authorized data search.
Secure ranked search over encrypted data - Wang et al [79].	Trapdoor-based cryptography. Utilizes untrusted storage provider to execute search query. Search result are sorted according to frequency of a single trapdoor.	<ul style="list-style-type: none"> • Limited searching capabilities - search queries are confined to trapdoors. • Can only search for single keyword at a time cannot be utilized for complex queries.
Google search appliance [81], Windows enterprise search [82].	Searchable data index managed by trusted entity i.e., private cloud. Single enterprise wide centralized index.	<ul style="list-style-type: none"> • Poor utilization of cloud infrastructure.

This chapter presents a brief overview of cryptographic primitives and protocols that provide the building blocks for oblivious access control policy framework. These building blocks ensure that the evaluation of access control policy does not facilitate any malicious entity (i.e., access control policy evaluator, unauthorized user) to compromise privacy of the data. Access control policy is evaluated by using homomorphic encryption. Obliviousness of access control policy evaluation is achieved by employing private matching protocol.

3.1 Homomorphic Encryption

A cryptographic scheme is said to be homomorphic if it can be utilized to compute certain type of functions on ciphertext. The resultant ciphertext decrypts to a plaintext that is equivalent to the result of same function perform on the plaintext. An cryptographic scheme is said to be additively homomorphic if its encryption function \mathcal{E}_H holds the property i.e., $\mathcal{E}_H(x) * \mathcal{E}_H(y) = \mathcal{E}_H(x+y)$. An additively homomorphic cryptographic scheme is semantically secure if \mathcal{E}_H reveals no information about x and y , hence it is computationally infeasible to distinguish between the case $x \neq y$ and $x = y$ [84].

Public key encryption scheme proposed by Pascal Paillier [85] is additively homomorphic, and consists of subsequent fundamental algorithms.

3.1.1 Key Generation

Let p and q be two large primes and $n = p.q$. $\phi(n)$ denotes the Euler's totient function. Carmichael's function is represented by $\lambda(n)$. For n , the product of two primes, $\phi(n) = (p-1)(q-1)$ and $\lambda(n) = lcm(p-1, q-1)$. These two functions exhibits the following

properties over the multiplicative group $\mathbb{Z}_{n^2}^*$, i.e.,

$$|\mathbb{Z}_{n^2}^*| = \phi(n^2) = n \cdot \phi(n) \quad (3.1)$$

and for any $\omega \in \mathbb{Z}_{n^2}^*$

$$\omega^{\phi(n)} = 1 \pmod{n} \quad (3.2)$$

$$\omega^{n\phi(n)} = 1 \pmod{n^2} \quad (3.3)$$

Public key \mathcal{PK} is defined as (n, g) , where g is an element of $\mathbb{Z}_{n^2}^*$, and secret key \mathcal{SK} as $\lambda(n)$.

3.1.2 Encryption

To encrypt a message $m \in \mathbb{Z}_n$, randomly choose $y \in_R \mathbb{Z}_{n^2}^*$, and define an encryption function \mathcal{E}_H , such that

$$\mathcal{E}_H : \mathbb{Z}_n \times \mathbb{Z}_n^* \mapsto \mathbb{Z}_{n^2}^* \quad (3.4)$$

$$\mathcal{E}_H(m, y) = g^m y^n \pmod{n^2} \quad (3.5)$$

3.1.3 Decryption

To decrypt the ciphertext, L is defined as $(u - 1)/n$, $\forall u \in \{u | u = 1 \pmod{n}\}$. Ciphertext c can be decrypted by using secret key $\mathcal{SK} = \lambda(n)$, \mathcal{D}_H as

$$\mathcal{D}_H(c, \lambda(n)) = \frac{L(c^{\lambda(n)} \pmod{n^2})}{L(g^{\lambda(n)} \pmod{n^2})} \quad (3.6)$$

3.1.4 Homomorphic Operation

Arithmetic addition between the cipher texts, $c_1 = \mathcal{E}_H(m_1, y_1)$ and $c_2 = \mathcal{E}_H(m_2, y_2)$, is obviously computed as:

$$\begin{aligned} \mathcal{E}_H(m_1, y_1) &= g^{m_1} y_1^n \pmod{n^2} \\ \mathcal{E}_H(m_2, y_2) &= g^{m_2} y_2^n \pmod{n^2} \\ \hline \mathcal{E}_H(m_1, y_1) * \mathcal{E}_H(m_2, y_2) &= g^{m_1+m_2} (y_1 * y_2)^n \pmod{n^2} \\ &= \mathcal{E}_H(m_1 + m_2) \end{aligned} \quad (3.7)$$

3.2 Private Matching

Private matching [86] is a value matching protocol. It assists two interactive entities to compute set intersection over their private set of values, without revealing any element of their private set to each other. It uses homomorphic encryption to identify the commonalities amongst the private sets, whilst ensuring privacy of each set.

Suppose, there is a client \mathcal{C} and a server \mathcal{S} . \mathcal{C} has its own private set of values $\mathcal{X} : \{x_1, x_2 \dots x_n\}$, so does $\mathcal{S}, \mathcal{Y} : \{y_1, y_2 \dots y_n\}$. \mathcal{C} wants to compute set intersection with \mathcal{S} over the private set of values (i.e., \mathcal{X}, \mathcal{Y}). However, \mathcal{C} does not want to seep out any information about \mathcal{X} , with an exception of set cardinality. To identify the commonalities between \mathcal{X} and \mathcal{Y} , \mathcal{C} computes a polynomial (see equation 3.8), whose roots are members of \mathcal{X} .

$$P(x \in \mathcal{X}) = (x - x_1)(x - x_2) \dots (x - x_n) = \sum_{i=0}^n \alpha_i x^i \quad (3.8)$$

\mathcal{C} then sends the homomorphically encrypted coefficients $(\hat{\alpha}_{0\dots n})$ of $P(x)$ to \mathcal{S} . By using $\hat{\alpha}$, \mathcal{S} evaluates $P(y)$ for every element of its private set. It then computes oblivious value by multiplying evaluated $P(y)$ with a random number r and adding it to y , i.e., $\mathcal{E}_H(r.P(y) + y)$, where \mathcal{E}_H is a homomorphic encryption algorithm. These oblivious values are then send to \mathcal{C} for decryption. At \mathcal{C} , the decryption of an oblivious value results in y , if $P(y)$ computed by \mathcal{S} is evaluated at z , such that $\langle z \subseteq \bigcap \mid (z \in \mathcal{X}) \wedge (z \in \mathcal{Y}) \rangle$. Otherwise, \mathcal{C} ends up generating a random value. At the end of this protocol, \mathcal{C} learns only the intersection set; whereas, \mathcal{S} ascertains nothing more than the cardinality of \mathcal{X} . Figure 3.1 illustrates the interaction between the client and server.

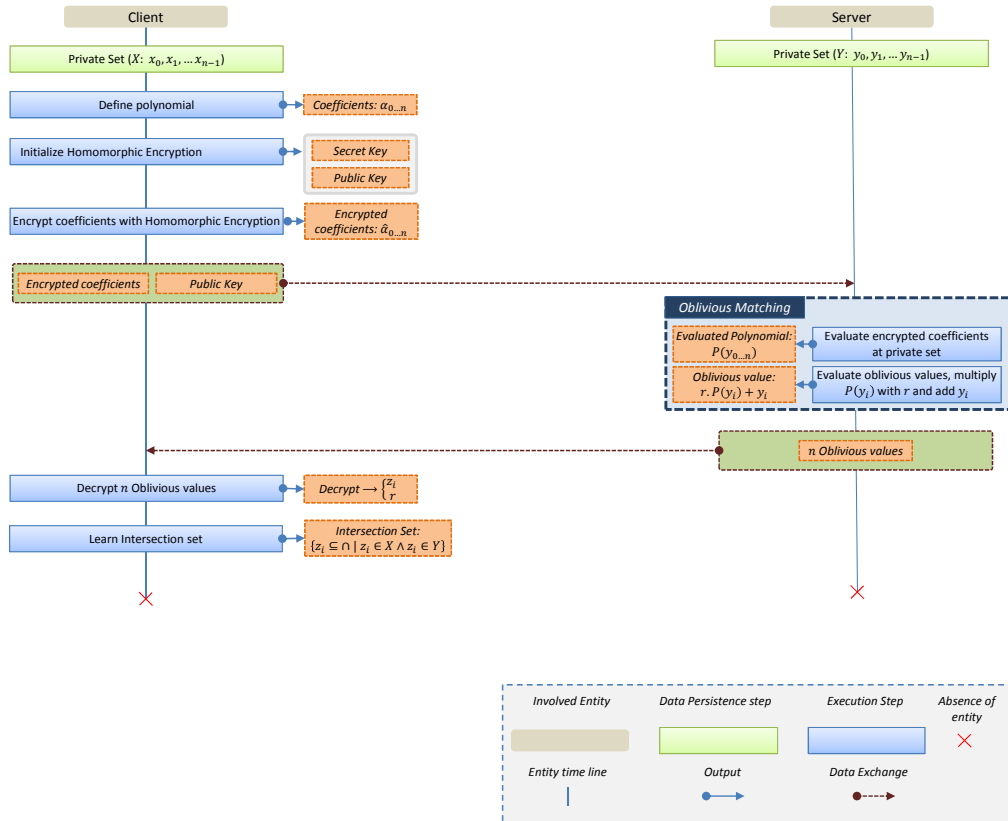


Figure 3.1: Private matching: entity interaction and time-line.

3.3 Proxy Re-Encryption (PRE)

Proxy Re-Encrypt (PRE) is a cryptographic primitive, which transforms the ciphertext from one secret key to another without revealing the secret key to a semi-trusted party [87]. Through PRE, ciphertext encrypted with Alice secret key can be transformed to another ciphertext, which Bob can decrypt without revealing any information to the intermediary (*semi-trusted server*). PRE consists of four fundamental algorithms: Key Generation, Encryption, Re-Encryption, and Decryption. Suppose Alice wants to send a message m to Bob through an intermediary server by using PRE, following are the steps, that will be executed.

3.3.1 Key Generation

Alice first selects a Bilinear Group \mathbb{G} of prime order q with g generator. Two random numbers a and b of order q are generated. a and b are then used to generate respective secret keys $SK_a = a$ and $SK_b = b$. Consequently public keys are produced as $PK_a = g^a$ and $PK_b = g^b$. Once public keys are defined Alice selects a random number $r \in \mathbb{Z}_p^*$, along with a Bilinear Map of \mathbb{G} as $Z = e(g, g)$. Finally, proxy-key is generated as $RK_{a \rightarrow b} = (g^b)^{1/a}$ and is handed over to a semi-trusted server responsible for ciphertext transformation.

3.3.2 Encryption

In order to encrypt message m , with Alice public key, ciphertext is computed as $C_a = (Z^r.m, g^{ra})$.

3.3.3 Re-Encryption

This step is executed by a semi-trusted server. Ciphertext is transformed from $C_a \rightarrow C_b$ by using proxy-key $RK_{a \rightarrow b}$

$$\begin{aligned}
 C_b &= (Z^r.m, e(g^{ra}, RK_{a \rightarrow b})) \\
 &= (Z^r.m, e(g^{ra}, g^{b/a})) \\
 &= (Z^r.m, Z^{rb})
 \end{aligned} \tag{3.9}$$

3.3.4 Decryption

To decrypt the ciphertext C_b , Bob uses his secret key SK_b , communicated to him by Alice through secure means i.e., *SSL*. Message m can be obtained as $m = \frac{Z^r \cdot m}{(Z^{rb})^{1/b}}$

This chapter presents extension of private matching protocol called Delegated Private Matching (DPM). Private matching is an interactive protocol between two entities, enabling them to compute set intersection over their private set of values. However, both entities must actively participate in the protocol by assuring their availability. DPM relaxes the availability requirements and enables one of the entities to delegate its private matching capabilities to an untrusted entity without losing privacy of its private set. The subsequent section discusses the limitations of private matching within the context of public cloud storage services. After that DPM along with its security analysis is presented.

4.1 Availability Requirement for Involved Entities

Private matching is a special case of secure multi-party computation. It engages two entities (i.e., client and server) to compute set interaction over their respective private set of values. It enables client to learn common elements between both of the private set - whilst restraining him to learn any value other than interest set. Besides this, server can only learn cordiality of the client's set. Private matching enables secure computation within untrusted domain; however, it restricts involved entities to actively participate (i.e., remain online during the execution of protocol) in the protocol.

Cloud storage services enable their subscribers to share outsourced data with other subscribers - without availability requirements. Thus, a subscriber can outsource its data to a cloud storage service and after that, CSP takes on the responsibility of provisioning outsourced data to other subscribers. This dissertation utilizes private matching as a mechanism to process confidential data (i.e., private set of values) within untrusted domain. Secure computation enables the proposed

methodologies to enforce access control policies and search encrypted data (Chapter 5 and Chapter 6 respectively) presented in this dissertation.

Since, private matching is an interactive protocol it greatly affects the capability of a subscriber to enforcement access control policies and to enable authorized subscribers to search outsourced data. To relax the availability requirement on subscribers outsourcing the data to a cloud storage service, we extended private matching to a non-interactive protocol called delegated private matching protocol. The functional details and security analysis of proposed extension is presented in subsequent sections.

4.2 Oblivious Private Matching in Untrusted Domain

Delegated private matching is an extension of private matching protocol. It relaxes the availability requirement on one of the involved entity - in the context of cloud-based data sharing services it relaxes the data owner's availability requirement. It assists passive entity (i.e., data owner) to delegate the private matching task to a third party (i.e., cloud service provider) without compromising privacy of its private set. The underlying concept of defining a polynomial (\mathcal{P}), from the elements of private set remains the same as discussed in Section 3.2.

To illustrate the conceptual details of delegated private matching we use notion of client \mathcal{C} and server \mathcal{S} - similar to Section 3.2. \mathcal{C} represents the subscriber who wants to compute set interaction; whereas, \mathcal{S} represents the data owner who delegates its private matching capabilities to an untrusted entity.

Both \mathcal{C} and \mathcal{S} have their own private set of values i.e., $\mathcal{X} : \{x_1, x_2 \dots x_n\}$, $\mathcal{Y} : \{y_1, y_2 \dots y_n\}$ respectively. However, \mathcal{S} does not want to compute set interaction, as it will be working as a passive entity. It delegates private matching capability to a third party called Validator (\mathcal{V}) (i.e., cloud service provider). Although set intersection is delegated to \mathcal{V} , still \mathcal{S} does not want \mathcal{V} to learn any information about the private set (\mathcal{Y}). In the subsequent illustration of delegated private matching, it is assumed that \mathcal{S} knows the public key of \mathcal{C} .

\mathcal{S} selects a random mask (\tilde{r}) of an arbitrary length. It then encodes each element of its private set, using a publicly known encoding function i.e., $encode(y, \tilde{r}) \rightarrow \hat{y}$, where $y \in \mathcal{Y}$. Once $\hat{\mathcal{Y}}$ (encoded private set) is computed, \tilde{r} is encrypted with \mathcal{C} 's public key. Finally, encrypted random

number and encoded private set are send to \mathcal{V} . After that availability of \mathcal{S} is not required, \mathcal{V} can obviously evaluate the matching process without compromising privacy of any of the involved entity's private set.

To perform set intersection \mathcal{C} obtains the encrypted random number from \mathcal{V} . It then decrypts it by using its private key. After that it encodes the private set of values by using the same encoding function as did by \mathcal{S} , i.e., $encode(x, \tilde{r}) \rightarrow \hat{x}$, where $x \in \mathcal{X}$. Once $\hat{\mathcal{X}}$ (encoded private set) is computed. \mathcal{C} then computes a polynomial $P(\hat{x})$, whose roots are members of $\hat{\mathcal{X}}$ (see equation 3.8). It then initializes a homomorphic encryption key pair (secret and public key) and encrypts the coefficients $\alpha_{0...n}$ of $P(\hat{x})$ by using homomorphic encryption secret key. After that encrypted coefficients ($\hat{\alpha}_{0...n}$) along with homomorphic encryption public key are sent to \mathcal{V} .

On receiving $\hat{\alpha}_{0...n}$, \mathcal{V} evaluates $P(\hat{y})$ for every element of $\hat{\mathcal{Y}}$, by using $\hat{\alpha}$. It then computes oblivious value by multiplying $P(\hat{y})$ with a random number r and adding it to \hat{y} , i.e., $\mathcal{E}_H(r.P(\hat{y}) + \hat{y})$. These oblivious values are then send to \mathcal{C} .

Finally, to identify the commonalities between \mathcal{X} and \mathcal{Y} , \mathcal{C} decrypts the oblivious values. Decryption of an oblivious value reveals \hat{y} , if $P(\hat{y})$ computed by \mathcal{V} is evaluated at z such that $\langle z \subseteq \bigcap ((z \in \hat{\mathcal{X}}) \wedge (z \in \hat{\mathcal{Y}})) \rangle$. Otherwise \mathcal{C} ends up decrypting a random value.

During the entire execution of DPM, \mathcal{V} learns nothing more than the cardinality of \mathcal{S} 's private set. This can be easily mitigated by adding dummy values. With the amalgam of a public key cryptography and homomorphic encryption \mathcal{V} learns no useful information, yet being able to obviously evaluate the private matching protocol. Figure 4.1 illustrates the interaction between the client, server and validator, along with the availability of each entity during the protocol.

4.3 Security Analysis

There are number of parameters on which security of DPM depends. It utilizes random masks (r) to ensure that only authorized subscribers are able to compute set intersection, without revealing any information to the validator \mathcal{V} . As encoding function is publically known, every element of private set is hashed with random mask before it can be encoded. To make entire process of private matching oblivious to CSP, homomorphic encryption is utilized. It ensures that CSP can process (i.e., evaluate polynomial for each element of delegated private set) encrypted data (i.e., delegated

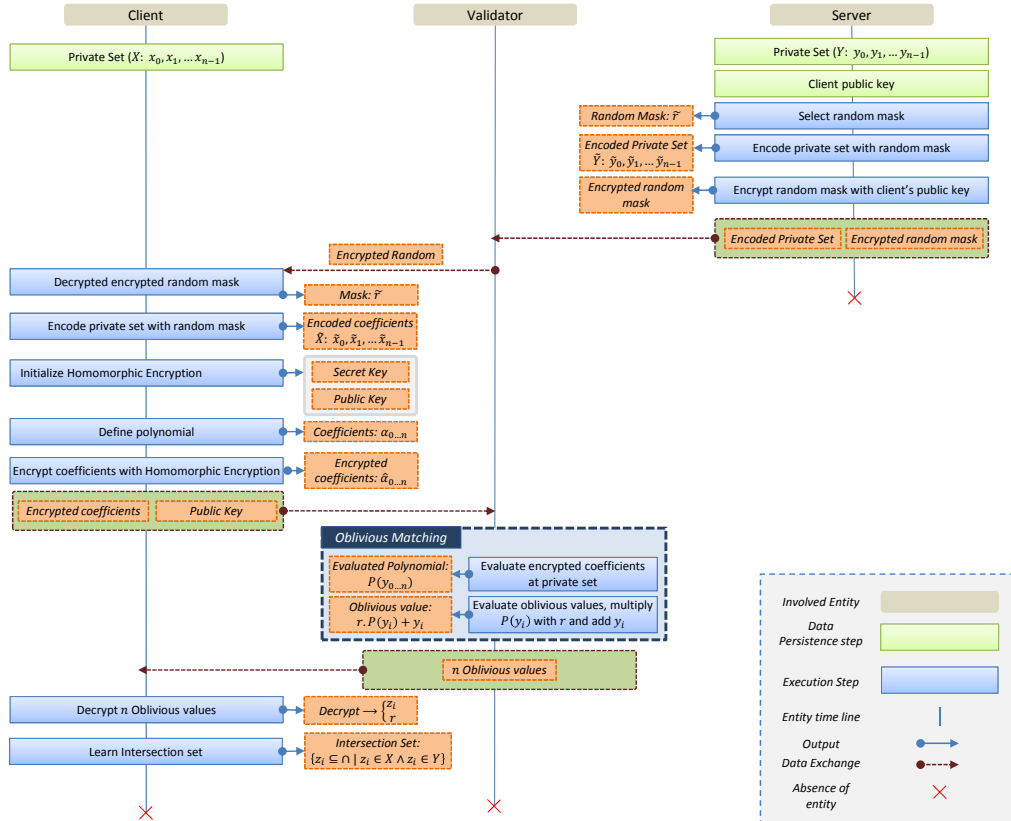


Figure 4.1: Delegate private matching: entity interaction and time-line.

private set of values, $\hat{\mathcal{Y}} : \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ and private matching requests, $(\hat{\alpha}_{0\dots n})$ without the need to decrypt it.

DPM relies on the cryptographic properties of Pseudo Random Generator (PRG) to generate a random mask (r) and one-way hash function H to compute hash of private set along with random mask i.e., $H(\mathcal{PS}_{0\dots n}|r)$, where $\mathcal{PS}_{0\dots n}$ are the elements of private set \mathcal{X} and \mathcal{Y} . By utilizing a secure PRG and one-way hash function, private values are obfuscated from adversary. Furthermore, to restrain unauthorized subscribers from computing set interaction, r is encrypted with public key of authorized subscribers. To process the encrypted data Pascal Paillier is utilized, it is an additively secure homomorphic cryptosystem. For any two numbers x and y , homomorphically added (i.e., $\mathcal{E}_H(x) * \mathcal{E}_H(y) = \mathcal{E}_H(x + y)$) it is computationally infeasible for an adversary to learn whether $x = y$ or $x \neq y$.

Since, DPM relies on polynomial evaluations following proof illustrates that polynomial defining private set of values is unique thus restraining malicious evaluator (\mathcal{V}) to falsely replying to private matching request.

Proof of uniqueness: Suppose, x_0, x_1, \dots, x_n be $n + 1$ distinct data points in interval $[a, b]$. There exists a unique polynomial p of degree n or less that interpolates $f(x)$ at (x_i) , that is, $p(x_i) = f(x_i)$, for $0 \leq i \leq n$.

To prove the uniqueness, the Fundamental Theorem of Algebra states that a polynomial $p(x)$ of degree n and with real or complex coefficients can be factorized over the complex domain into a product $a_n(x - r_1)(x - r_2) \dots (x - r_n)$, where a_n is the leading coefficient and r_1, r_2, \dots, r_n are all of its n complex roots.

Now, suppose p and q are two distinct polynomials of degree at most n that agree at (x_i) , then $p - q$ is a polynomial of degree at most n that vanishes at (x_i) . Therefore, by the Fundamental Theorem of Algebra,

$$(p - q)(x) = c \sum_{i=0}^n (x - x_i) \quad (4.1)$$

where c is some real number. The left hand side has degree at most n , the right hand side has degree exactly $n + 1$ unless $c = 0$. Therefore, $c = 0$ and $p = q$, testify that p is unique.

4.3.1 Malicious Client

Delegated private set of values are persisted by untrusted validator; however, in the current context of malicious client, we consider that client is working independently and validator is not assisting him. To compromise privacy of delegated private set of values ($\hat{\mathcal{Y}} : \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$), malicious client can randomly select its private set of values and use them to execute private matching protocol. Since, $\hat{\mathcal{Y}} : \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ are encoded with random mask r , to properly encode the randomly selected values r is required.

As random mask r is encrypted with public key of authorized subscribers, malicious client would not be able to learn it. In other words malicious client would either has to break the one-way hash function or it would has to learn r encrypted under asymmetric encryption. Clearly, both problems are considered to be computationally infeasible. Thus, there is no way a malicious client can learn any information about $\hat{\mathcal{Y}} : \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ without gaining access access to random mark, or private key of authorized subscriber.

4.3.2 Malicious Validator

Malicious validator can compromise privacy of delegated private set of values ($\hat{\mathcal{Y}} : \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$), by enabling malicious client to learn private set of values in multiple ways. It can either transfer entire $\hat{\mathcal{Y}}$, to malicious client or it can falsely executing the protocol to enable malicious client to learn subset of $\hat{\mathcal{Y}}$, i.e., adding zero value instead of randomized value when evaluating polynomial, $\mathcal{E}_H(0.P(\hat{y}) + \hat{y})$.

Since, $\hat{\mathcal{Y}}$ is hashed with random mask r , malicious validator cannot compare hashed values of randomly selected values with $\hat{\mathcal{Y}}$. Similar to attack scenario discussed above, it would be computationally infeasible for both malicious validator and client to compromise privacy of delegated private set of values. Even if, validator behave maliciously and assist client to learn $\hat{\mathcal{Y}}$, still it will be of no use as adversary do not have access to r and private key that can decrypt r . Both random mark and private key are required to successfully compute private matching with $\hat{\mathcal{Y}}$.

Chapter 5

Oblivious Access Control Policy Evaluation

5.1 Introduction

This chapter explains the theoretical details of Oblivious Access Control Policy Evaluation (O-ACE). The notion of O-ACE is to enable policy evaluator to process encrypted access control policies, without learning any useful information, during the entire process of access control policy evaluation. O-ACE conceals access control policy and access parameters from the policy evaluator, since both can reveal confidential information about the resources protected by the access control policy. Cryptographic primitives and protocols discussed in chapter 3 enable the processing of encrypted access control policy and access parameters. Since, access control policy and access parameters are in encrypted form, the result of access control policy evaluation is oblivious to a policy evaluator.

The aim of O-ACE is to realize a privacy-aware access control policy evaluation that can govern access to the resources protected by an access control policy. Privacy-awareness ensures that policy evaluator cannot utilize the procedure of access control policy evaluation to compromise privacy of the resources. Subsequent sections present the conceptual, and security models of O-ACE along with its design goals, and assumptions in a context of cloud-based data sharing system. An abstract idea of proposed access control policy evaluation is presented followed by the descriptive detail of O-ACE.

5.2 Models, Design Goals and Assumptions

The proposed access control policy evaluation is designed for an untrusted domain in which policy evaluator can exploit policy evaluation procedure to compromise privacy of the confidential

resources (i.e., outsourced data, application/software services). Before presenting the details of O-ACE, it is important to understand the conceptual, and security models for which O-ACE is designed. System design goals are also presented to illustrate its functional importance for privacy-aware data sharing in the context of following models.

5.2.1 Conceptual Model

To enforce access control policy in an untrusted cloud storage system: credential issuing authority, data provider, data consumer, and cloud storage service provider are considered as the involved entities. For brevity these entities are referred as authority, owner, user, and cloud server respectively. Authority is a trusted entity which takes on the responsibility of issuing identity attributes to the users and their identity assertions to the owner. Owner utilizes the storage facility provided by the cloud server, by outsourcing the encrypted data, that need to be shared e.g., pictures, text documents, multimedia files, along with the encrypted access control policy. Cloud server obviously evaluates the access control policy and consequently provision to the outsourced data only to the authorized user.

5.2.2 Security Model

Threats faced by the owner when outsourcing confidential data to a cloud server can be primarily divided into two categories, internal and external threats. **Internal threats:** cloud server himself is interested in the outsourced data for some business needs (*i.e., related ad serving, selling confidential data for some wicked motives*). **External threats:** fraudulent user seeks access to the outsourced data, otherwise not allowed.

Internal threats can be reasonably mitigated by the use of appropriate encryption algorithms. Once data is encrypted, cloud server cannot learn any information from it that can be utilized to compromise privacy of the outsourced data. However, these algorithms fail to obstruct the external threats, when users can derive valid decryption key and access outsourced data according to their access parameters. Over the time cloud server can learn the access parameters which are required to generate a valid decryption key, consequently assists malicious users to compromise privacy of the outsourced data.

To eliminate the external threats owner must restrain illicit data access by defining access control policy and enforcing it in the untrusted domain. Since, cloud server is not a trustable entity access control policy should have a property of obliviousness. In other words, cloud server should not be able to differentiate between an authorized and fraudulent user, yet being able to assist authorized user in deriving valid decryption key. This ensures that cloud server cannot learn any information about the access parameters that can be used to derive valid decryption key.

Cloud storage can be utilized to store and share personal information i.e., healthcare and financial records. In this context, cloud server can exploit access parameters of a user who seeks access to the outsourced data. Access parameters can assist cloud server to learn confidential about the outsourced data. For example, if a doctor specialized in diabetes mauritius, seeks access to the outsourced data (medical reports), then cloud server can reasonably infer that owner is a diabetic patient, or is concerned about her diabetic readings. Access parameters of a user must be encrypted to restrain cloud server from leaving any information that can lead to a potential privacy breach. Obliviousness property discussed earlier ensures that access control policy evaluation in a domain of cloud server is similar to a block box. Its input is a set of encrypted access parameters and its outputs assist authorized users to gain access to the outsourced data.

5.2.3 System Design Goals

The proposed access control policy evaluation framework is envisioned as a cloud storage system that enables the owner to share confidential data with authorized users, without compromising privacy of the outsourced data. As owner cannot remain always online to enforce access control policy, cloud server must obviously evaluates the access privileges of a user, and disseminate the appropriate secret values, which can be used to derive valid decryption key. Since, cloud server is not a trustable entity, owner must be able to define access control policy which does reveal any information about the attributes required to decrypt the outsourced data. Besides this, these policies must not lose their efficacy in restricting unauthorized users to successfully decrypt the outsourced data.

The pivotal design goal of O-ACE is to ensure privacy of the involved entities (owner, and user) and outsourced data. Specifically, from owner's perspective O-ACE must:

- associates authenticity of data access with the identity attributes of a user
- evaluates encrypted access control policy on cloud server
- does not disclose result of policy evaluation to cloud server
- enables authorized users to derive data decryption key

whereas, from user's point of view O-ACE must:

- ensures privacy of identity attributes
- provisions access to the outsourced data, without learning any personal information of a user

5.2.4 Assumption and Notations

In the descriptive detail of O-ACE following assumptions are considered with the intent of simplicity. These assumptions conforms to the security model, and does not undermine any of the privacy threat discussed in Section 5.2.2.

- Cloud server is honest but curious i.e., cloud server performs the delegated task honestly but is also interested in the contents of the outsourced data - similar to [88].
- There exists a trustable authority, which issues identity attributes to individual users¹.
- Outsourced data is shared with the users to whom authority can issue identity attributes.
- There is no concern of privacy infringement in asserting identity attributes of a user to the owner. These attributes are merely account for uniquely delineating the user within an organization².

In order to elaborate O-ACE generally, we intentionally avoid differentiating cryptographic keys of individual users. Throughout the descriptive detail of O-ACE, cryptographic keys for a

¹Authority can issue a root level certificate to an organization, which uses it to sign X.509 v3 certificate (*attribute certificate*) of its employees [89].

²In Section 5.6 we discuss how to prevent adversary from gaining knowledge about the user attributes and then exploiting them to access the outsourced data.

Table 5.1: Notations used in the descriptive detail of O-ACE.

<i>Notation</i>	<i>Description</i>
\mathcal{F}	Data to be shared with legitimate users.
$att_{0...n}$	Attributes issued to a user by the authority.
\tilde{r}	Randomly selected mask i.e., $\tilde{r} \in_R \mathbb{Z}_q$.
H_1	One way hash function which encodes att_i to an integer of arbitrary length i.e., $H_1(att_i): \{0, 1\}^* \rightarrow \mathbb{Z}$.
H_2	One way hash function which encodes att_i to integer of module q , where q is a prime i.e., $H_2(att_i): \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$.
l_i	Legitimacy value: att_i encoded by H_1 .
\hat{l}_i	Mandatory value: att_i encoded by H_2 .
ψ	Pseudo random function which outputs the symmetric encryption key of an arbitrary length.
ω	Legitimacy key: encrypts the legitimacy values.
$\mathcal{E}_H, \mathcal{D}_H$	Homomorphic encryption and decryption algorithms.
σ_{pk}, σ_{sk}	Public and secret key for homomorphic encryption.
$\mathcal{E}_S, \mathcal{D}_S$	Symmetric encryption and decryption algorithms.
κ	Master key (Encryption \ Decryption key) for an arbitrary symmetric encryption.
$\mathcal{E}_A, \mathcal{D}_A$	Asymmetric encryption and decryption algorithms.
k_{pub}, k_{pri}	Public and private key pair for asymmetric encryption.

particular user j can be referred by affixing j as a subscript, without changing the actual usage semantics i.e., δ_u can be binded to a user j as δ_{u_j} , same applies to the rest of keys. Table 6.1 illustrates the notations that are used to explain the core concept of O-ACE.

5.3 Application Scenario and Abstract Idea of Oblivious Access Control Policy Evaluation

We briefly present the application scenario in which access control policy is evaluated obliviously to ensure privacy of the outsourced data in an untrusted domain. Abstract idea of O-ACE is also discussed. On the basis of following scenario, abstract idea will be elaborated with its technical detail in the subsequent section.

Suppose Bob is an under cover agent working on some special assignment. He has collected substantial evidence against a drug lord in downtown area, which he wants to share only with the concerned authorities. Since, Bob cannot presents the evidence (data) in person to the authorities, he decides to use cloud storage facility provided by the Eve. However, due to the sensitivity of the data, Bob does not trust Eve and wants to upload (outsource) the encrypted data. As Bob is working under cover for a quite long time, he does not know the lead detective who in charge of drugs related criminal cases in downtown area.

Bob contacts the Home Office (authority) which asserts the identity information of a lead detective responsible for crime related to drugs in downtown area. Since, authority has the information about each employee, it asserts the identity attributes of Alice, along with her public key. Bob encodes the asserted attributes into legitimacy values by using a publicly known encoding function and into mandatory values by employing a private encoding function. Mandatory values are used to derive the master key that encrypts the evidence (data) with arbitrary symmetric encryption algorithm. Whereas, legitimacy values initialize a pseudo random function which generates the encryption key of an arbitrary symmetric encryption algorithm, that encrypts the respective mandatory values. Legitimacy values are then concealed with symmetric encryption algorithm. Finally, encrypted evidence is outsourced to the cloud server along with concealed legitimacy and mandatory values.

Out of bounds, Alice receives information about the shared contents. She encodes her identity attributes into legitimacy values and engages in an oblivious policy evaluation protocol with Eve, as a result she learns the mandatory values. During the oblivious policy evaluation, Eve neither learns the legitimacy values nor the mandatory values. Whereas, Alice only learns the concealed mandatory values if she holds the required set of attributes used by Bob to encode the legitimacy and mandatory values. Once, Alice has the concealed mandatory values, she uses the legitimacy values to decipher them and consequently derives the master key. Ultimately, she decrypts the outsourced evidence by using the master key. Fig. 5.1 illustrates the conceptual model of our proposed cloud-based data sharing system with oblivious access control policy evaluation.

5.4 Enforcing Oblivious Access Control Policy for Cloud-based Data Sharing

In this section we present the technical detail of O-ACE in the context of cloud-based data sharing service, in which outsourced data, access control policy and identity attributes of a user are considered as confidential information. The ration of considering access control policy and identity attributes as confidential information is discussed at length in Section 5.2.2. O-ACE utilizes Delegated Private Matching (DPM) along with identity attributes, in such a way that possession of certain identity attributes validates the legitimacy and authenticity of a user. We have coupled the derivation of master key with the identity attributes, such that user possessing a valid set of attributes can learn the information which can generate the valid master key.

To ensure privacy of the outsourced data and involved entities (owner, and user), O-ACE processes the data in three fundamental steps, initialization, data outsourcing, and file access. These steps ensures that outsourced data can only be accessed (decrypted) by authorized users, and during the whole process cloud server is unable to learn any useful information that can lead to a potential privacy breach. Fig. 5.2 illustrates the proposed system in terms of exchange of data and availability of the entities.

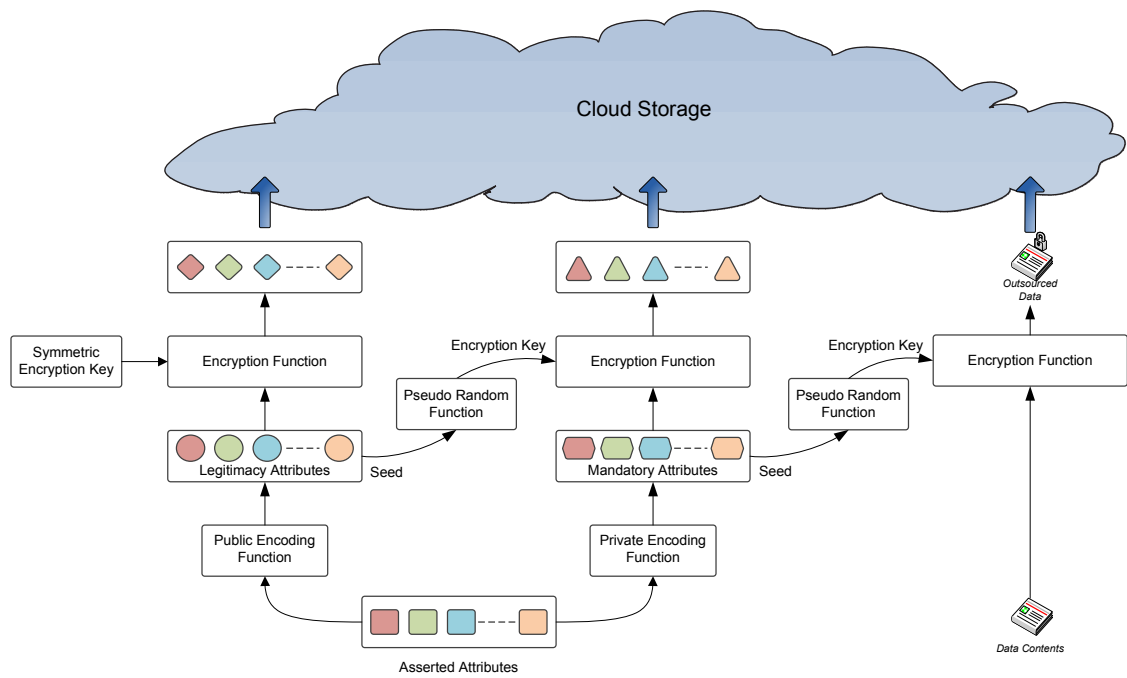


Figure 5.1: Oblivious access control policy evaluation (O-ACE) for cloud-based data sharing.

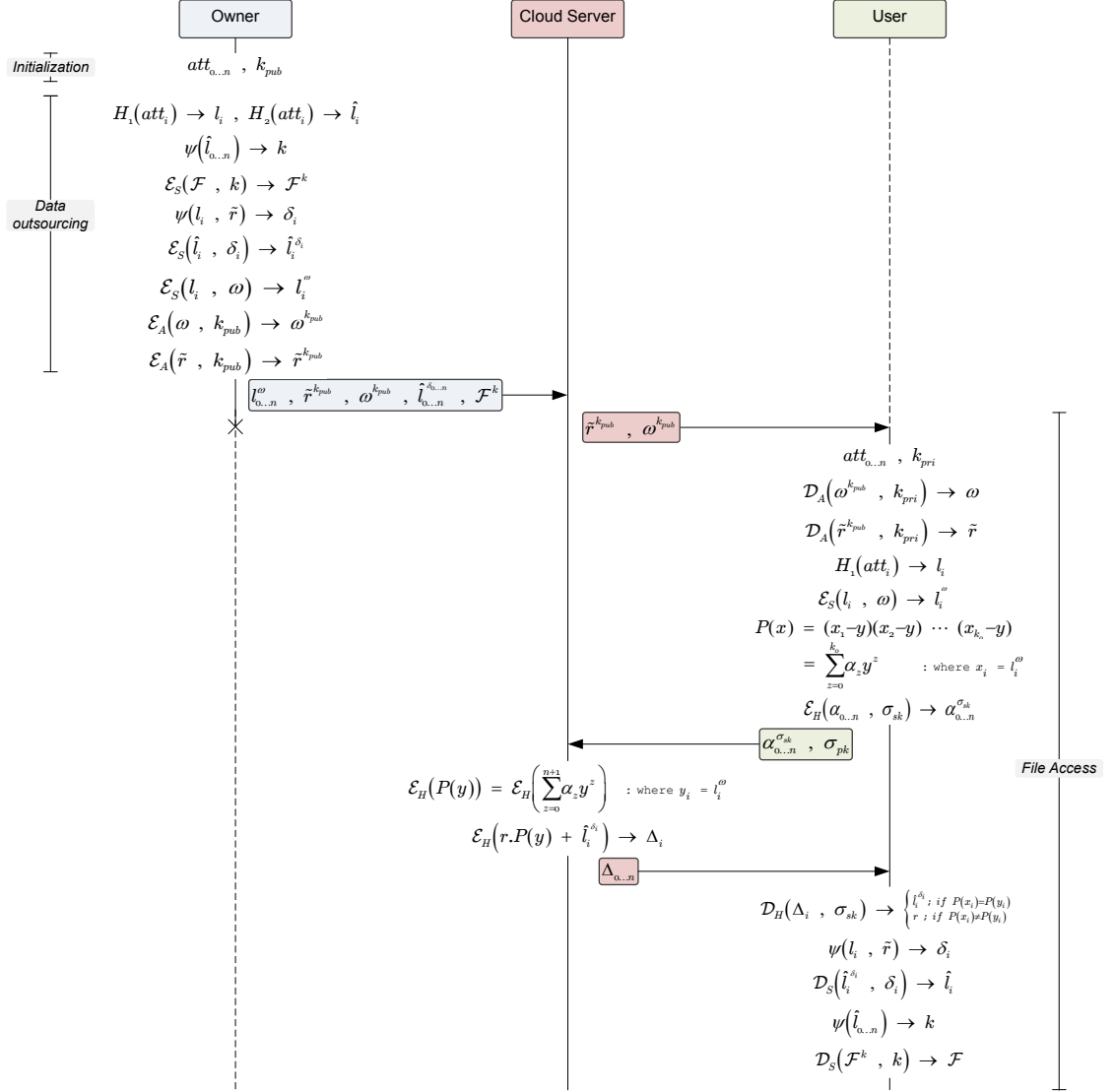


Figure 5.2: Exchange of private values between the owner, cloud server and user during O-ACE.

5.4.1 Initialization

In order to share confidential data, owner contacts the authority to obtain identity information of the target user. It specifies the identity selection criteria e.g., Department: Drug Control Division, Designation: Lead Detective, In-charge of: Downtown Area. In response authority asserts the identity attributes ($att_{0...n}$) of an employee (user) that fulfills the identity selection criteria. Besides this, authority also provides the public key (k_{pub}) of the selected user.

5.4.2 Data Outsourcing

Once owner has the asserted identity information ($att_{0...n}$) along with the user public key (k_{pub}), it processes them in such a way that only legitimate user manages to gain access to the outsourced data. Data outsourcing is further divided into four cohesive steps, policy modeling, data concealment, policy concealment, and delegation.

- *Policy Modeling*: Asserted identity attributes ($att_{0...n}$), that uniquely defines the target user are used to model the access control policy. By using the encoding functions owner exploits these attributes in such a way that their possession ensures the legitimacy and authenticity of the user. For each asserted identity attribute (att_i), owner computes legitimacy and mandatory values by using the encoding functions (i.e., $l_i : H_1(att_i)$ and $\hat{l}_i : H_2(att_i)$), where H_1 and H_2 are public and private encoding functions respectively. The rationale of applying two separate encoding functions is elaborated in the subsequent steps.
- *Data Concealment*: Mandatory values ($\hat{l}_{0...n}$) generated in the previous step are used to seed the pseudo random function (ψ) which initializes the encryption key (master key : k) of an arbitrary symmetric encryption algorithm. The derived master key encrypts the data (\mathcal{F}^k), which is then outsourced to the cloud server. For brevity we consider that these mandatory values are used in a cascading manner, resulting in the derivation of a single master key (i.e., $\psi(\hat{l}_0) \xrightarrow{k_0} \psi(\hat{l}_1, k_0) \xrightarrow{k_1} \dots \xrightarrow{k_{n-1}} \psi(\hat{l}_n, k_{n-1}) \rightarrow k$) that conceals the data. However, each mandatory value can be used to derive a unique master key (i.e., $\psi(\hat{l}_i) \rightarrow k_i$) that encrypts the respective data block.
- *Policy Concealment*: Mandatory values ($\hat{l}_{0...n}$) ensures that master key can only be derived

by the user possessing these values. In order to conceal mandatory values, a random mask (\tilde{r}) is generated. Then, for each mandatory value \hat{l}_i , a pseudo random function (ψ) is initialized with the corresponding legitimacy value l_i and \tilde{r} ; consequently, a symmetric encryption key (mandatory key: δ_i) is generated (i.e., $\psi(l_i, \tilde{r}) \rightarrow \delta_i$). Each individual \hat{l}_i is encrypted with the respective δ_i . As the policy evaluation is carried on the cloud sever by using the legitimacy values, there is a need to conceal them as well. For that owner generates a random symmetric encryption key (legitimacy key: ω), and encrypts each legitimacy values with it (i.e., $l_{0...n}^\omega$).

Once, the legitimacy and mandatory values are encrypted, random mask and legitimacy key are concealed by using the public key (k_{pub}) of the target user, obtained during the initialization phase.

- *Delegation*: Up till now, the owner has modeled the access control policy by encrypting the data with mandatory values. Also, mandatory values are concealed in such a way that, only authorized user can learn them. Now owner delegates the policy evaluation process to the cloud server by outsourcing the encrypted legitimacy and mandatory values ($l_{0...n}^\omega$ and $\hat{l}_{0...n}^{\delta_{0...n}}$), along with the concealed random mask and legitimacy key ($\tilde{r}^{k_{pub}}$ and $\omega^{k_{pub}}$).

5.4.3 File Access

In order to gain access to the outsourced data authorized user needs mandatory values. For that user engages in a delegated private matching protocol (DPM) with the cloud server, at the end of which user learns the mandatory values, if it possess the required set of legitimacy values. To evaluate the access control police and assist authorized user in deriving the valid master key, file access is divided into three cohesive step. In the first step user ensures the privacy of its identity attributes. In the second step access control policy is obviously evaluated at the cloud server, and in the last step, user accesses (decrypts) the outsourced data if its identity attributes adhere to the access control policy.

- *Attribute Preparation*: To access the outsourced data, user first obtains the concealed random mask ($\tilde{r}^{k_{pub}}$) and legitimacy key ($\omega^{k_{pub}}$) from the cloud server. Then by using k_{pri} it

deciphers them to get \tilde{r} and ω . After that, for each of its identity attribute att_i , legitimacy value (l_i) is computed as $H_1(att_i)$. Legitimacy values ($l_{0...n}$) are then concealed by using ω . A polynomial $\mathcal{P}(y)$ is computed having roots $l_{0...n}^\omega$. User then initializes a homomorphic encryption and encrypts the co-efficients ($\alpha_{0...n}$) of $\mathcal{P}(y)$ with the homomorphic secret key (σ_{sk}). After that, encrypted co-efficients ($\alpha_{0...n}^{\sigma_{sk}}$) along with homomorphic public key (σ_{pk}) are sent to the cloud server.

- *Policy Evaluation:* On receiving encrypted co-efficients ($\alpha_{0...n}^{\sigma_{sk}}$), cloud server homomorphically evaluates $\mathcal{P}(y)$ with encrypted co-efficients ($\alpha_{0...n}^{\sigma_{sk}}$), for the concealed legitimacy values provided by the owner ($l_{0...n}^\omega$). Once $\mathcal{P}(y)$ is evaluated, cloud server computes oblivious value (Δ_i) as $\mathcal{E}_H(r \cdot \mathcal{P}(y) + \hat{l}_i^{\delta_i})$, for each of the concealed mandatory value ($\hat{l}_i^{\delta_i}$). Then the resultant oblivious values ($\Delta_{0...n}$) are sent to the user.
- *Mandatory Value Recovery:* On receiving oblivious values ($\Delta_{0...n}$) user utilizes its homomorphic secret key (σ_{sk}) to decrypt ($r \cdot \mathcal{P}(y) + \hat{l}_i^{\delta_i}$). As $\mathcal{P}(y)$ was evaluated having roots l_i^ω , the decryption function reveals the concealed mandatory value ($\hat{l}_i^{\delta_i}$). However, if the concealed legitimacy values generated by the user does not match with the values provided by the owner to the cloud server, decryption of Δ_i would reveal a random number, thus restraining the user to generate a valid master key.

Once, user has the concealed mandatory values ($\hat{l}_{0...n}^{\delta_{0...n}}$), it utilizes the legitimacy value (l_i) along with the random mask (\tilde{r}) to initialize pseudo random function (ψ) which generates the mandatory key (i.e., $\psi(l_i, \tilde{r}) \rightarrow \delta_i$). It then deciphers the concealed mandatory values by using the mandatory keys. Once, all of the mandatory values are obtained master key is derived as illustrated in Section 5.4.2 (data concealment). Eventually outsourced data is accessed by using the master key.

5.5 Complexity Analysis

In this section we illustrate the computational complexity of proposed methodology of access control enforcement within the untrusted domain of cloud service provider³. Following analy-

³For the sake of simplicity, cryptographic operations are considered to take constant time.

Table 5.2: O-ACE computational complexity and estimation of transmitted values.

O-ACE Steps	Operations	Input size	Computational Complexity	Transmitted Values
Policy Modeling	<i>Public Encoding</i>	N	$O(N)$	$(2N + 2)$ values
	<i>Private Encoding</i>			
	<i>Asymmetric Encryption</i>			
Data Access Request	<i>Attribute Preparation</i>	n	$O(n^3)$	$(n + 2)$ values
	<i>Polynomial Modeling</i>			
	<i>Asymmetric Encryption</i>			
Access Control Evaluation	<i>Polynomial Evaluation</i>	$(n + 1) \cdot N$	$O(n^2 \cdot N)$	N values
Mandatory Value Recovery	<i>Asymmetric Decryption</i>	N	$O(N)$	

sis consider the estimated steps required to model, enforce and evaluate oblivious access control policy. Data that need to be transmitted at each step is also considered. Table 5.2 delineates the computational complexity along with the transmitted data. Both computational complexity and transmitted data are directly proportional to number of identity assertions used to model access control policy and identity attributes used to access encrypted outsourced data.

5.5.1 Policy Modeling

Policy modeling comprises of two encoding functions and an asymmetric encryption function. Encoding functions transform identity assertions into legitimacy and mandatory values. Each encoding function is executed N times for every identity assertion, where N is the number of identity assertions used to derive data encryption key. With Big- O notation computational complexity of policy modeling can be described as $O(N)$. In total $2N + 2$ values are transmitted to the cloud server i.e., N legitimacy and mandatory value pairs, a random seed used to derive mandatory key and a symmetric key to conceal random seed.

5.5.2 Data Access Request

Outsourced data can be accessed by engaging in oblivious access control policy evaluation with the cloud server and learning mandatory values. Data access request comprises of attribute preparation (public encoding function), polynomial modeling, and an asymmetric encryption operation. These operations process n identity attributes, thus require $O(n^3)$ computational time to prepare data

access request (i.e., polynomial modeling and coefficient sign check). Since, n identity attributes are used to prepare data access request, the degree of polynomial defined by polynomial modeling is $n + 1$. In total $n + 2$ values are transmitted to the cloud server, $n + 1$ polynomial co-efficients and a public key for homomorphic operations over encrypted access control policy.

5.5.3 Access Control Evaluation

Oblivious access control evaluation comprises of a single operation (polynomial evaluation). Access control policy is evaluated for $n + 1$ number of polynomial co-efficients provided by a subscriber. Polynomial is individually evaluated for all the N concealed legitimacy values provided by the data owner. The computational complexity of access control evaluation depends on number of identity assertions used to model access control policy (N) and number of identity attribute used to generate data access request - with Big- O notation it can be expressed as $O(n^2.N)$. In total N oblivious values are transmitted back to the subscriber.

5.5.4 Mandatory Value Recovery

To learn the result of oblivious access control policy evaluation, subscriber needs to decrypt the oblivious response. Mandatory values recovery comprises of asymmetric decryption process. It processes N oblivious values transmitted by the cloud server. The computational complexity to learn mandatory values is $O(N)$. In total subscribers deciphers N values; however it can only learn those values for which it has corresponding valid identity attributes i.e., set intersection between identity assertions and identity attributes.

5.6 Security Analysis

In this section we exam the computational complexity of a malicious user to gain access to the outsourced data. For that, a malicious user would need mandatory values ($\hat{l}_{0...n}$) with which valid master key can be derived. The proposed system uses the standard cryptographic primitives to ensure privacy of the outsourced data; thus, inheriting their computational complexities. As discussed in Section 5.4 and 5.7, O-ACE uses one way hash function to model the access control

policy. Symmetric encryption function is used to ensure that only authorized users can decrypt the response send by the cloud server. Private matching is used to obviously evaluate the access control policy against the identity attributes provided by the user. Homomorphic encryption is used to ensure privacy of access control policy and identity attributes. Reader may refer to [85], [86], and [90] for the security analysis of aforementioned cryptographic primitives.

As discussed O-ACE inherits the computational complexities of the underlying cryptographic primitives. However, cloud server, client and even authority can act maliciously by teaming up with each other to compromise privacy of the outsourced data. In the subsequent security analysis we examine O-ACE against the malicious behavior of the involved entities. First, we discuss the efficacy of O-ACE when cloud server does not perform its task honestly, falsifying our assumption that cloud server is honest but curious (see Section 5.2.4). Second, we discuss, up to what extent O-ACE provides protection when multiple malicious users can learn the partial set of mandatory values ($\hat{l}_{0...n}$) and combine them to derive a master key. Third, we review the efficacy of O-ACE when authority seeps out information about the asserted identity attributes of unauthorized users.

5.6.1 Malicious Cloud Server

There are two possibilities through which cloud server can assist unauthorized users to learn the mandatory values ($\hat{l}_{0...n}$). Firstly, bypassing the access control policy evaluation and simply handing over the concealed mandatory values ($\hat{l}_{0...n}^{\delta_{0...n}}$) to the user(s). Secondly, by incorrectly evaluating the access control policy i.e., instead of the adding a random value (r) to compute oblivious values (i.e., $\Delta_{0...n}$) it adds a zero value. In both of these cases unauthorized user learns the mandatory values ($\hat{l}_{0...n}^{\delta_{0...n}}$) that are encrypted with the masked legitimacy values i.e., $\psi(l_{0...n}, \tilde{r}) \rightarrow \delta_{0...n}$. In order to decipher the mandatory values unauthorized user would have to gain access to the encoded attributes ($H_1(att_{0...n}) \rightarrow l_{0...n}$), and random mask (\tilde{r}) which is encrypted with the authorized user's public key i.e., $\tilde{r}^{k_{pub}}$.

As access control policy is obviously evaluated, cloud server cannot identify the user that can access the outsourced data successfully. Thus malicious user would have to try all possible encoded attributes in order to learn the correct legitimacy values ($l_{0...n}$). This could be a trivial task if possible identity attributes are limited as legitimacy value can be learned by simply apply-

ing the publicly known encoding function. However, to mitigate this threat we have masked the legitimacy value before it can be used to conceal the mandatory values i.e., $\psi(l_{0...n}, \tilde{r}) \rightarrow \delta_{0...n}$. This random mask (\tilde{r}) is encrypted with the authorized user's public key. In order to decipher the mandatory values unauthorized user would have to decipher $\tilde{r}^{k_{pub}}$ without private key, thus making its computational complexity equal to asymmetric encryption algorithm.

5.6.2 Malicious Clients

We now consider the scenario in which cloud server is performing its task honestly; however, unauthorized users team up to gain access to the outsourced data otherwise not allowed. In order to successfully gain access to the outsourced data, unauthorized users must team up in such a way that collectively they can learn the mandatory values ($\hat{l}_{0...n}$) which can be used to derive the master key. As access control policy is obviously evaluated on the cloud server, unauthorized users cannot identify the identity attributes ($att_{0...n}$) which conform to the access control policy. Individual unauthorized user would have to learn subset of the mandatory values ($\hat{l}_{0...\hat{n}} \subset \hat{l}_{0...n}$) and then combine them altogether to learn the entire set i.e., $\hat{l}_{0...\hat{n}_1} \cup \hat{l}_{0...\hat{n}_2} \cup \dots \hat{l}_{0...\hat{n}_n} = \hat{l}_{0...n}$. It is a nontrivial task to identity the team members that can completely learn the mandatory values ($\hat{l}_{0...n}$).

Similar to the scenario discussed in the previous section, if there are limited number of the identity attributes then unauthorized users can effortlessly combine their partial set of mandatory values. Although mandatory values can be obtained but still unauthorized users need the random mask (\tilde{r}) which is used to mask the legitimacy values (i.e., $\psi(l_{0...n}, \tilde{r}) \rightarrow \delta_{0...n}$). As cloud server is working honestly unauthorized users cannot obtain the random mask encrypted with the user's public key with whom data owner wants to share the outsourced data. Even if random mask can be learned still unauthorized users would have to decipher $\tilde{r}^{k_{pub}}$ without private key (k_{pri}), making its computational complexity equal to asymmetric encryption algorithm.

5.6.3 Malicious Identity Provider

O-ACE is highly dependent on the identity assertions and attributes provided by an authority. In the descriptive detail of O-ACE we assumed that authority is a trustable entity (see Section 5.2.4).

For the security analysis we now consider that the authority is working maliciously and seeps out information about the identity assertions and attributes to unauthorized users and cloud server. They can then use it to learn the mandatory values $(\tilde{l}_{0...n})$. However; similar to the scenarios discussed previously access to the mandatory values $(\tilde{l}_{0...n})$ do not compromise privacy of the outsourced data. The attacker would need the random mask (\tilde{r}) , which masks the legitimacy values. As \tilde{r} is concealed with the authorized user's public key the attacker would have to revert back the asymmetric encryption without private key (k_{pri}) .

5.7 Implementation

To demonstrate the practicality of O-ACE, delegated private matching process is implemented as a Java Web service and deployed on Google App Engine. The functionality of authority (trusted third party) is emblemized as a web service which issues identity certificates and assertions. Confidential data (documents and images) are outsourced to the Google Cloud i.e., Google Docs. The owner and client components are realized as a standard Java program as well as an Android based mobile application. Owner component assists the owner to process the identity assertions and outsource encrypted data along with access control policy to Google Cloud. Client component processes the user's identity certificate and access the outsourced data after learning mandatory values.

Implementation of O-ACE utilizes the standard cryptographic primitives provided by the jdk 1.6. O-ACE is realized as

- X.509 v3 certificates [89] are issued as identity certificates, created by using Bounty Castle API [91].
- Security Assertion Markup Language (SAML) [92] is utilized to assert identity assertions.
- Public encoding function uses SHA-512 as a hash function. The output of of SHA-512 is encoded as a BigInteger of arbitrary length.
- Private encoding function uses HMAC (SHA-512) with key length of 512 bits. The output of SHA-512 is encoded as a BigInteger of arbitrary length.

- Pascal Paillier cryptosystem is utilized as a homomorphic cryptosystem to ensure privacy of the data involved in delegated private matching.

Outsourced data is encrypted with AES by using 256 bit key. AES encryption key is initialized with the mandatory values. However, as the proposed system is not confined to any specific encryption algorithm, AES can be replaced with any suitable encryption algorithm, according to the security needs and computational limitations of the owner and user. Besides this the implementation of delegated private matching is not bound to any cloud provider. Google App Engine is selected for its native support of Java; however, the implementation of can be deployed to any cloud infrastructure or platform (i.e., Amazon EC2, Microsoft Azure) that provide Java runtime and can persist private set of values (i.e., $l_{0...n}^{\omega}$ and $\hat{l}_{0...n}^{\delta}$).

5.8 Evaluation

In this section implementation of O-ACE discussed in Section 5.7 is evaluated when deployed in real environment. The evaluation process identifies the computational requirements and storage cost of O-ACE in cloud infrastructure. It also highlights the fact that for the data owner and client O-ACE exerts reasonable computational load and can be deployed on devices having limited computational and storage capabilities i.e., smartphones and tablets.

The evaluation process is divided into two phases. First phase examines the access control policy evaluation processing time in Google AppEngine by using a single compute node having processing power of 1.20 GHz. Second phase evaluates the computation time of the data owner and client components on a PC having 2.60 GHz Dual Core Processor with 2.0 GB main memory. In addition, owner and client components are also evaluated on an Android device having 800MHz processing power.

The discussed time measurements are in milliseconds, and are averaged over 25 different trials. The evaluation results do not consider any subsequent network transmission time i.e., time required to obtain identity assertions and certificate, and network latency of Google AppEngine. These parameters are network dependent and are beyond the scope of our evaluation process.

5.8.1 Phase 1: Performance Analysis of Access Control Policy Evaluation on Google AppEngine

For the computational analysis of access control policy evaluation on Google AppEngine, execution time of a billable CPU and estimated CPU usage cost for 1000 request (cpm_usd), are considered. Figure 5.3 shows the test result of access control policy evaluation, comprises of different number of attributes i.e., 2, 4, 6, 8, and 10.

The computational time in Figure 5.3, is directly proportional to the number of attributes in the access control policy. To evaluate access control policy Google AppEngine manipulates the polynomial co-efficients provided by the data owner, over the values sent by the client (i.e., delegated private matching). To model access control policy with n , attributes $n + 1$ co-efficients are required. Thus, access control policies involving higher number of attributes are modeled with polynomial of higher degree as compared to access control policies having fewer attributes. Access control policies are modeled with arbitrary sized integer values (starting from 120 bit), thus the data points that need to be satisfied during access control policy evaluation demands more computational time. However, the size of the integer values (encoded identity assertions) can be changed according to the computational capabilities of the data owner and client.

5.8.2 Phase 2: Performance Analysis of Data Owner and Client Components

To evaluate the computational complexity of O-ACE for data owner, time required to model the access control policy is considered. Whereas, for client time required to process the identity attributes for delegated private matching and to obtain the mandatory values are measured. Figure 5.4 presents the test results for the data owner, on a personal computer and mobile device. Similarly, Figure 5.5 and 5.6 show the computational cost for a client application deployed on a personal computer and mobile device.

Owner component is only responsible for Policy Modeling by encoding the identity assertions into mandatory and legitimacy values and then further concealing the mandatory values with legitimacy values. Policy Modeling comprises of two hashing function (public and private encoding) and a symmetric encryption function.

Computational complexity of a client component is more than the owner component, as it need

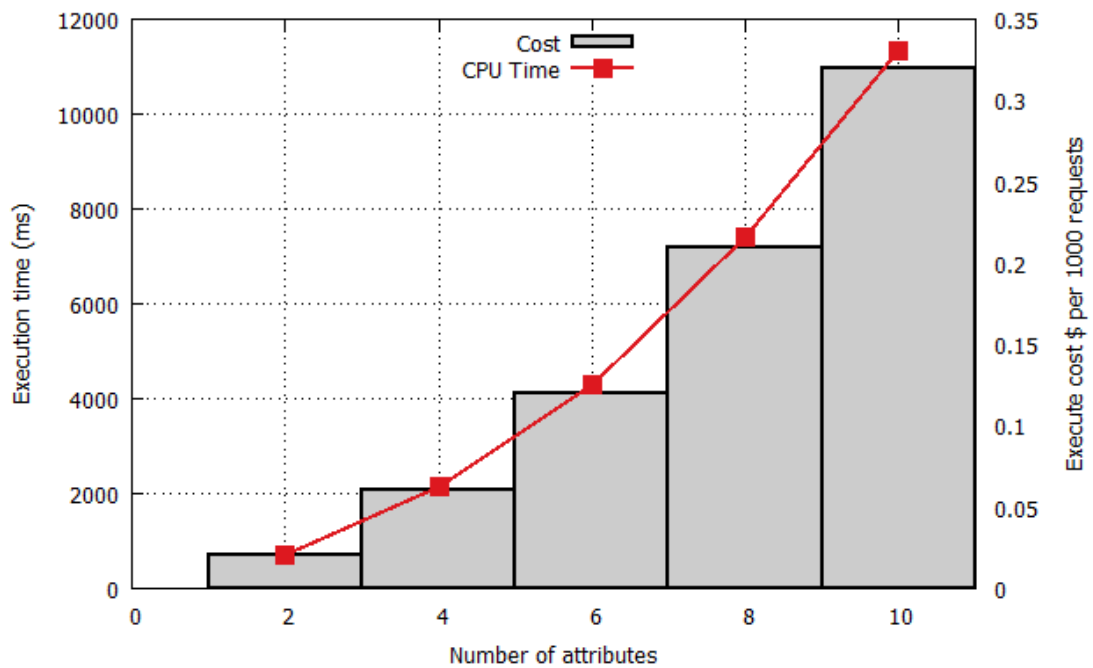


Figure 5.3: Computational time and cost of access control policy evaluation on Google App Engine.

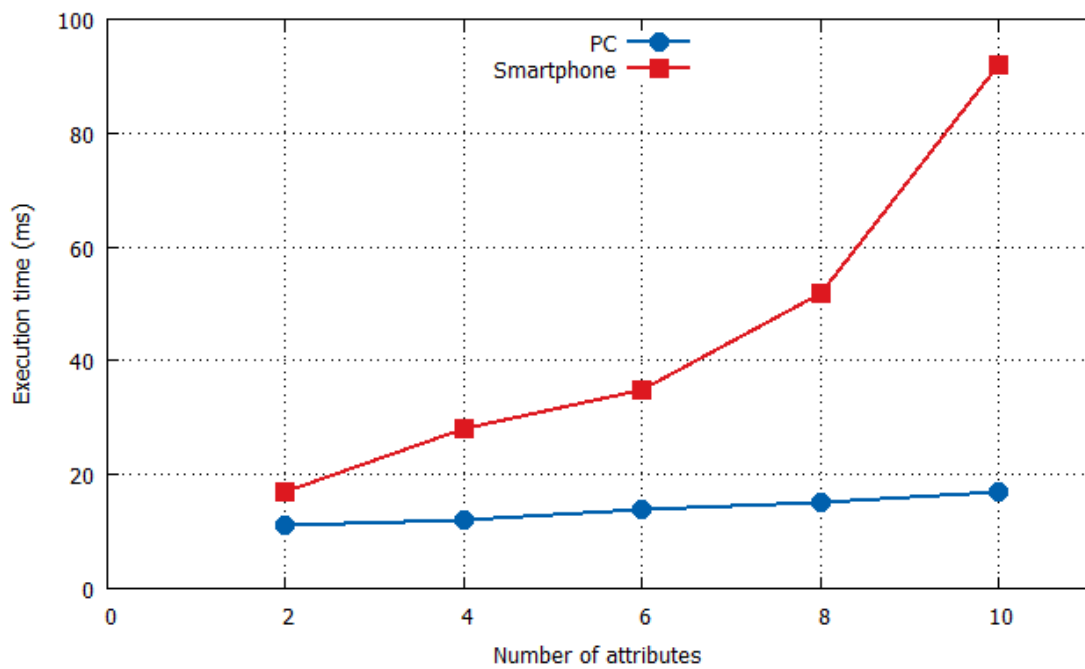


Figure 5.4: Computational time required to model the access control policy.

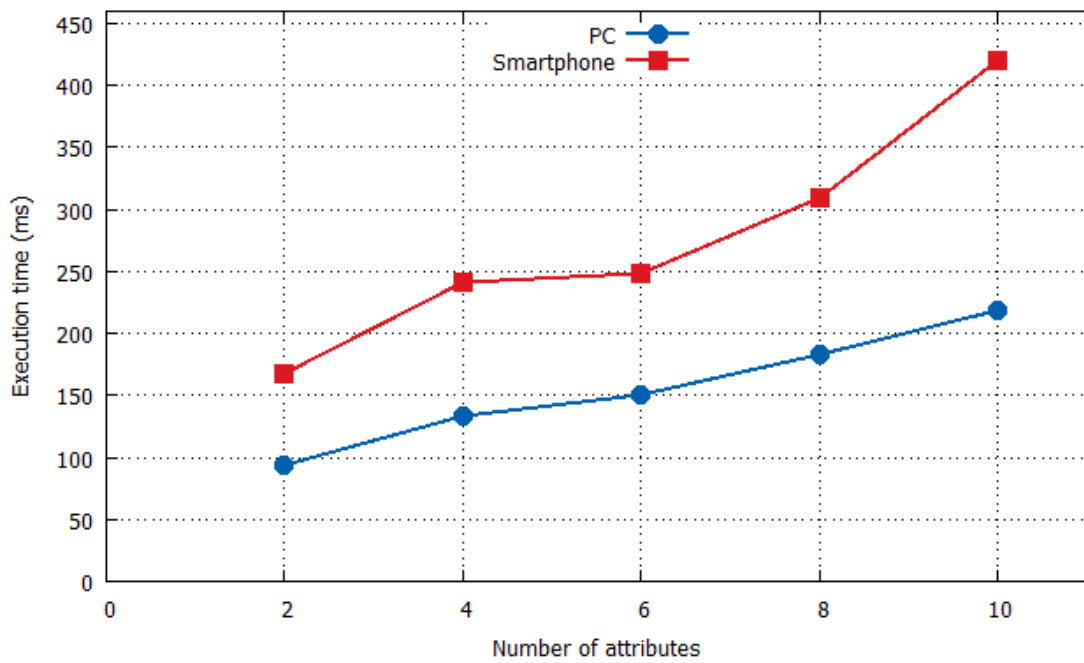


Figure 5.5: Computational time required to process the identity attributes.

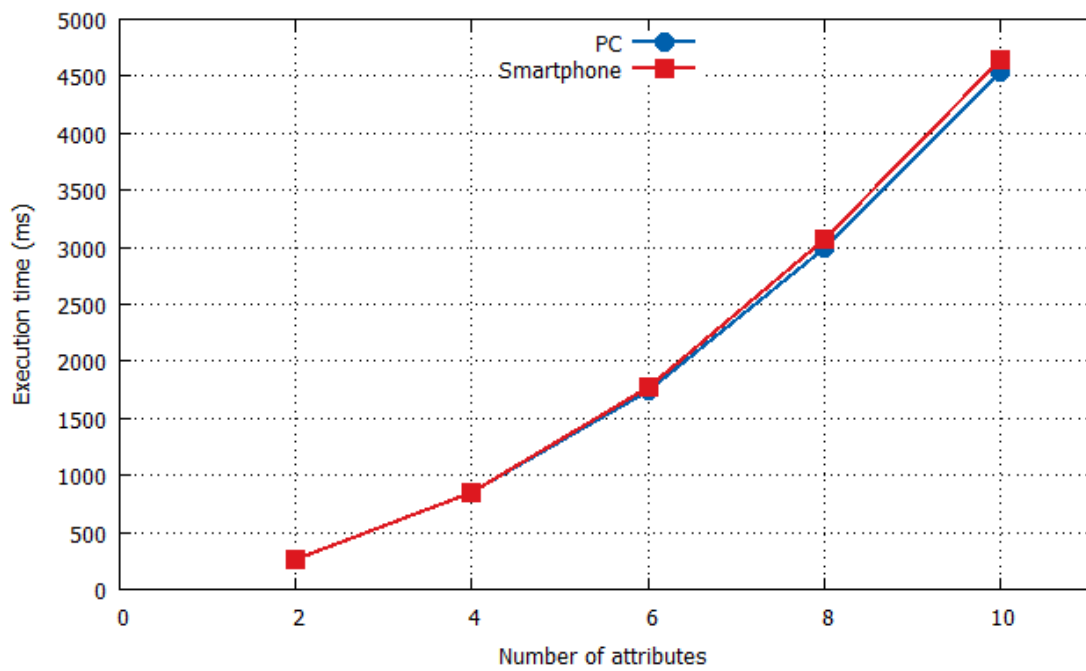


Figure 5.6: Computational time required to recovery mandatory values.

to ensure privacy of the identity attributes and also prepare these attributes for delegated private matching. Client component first process the identity attributes and then extract the mandatory values from the response send by the cloud server. Attribute processing comprises of a hashing function for encoding the identity attributes, a function to define a polynomial for the encoded attributes and a homomorphic encryption function which ensures privacy of the defined polynomial (co-efficients). For the extraction of mandatory values client component first need to homomorphically decrypt the co-efficients send by the cloud sever. Then the decrypted co-efficients are further decrypted by using a symmetric encryption algorithm to reveal the mandatory values.

5.9 Discussion

Access control policy evaluates access privileges of an entity and grants or denies access to the required resources (i.e., data, services, and business logic). As it contains information about the valid access parameters it must be evaluated by a trusted entity (i.e., policy evaluator). An untrusted or malicious entity can exploit it to gain unauthorized access to the resources. To achieve data privacy in an untrusted domain (i.e., cloud storage), existing systems either rely on a trusted third party or data owner to manage and distribute appropriate decryption keys to the authorized users. These methodologies tend to decrease utility of cloud storage by restricting data owner to stay always online and exerting computational load on the access control policy evaluator.

In order to achieve data privacy without engaging trusted third party or data owner itself we proposed oblivious access control policy evaluation in cloud storage called O-ACE. Through O-ACE cloud service provider can grants or denies access to the outsourced resources without learning any useful information about the access control policy. We extended the notion of private matching to an untrusted domain called Delegated Private Matching (DPM). Privacy of the identity attributes and obliviousness of the access control policy is achieved by DPM. Through DPM authorized user can learn the values that derive the valid decryption key; whereas, unauthorized learns the random value which does not reveal any information about the access control policy.

For the security analysis of O-ACE, we analyzed the risk of privacy breach when entities (i.e., cloud server provider, user, and identity provider) behave maliciously. Even if malicious entities team up with each other confidentiality of the outsourced data is still preserved. To demonstrate

the security viability of O-ACE we analyzed the situation in which all of the entities behaved maliciously. Even in the worst case the computational complexity for the attackers was equivalent to reverting asymmetric encryption without valid key pair.

So far Attribute Based Encryption (ABE) [93] has leveraged the data sharing systems to achieve access control policy evaluation on the client side. However, ABE is 100 ~ 1000 times slower than RSA [94]. Current implementation of ABE uses Bison and YACC parsing packages to extract access control policy from the cipher text, thus it is difficult to realize it for the mobile devices. Systems based on ABE have to generate secret key for the legitimate users, exerting computational load on the data owner. In O-ACE data owner just needs to disseminate random seed to the legitimate users. Apart from that, to revoke a user in system using ABE, data owner needs to update the access control policy so that revoked user cannot conform to it. Whereas, in O-ACE data owner only needs to update the random seed (\tilde{r}) and access of the revoked user can be restrained as it cannot learn the mandatory values without the random seed (\tilde{r}).

To highlight the practicality of O-ACE we realized a cloud-based data sharing system which assists data owner to achieve fine-grained access control over the outsourced data. However, the applicability of O-ACE is not restricted to data sharing systems only. It can be used to provision services whose response can only be decrypted by the authorized users. Similarly it can also be used to provide access to compute and storage resources (i.e., password protected virtual machines and databases). With O-ACE there is no need to engage multiple administrative entities (e.g., Key Manager, Policy Manager) to restrain illicit data access. Cloud service provider obviously performs the task of Key Manager by storing and distributing the mandatory values, and also takes over the responsibility of Policy Manager by obviously matching the identity attributes with valid access parameters.

With O-ACE we have achieved fine-grained access control over the outsourced data. However, so far O-ACE does not support logical conditions for access control policy evaluation like ABE does. Support for the logical conditions can be added by encoding the range values, and distributing the appropriate mandatory values during policy evaluations. Although it will increase the number of legitimacy and mandatory values cloud service provider has to maintain; however, it will not affect security of O-ACE. The current implementation of O-ACE assumes that identity

attributes assigned to a user do not contain any private information and can be asserted to the data owner. To avoid disclosing any private information, identity provider can assert the masked identity assertions to the data owner, and disseminates the mask to the respective user.

Through the experiments we have shown that O-ACE can be realized for the devices having limited computational capabilities. Also it exerts reasonable computational load on the cloud service provider, casting around $0.01 \sim 0.30$ dollars per 1000 requests. Additionally, it uses standard cryptographic primitives which are natively provided by the most programming languages. Thus, it can be effortlessly realized according to the security requirements and computational capabilities of the involved entities.

5.10 Summary

In this chapter, Oblivious Access Control Policy Evaluation O-ACE is presented - a method for access control policy evaluation in an untrusted domain. It enables owner to delegate the task of access control policy evaluation to an untrusted policy evaluator by utilizing Delegated Private Matching (DPM). O-ACE ensures that policy evaluator cannot learn any useful information from the procedure of access control policy evaluation, that can compromise privacy of the outsourced data and user seeking access to it.

O-ACE utilizes amalgam of cryptographic primitives and protocols to ensure that access control policy governing access to the outsourced data and access attributes of a user can be evaluated in encrypted form. The evaluation result either grants or denies access to a user. Even though access control policy and access parameters are evaluated in encrypted form still policy evaluator manages to govern data access without the need to decrypt them. DPM along with homomorphic encryption ensures the entire process of access control policy evaluation appears oblivious to a policy evaluator.

In this chapter, O-ACE is presented in the context of cloud-based data sharing service. O-ACE leverage owner to share confidential data with multiple users. Encrypted data along with encrypted access control policy is outsourced to a cloud server. O-ACE enables cloud server to evaluate access control policy obliviously, and make authorized users to learn secret values. These values assist authorized users to derive valid decryption key that can decipher the outsourced data. For

malicious users the evaluation result of O-ACE is randomized, and never reveals any information whatsoever to any of the involved entity with malicious intent.

6.1 Introduction

This chapter presents Oblivious Term Matching (OTM) - a methodology to search encrypted data, outsourced to an untrusted domain. OTM provides privacy-aware data search by enabling authorized users to execute search queries according to their access privileges. Users submit their search queries in encrypted form to prevent query evaluator from compromising privacy of encrypted data, by analyzing the query evaluation procedure. Query evaluator cannot deduce confidential information about the outsourced data or data owner, by merely learning the presence or absence of particular keywords. OTM utilizes private matching protocol to ensures that the result of query evaluation appears oblivious to a query evaluator.

OTM is designed to provide data searching capability to existing storage facilities provisioned by untrusted storage provider. Since, OTM processes the search queries in encrypted form, risk of potential privacy breach is eliminated as storage provider cannot exploit the procedure of query evaluation. OTM ensures that users can only search encrypted data on which access is permitted by the data owner. Execution of unauthorized search query does not help malicious users, even if they team up with the storage provider. In the following we present the conceptual and security model of OTM, along with its design goals. Application scenario in which OTM can provide privacy-aware searching capability is also discussed, followed by its technical detail.

6.2 Models, Design Goals, and Assumptions

The proposed methodology to search encrypted data is designed to provision privacy-aware searching capability within the untrusted domain of a storage provider. In the subsequent sections

we discuss the conceptual model of OTM. We also present the security model to delineate the threats faced by searching encrypted data considering the malicious intents of a storage provider and unauthorized users. System design goals that provide functional guidelines for the design of OTM are also discussed.

6.2.1 Conceptual Model

To realize a cloud storage with privacy-aware searching capabilities over outsourced data: cloud service provider, data owner, data consumer, and trusted third party are considered as the involved entities. For brevity these entities are referred as cloud server, owner, user, and third party respectively. Cloud server provisions storage and compute facilities on subscription basis. Owner owns the confidential data that need to be shared with users. Users can access and search the cloud storage with respect to their access privileges. Third party transforms the search criteria submitted by a user to an oblivious search query. Cloud server evaluates the oblivious queries without learning any information about the search criteria and the outsourced data. Besides this, cloud server cannot identify the files, which satisfy the search criteria specified by a user.

6.2.2 Security Model

In cloud storage services, search ensures that only desired data contents can be accessed. Since, cloud server is an untrusted entity often encrypted data is outsourced to cloud storage services. Although encryption ensure data privacy; however, cloud sever and malicious users can exploit search queries to compromise privacy of the outsourced data.

Cloud server can learn data contents which contains similar keywords, and can formulate attack strategy accordingly. For example, outsourced data related to financial statements is searched more frequently at the end of fiscal year. Cloud server can identity files which showed up in search results most frequently, and then try to deduce information regarding company financial situation i.e., if files related to legal liabilities are more frequently searched as compared to revenue statement, cloud server can reasonably infer that company is most likely to report reduced earning in current fiscal year.

Unauthorized users can submit trapdoors for particular keywords and eventually can compro-

mise privacy of the outsourced data. For example, financial records of a company, stored in a cloud storage services, are accessible to owner and employee from accounts department. An authorized user manages to learn trapdoor of a keyword *layoff*. Since, cloud server is an untrusted entity it can assist unauthorized user to search cloud storage to predict company's layoff strategy, and thus act accordingly in legal matters.

To prevent cloud server and malicious users from exploiting search over encrypted data, search query must possess the property of obliviousness and privacy-awareness. Obliviousness ensures that cloud server cannot learn result of query evaluation, consequently cannot identify data contents that fulfill the search criteria. Privacy-awareness guarantees that search queries are evaluated according to access privileges of a user, even if unauthorized users team up with cloud server, they cannot deduce confidential information about the outsourced data.

Definition: Obliviousness, and Privacy-awareness

6.2.3 System Design Goals

The proposed methodology to search encrypted data provisions data searching capabilities within untrusted domain of a cloud server with privacy consideration. Search queries can be exploited by cloud server, these queries must be evaluated obliviously to prevent potential lost of data privacy. Since, cloud server is not an trustable entity, it can also assist unauthorized users to deduce confidential information by merely identifying the presence or absence of particular keywords. Thus, search queries must be executing according to user's access privileges. While ensuring privacy of data, searching methodology must no abate efficacy of cloud storage service, and ability of authorized users to search required data contents independently.

OTM is designed to efficiently and independently search encrypted data without losing data privacy within cloud storage system. Within the context of data privacy OTM:

- thwarts cloud server to determine result of query evaluation
- restrains cloud server from learning search criteria and relating search queries submitted by different users
- prevents unauthorized users to search cloud storage and deduce any information whatsoever

whereas, with respect to efficacy of data search for cloud storage OTM is expected to:

- leverages owner to associate arbitrary number of keywords with the encrypted data, without exchanging trapdoors with users
- facilitates users to define their search queries without the need of auxiliary information provided by the owner

6.3 Application scenario and abstract idea for searching encrypted data in cloud storage

In this section we briefly illustrate application scenario in which privacy-aware data searching is utilized to provision data searching capability to authorized users in a cloud storage service. An abstract idea of OTM is also described, which will be elaborated with its technical details in the subsequent section.

Suppose, Datamine is an advisory firm which provides market analysis and trend discovery services to its customers. Alice is a director of research at Datamine. She is working on two distinct projects. One of the project mines social networks' data to devise effective advertisement campaigns. Whereas, the other project processes the healthcare data to identify early signs of an epidemic. Alice's clients exchange their data in text files, which she stores on cloud storage, owned by Eve. For each project, Alice maintains a separate directory, which contains the data along with the index (i.e., keywords). Alice does not trust Eve as the data stored on the cloud storage contains confidential information, which Eve can exploit. To ensure privacy of the data, Alice first encrypts the data and corresponding index, and then outsources them to the cloud storage.

Bob and Mallory are research analysts at Datamine, working with Alice. Bob is an expert in dealing with data related to social networks. Mallory is good in processing healthcare data. Alice has granted access to both of the research analysts on their respective directories by exchanging data decryption and secret key to conceal search criteria. Whenever Bob and Mallory need to search for a file containing particular keywords, they define selection criteria by encoding keywords with publically known encoding function. Selection criteria are then encrypted with secret key and submitted to the trusted party, which models oblivious query. Oblivious query is then sub-

mitted to the cloud server as a search query. Cloud server then obviously evaluates the query on encrypted index, and replies back the response. Trusted party processes the cloud server response and sorts the results according to the concealed selection criteria. Query execution result is then sent back to the respective user.

At Datamine, Alice is dealing with confidential data. She does not allow Bob to query directory, which persists the healthcare data. Similarly, Mallory cannot query the social network data. Even if one of them behaves maliciously and teams up with Eve, still they will not be able to successfully query the encrypted index. For an attacker result of a malicious query is always a randomized response. Oblivious evaluation of the query ensures that Eve cannot learn the keywords, which Bob and Mallory are looking for. Nevertheless, Eve manages to accurately evaluate the query, without compromising privacy of the query and outsourced data. Figure 6.1 presents an abstract model of our proposed oblivious search for cloud storage.

6.4 Searching cloud storage with oblivious term matching (OTM)

Oblivious query evaluation in a cloud storage system is achieved by uniquely combining homomorphic encryption and proxy re-encryption. The amalgam of these cryptographic primitives ensures that cloud server cannot learn any information about the outsourced data. Most importantly, cloud server evaluates the query submitted by a user, without learning the search criteria and the result of query execution. Besides this, it also ensures that unauthorized users cannot query the outsourced data on which access is not granted by the owner. Table 6.1 illustrates the notations that we use to explain the core concept of our proposed oblivious search for cloud storage.

To achieve privacy-aware data search in an untrusted domain of cloud server, the proposed privacy-aware search for cloud storage is divided into five steps namely: setup, data outsourcing, and query generation, searching and response extraction.

6.4.1 Setup

Privacy-aware data search in a cloud storage system is achieved by searching an inverted index (\mathcal{I}) associated with the outsourced data (\mathcal{F}). For each \mathcal{F} , the owner generates \mathcal{I} by utilizing an

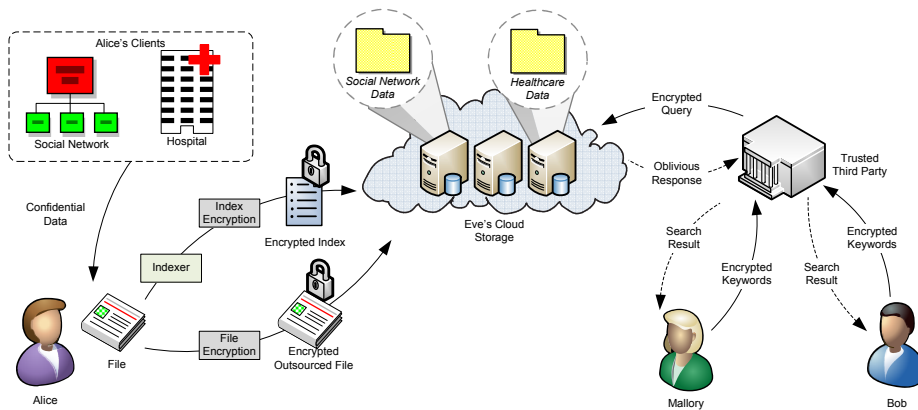


Figure 6.1: Abstract model of privacy aware oblivious data search in a cloud storage.

Table 6.1: Notations used in the descriptive detail of OTM.

<i>Notation</i>	<i>Description</i>
\mathcal{F}	Confidential file that need to be shared.
$\mathcal{I} = \{kw_0 \dots, kw_n\}$	Index file that contains n keywords.
\mathcal{H}	Public encoding function which encodes an arbitrary string to an integer value of q modulo; where q is a large prime.
$\mathcal{E}_P, \mathcal{D}_P, \mathcal{T}_P$	Proxy Re-encryption, decryption, and ciphertext transformation algorithms.
$\omega_o, \omega_u, \omega_{o \rightarrow u}$	Proxy Re-encryption keys for the owner, user and cloud server respectively.
$\mathcal{E}_H, \mathcal{D}_H$	Homomorphic encryption and decryption algorithms.
σ_{pk}, σ_{sk}	Public and secret keys for homomorphic encryption algorithm.
$\mathcal{E}_A, \mathcal{D}_A$	Asymmetric encryption and decryption algorithms.
k_{pub}, k_{pri}	Public and private key pair for asymmetric encryption algorithm.
$\alpha_{0 \dots n}$	List of coefficients of a polynomial P that defines the search query.
$\Delta_{y_{0 \dots n}}$	Oblivious values: query execution result by a cloud server provider.
$\psi_{0 \dots n}$	Query execution result computed by the user from the oblivious values ($\Delta_{y_{0 \dots n}}$).
sk	Third party secret to conceal keyword frequency in inverted index.

indexing algorithm. \mathcal{I} contains a list of keywords, along with the frequency of each keyword ($\mathcal{I} = (kw_0, f_0), \dots, (kw_n, f_n)$). Search queries are evaluated against these keywords. Once, \mathcal{I} is generated, the owner initializes proxy re-encryption by generating owner key (ω_o), user key (ω_u), and transformation key ($\omega_{o \rightarrow u}$). ω_o ensures the privacy of keywords within \mathcal{I} , whereas keyword frequencies are concealed with third party's secret key (sk). ω_u is used by the user to encrypt search criteria. The owner only shares ω_u with the authorized users. $\omega_{o \rightarrow u}$ is used by the cloud server to transform ciphertext (encrypted inverted index). Transformation of ciphertext ensures that the owner does not need to outsource separate encrypted index for each authorized user.

6.4.2 Data Outsourcing

To ensure that cloud server can obviously evaluates the search query submitted by an authorized user, owner encodes $\mathcal{I}_{kw_0 \dots n}$ by using a publicly known encoding function i.e., $\mathcal{H}(\mathcal{I}_{kw_0 \dots n}) \rightarrow \hat{\mathcal{I}}_{kw_0 \dots n}$. The encoded keywords ($\hat{\mathcal{I}}_{kw_0 \dots n}$) are then encrypted with proxy re-encryption algorithm by using ω_o i.e., $\mathcal{E}_P(\hat{\mathcal{I}}_{kw_0 \dots n}, \omega_o) \rightarrow \hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$. To ensure that the cloud server cannot learn any information from the inverted index owner encrypts $\mathcal{I}_{f_0 \dots n}$ with third party's secret key i.e., $\mathcal{E}_S(\mathcal{I}_{f_0 \dots n}, sk) \rightarrow \mathcal{I}_{f_0 \dots n}^{sk}$. After that, the owner encrypts ω_u with the public key of the user to whom it wants to grant searching capabilities over the outsourced data i.e., $\mathcal{E}_A(\omega_u, k_{pub}) \rightarrow \omega_u^{k_{pub}}$.

In a cloud storage system, outsourced data can be shared with multiple users - each having its own access privileges over the outsourced data. With proxy re-encryption owner does not need to encrypt $\mathcal{I}_{kw_0 \dots n}$ separately to permit each authorized user to query $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$. An authorized user can submit its query encrypted with its proxy re-encryption secret key (ω_{u_i}). Cloud server then transforms $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$ to $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_{u_i}}$ by using an appropriate transformation key ($\omega_{o \rightarrow u_i}$) provided by the owner. Thus, owner only needs to encrypt $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$ once, and n authorized users can query it, without compromising privacy of the outsourced data.

Once, the owner secures $\mathcal{I}_{kw_0 \dots n}$, $\mathcal{I}_{f_0 \dots n}$, and ω_u , it outsources $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$, $\mathcal{I}_{f_0 \dots n}^{sk}$ and $\omega_u^{k_{pub}}$ to the cloud server, along with the outsourced data. After that, availability of the owner is no longer required. Multiple users can engage in an oblivious query evaluation protocol with the cloud server. However, only authorized users can successfully query $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$. For others, the cloud server obviously generates a randomized response from which they cannot learn any information

about the outsourced data.

6.4.3 Query Generation

In order to privately search the cloud storage, user obtains its proxy re-encryption secret key from the cloud server and decipher it by using the private key i.e., $\mathcal{D}_A(\omega_u^{pub}, k_{pri}) = \omega_u$. The user then defines a search criteria ($\mathcal{C}_{kw_0 \dots l}$), that consist of a list of keywords $kw_0 \dots kw_l$. $\mathcal{C}_{kw_0 \dots l}$ is then encoded by using a publicly known encoding function i.e., $\mathcal{H}(\mathcal{C}_{kw_0 \dots l}) \rightarrow \hat{\mathcal{C}}_{kw_0 \dots l}$, where \mathcal{H} is same as used by the owner during data outsourcing. To ensure confidentiality of the keywords, $\hat{\mathcal{C}}_{kw_0 \dots l}$ is encrypted with proxy re-encryption by using the proxy re-encryption secret key i.e., $\mathcal{E}_P(\hat{\mathcal{C}}_{kw_0 \dots l}, \omega_u) = \hat{\mathcal{C}}_{kw_0 \dots l}^{\omega_u}$.

Once privacy of the search criteria is assured it is send to the third party who uses it to model oblivious search query. On receiving $\hat{\mathcal{C}}_{kw_0 \dots l}^{\omega_u}$ third party defines a polynomial ($P(x)$), such that each element of $\hat{\mathcal{C}}_{kw_0 \dots l}^{\omega_u}$ is a root of $P(x)$ i.e., $P(x \in \hat{\mathcal{C}}_{kw_0 \dots l}^{\omega_u}) = \sum_{i=0}^l \alpha_i x^i = 0$, see Section 3.2 for more details on defining a polynomial with multiple roots.

Once $P(x)$ is defined in accordance to $\hat{\mathcal{C}}_{kw_0 \dots l}^{\omega_u}$, third party then initializes homomorphic encryption by generating a public key (σ_{pk}) and secret key (σ_{sk}). The third party then encrypts the coefficients ($\alpha_{0 \dots l}$) of $P(x)$ with homomorphic encryption algorithm by using σ_{sk} i.e., $\mathcal{E}_H(\alpha_{0 \dots l}, \sigma_{sk}) = \alpha_{0 \dots l}^{\sigma_{sk}}$. After that, $\alpha_{0 \dots l}^{\sigma_{sk}}$ and σ_{pk} are transfered to the cloud server. Encrypted coefficients ($\alpha_{0 \dots l}^{\sigma_{sk}}$) are used to execute search query over encrypted inverted index ($\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$). Section 3.2 illustrates that coefficients ($\alpha_{0 \dots n}$) of a polynomial ($P(x)$) can be used to compute set intersection between two private sets. In the context of search over encrypted data, set intersection can be used to execute search query by matching search criteria with the inverted index.

6.4.4 Searching

Cloud server hosts the encrypted inverted index as an encrypted keywords ($\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$) and their concealed frequencies ($\mathcal{I}_{f_0 \dots n}^{sk}$) along with the outsourced data (\mathcal{F}). Encrypted query ($\alpha_{0 \dots l}^{\sigma_{sk}}$) submitted by the third party is evaluated against $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$. On receiving $\alpha_{0 \dots l}^{\sigma_{sk}}$ cloud server transforms $\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}$ with the respective user's transformation key ($\omega_{o \rightarrow u}$), provided by the owner i.e., $\mathcal{T}_P(\hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_o}, \omega_{o \rightarrow u}) \rightarrow \hat{\mathcal{I}}_{kw_0 \dots n}^{\omega_u}$. Once, the encrypted index is transformed, cloud server defines

a polynomial ($P(y)$), by using each element of $\alpha_{0..l}^{\sigma_{sk}}$ as a coefficient of $P(y)$. It then compute oblivious value (Δ_{y_i}), by evaluating $r.P(y_i)$, where $y_i \in \hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u}$ and r is a random number, i.e., $\Delta_{y_i} = r.P(y_i)$.

As the query is concealed by using homomorphic encryption, cloud server cannot learn any information from $P(y_i \in \hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u})$. Once the cloud server has evaluated $P(y_{0..n} \in \hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u}) = \Delta_{y_{0..n}}$, it replies back the query evaluation result - list of oblivious values along with the concealed keyword frequencies to the third party i.e., $\Delta_{y_{0..n}}, \mathcal{I}_{f_{0..n}}^{sk}$.

6.4.5 Response Extraction

On receiving the cloud server's response ($\Delta_{y_{0..n}}, \mathcal{I}_{f_{0..n}}^{sk}$), third party decrypts the oblivious values by using the homomorphic secret key i.e., $\mathcal{D}_H(\Delta_{y_i}, \sigma_{sk}) = \psi_i$. Where ψ_i can be zero or a random number. As the search query was modelled as a polynomial having roots equals of the concealed search criteria i.e., $P(x \in \hat{\mathcal{C}}_{kw_{0..l}}^{\omega_u}) = \sum_{i=0}^l \alpha_i x^i$, query evaluation at cloud server can result either in a zero or a non-zero value shown in equation 6.1.

$$P(y_i) = \begin{cases} \psi_i = 0 & \text{if } \{y_i | y_i \in \hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u} \wedge y_i \in \hat{\mathcal{C}}_{kw_{0..l}}^{\omega_u}\} \\ \psi_i \neq 0 & \text{if } \{y_i | y_i \in \hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u} \wedge y_i \notin \hat{\mathcal{C}}_{kw_{0..l}}^{\omega_u}\} \end{cases} \quad (6.1)$$

Zero value reveals that inverted index contain keyword that matches with the concealed search criteria specified by the user i.e., $\hat{\mathcal{C}}_{kw_i}^{\omega_u} \in \hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u}$. Whereas, non-zero reveals that concealed search criteria do not match with any of the keyword in inverted index, consequently third party recovers \tilde{r} . Once encrypted keywords are identified, third party decipher the corresponding frequency index by using the secret key i.e., $\mathcal{D}_S(\mathcal{I}_{f_i}^{sk}, sk) \rightarrow \mathcal{I}_{f_i}$. After that third party, sort the identified encrypted keywords according to the frequency count. Third party then replies back the oblivious query evaluation result to the user.

On receiving the third party response, user deciphers the search criteria by using its proxy re-encryption secret key. Through decryption user learns the keywords that matches with the encrypted index i.e., $\mathcal{D}_P(\hat{\mathcal{C}}_{kw_{0..k}}^{\omega_u}, \omega_u) = \hat{\mathcal{C}}_{kw_{0..k}}$, where k is the number terms that are identical between $\hat{\mathcal{C}}_{kw_{0..l}}$ and $\hat{\mathcal{I}}_{kw_{0..n}}^{\omega_u}$. During the query evaluation cloud server learns nothing about the

Table 6.2: OTM computational complexity and estimation of transmitted values.

OTM Steps	Operations	Input Size	Computational Complexity	Transmitted Values
Index Outsourcing	<u>Public Encoding</u>	N	$O(N)$	$(N + 3)$ values
	<u>Symmetric Encryption</u>			
	<u>Asymmetric Encryption</u>			
Query Generation	<u>Attribute Preparation</u>	n	$O(n)$	n values
	<u>Asymmetric Encryption</u>			
Query Modeling	<u>Asymmetric Encryption</u>	n	$O(n^3)$	$(n + 2)$ values
	<u>Polynomial Modeling</u>			
Query Evaluation	<u>Polynomial Evaluation</u>	$(n + 1) \cdot N$	$O(n^2 \cdot N)$	N values
Oblivious Result Processing	<u>Asymmetric Decryption</u>	N	Depends on Auxiliary Function	Matched records
	<u>Auxiliary Function</u>			
Query Results	<u>Asymmetric Decryption</u>	Depends on n		

inverted index or the search criteria. However, it accurately evaluates the search query and replies back the oblivious response. Whereas, third party only learns frequencies of the concealed keywords that matches the search criteria.

6.5 Complexity Analysis

In this section we analysis the computational complexity of oblivious data search for encrypted data within untrusted domain ¹. We also estimate the transmission of data between different entities to execute oblivious term matching. Table 6.2 shows the computational complexity and estimated transmitted values for each step of oblivious term matching. Both computational complexity and transmitted values are directly proportional to inverted index and oblivious query size.

6.5.1 Index Outsourcing

Oblivious term matching use pre-processed indexed keywords to search encrypted data. Index outsourcing prepares the inverted index for outsourcing to an untrusted domain of cloud service provider. Index outsourcing comprises of a public encoding function, and symmetric and asymmetric encryption operations. Computational complexity of index outsourcing depends on index terms generated by an indexing algorithm. With Big- O notation computational complexity of in-

¹For brevity cryptographic operations are considered to take constant time.

dex outsourcing can be expressed as $O(N)$, where N is the inverted index size. In total $N + 3$ values are transmitted to the cloud server, N size of inverted index, two proxy re-encryption keys and a symmetric encryption key for the third party to process keyword frequencies.

6.5.2 Query Generation

Since, third party is considered as trusted by curious entity, subscriber need to pre-process the search criteria to ensure privacy of the outsourced data and keywords used in search query. Query generation is comprises of public encoding and asymmetric operations. Input size of query generation is n , where n is the number of search criterion used by a subscriber. The computational complexity of query generation is $O(n)$. As subscriber only conceals the search criteria, total n values are transmitted to the third party.

6.5.3 Query Modeling

Query modeling comprises of two operations polynomial modeling and asymmetric encryption. Polynomial is modeled by using conceal search criteria as root values. Co-efficients of the polynomial are concealed by using asymmetric encryption. Since, oblivious query is modeled by using the search criteria specified by a subscriber, its input size is n . Thus computational complexity of query modeling is $O(n^3)$. In total $n + 2$ values are transmitted to the cloud server, $n + 1$ concealed co-efficients and a public key for homomorphic operation over encrypted inverted index.

6.5.4 Query Evaluation

Search over encrypted data is executed by evaluating polynomial on encrypted inverted index by using homomorphic properties of Pascal Paillier cryptosystem. Computational complexity of query evaluation process depends on two factors, size of inverted index (N) and size of oblivious query posted by a third party ($n + 1$). Thus the computational complexity of query evaluation, in terms of Big- O notation can be expressed as $O(n^2.N)$. Similarly, data transmission is directly related to index size. In total N oblivious values are transmitted to the third party, as $P(x)$ is separately evaluated for each of the index entry.

6.5.5 Oblivious Result Processing

Oblivious result processing is comprises of two operations, asymmetric decryption and post-processing of the results. Input size of this step depends on size of inverted index (N). Thus, the computational complexity of asymmetric decryption process can be defined as $O(N)$. However, the computational complexity of post-processing is subject to choice of processing that subscribers has requested i.e., sorting, mering. Whereas, number of transmitted values is a function over matched search criteria with inverted index and output of auxiliary processing function.

6.5.6 Result Extraction

Result extraction is the simplest of all steps in OTM. It only requires asymmetric decryption process over the values transmitted by the third party. Thus it computational complexity is the function of matched keywords and output of auxiliary processing at third party.

6.6 Security Analysis

This section presents the security analysis of OTM. Particularly, we focus on capabilities of malicious entities to learn encrypted search query and to deduce confidential information about the encrypted outsourced data. We examine the advantage of untrusted cloud service provider to learn the result of oblivious query evaluation and deduce information, which can lead to potential loss of privacy. We then discuss the scenario in which an unauthorized subscriber attempts to search encrypted data on which it does not have access. In the last, we examine the capability of a malicious third party to model unauthorized search queries in an attempt to learn presence or absence of a particular keyword.

OTM utilizes number of cryptographic primitives to ensure execution of encrypted search queries and to restrain malicious entities to deduce information that assists them to compromise privacy of the outsourced data. As illustrated in the descriptive detail of OTM, inverted index is encrypted with symmetric and asymmetric encryptions i.e., keywords are encrypted with asymmetric encryption (Proxy Re-Encryption) and keyword frequencies are encrypted with symmetric encryption. To ensure oblivious evaluation of search queries homomorphic encryption is utilized

along with private matching protocol. For the security analysis of these cryptographic primitives, readers may refer to [85], [86], and [90]. In the subsequent sections, we examine the capabilities of malicious entities to deduce confidential information within the context of OTM.

6.6.1 Malicious Cloud Server

The proposed methodology of encrypted data search utilizes computational power and storage facility of a cloud server to execute search queries, instead of relying on trusted third party. Cloud server persists inverted index $(\mathcal{I}_{kw_{0...n}}, \mathcal{I}_{f_{0...n}})$, that comprises of encrypted keywords $(\hat{\mathcal{I}}_{kw_{0...n}}^{\omega_o})$ and their respective frequencies $(\mathcal{I}_{f_{0...n}}^{sk})$. To compromise privacy of the outsourced data, cloud server either has to decipher inverted index or deduce information from the evaluation of encrypted search queries.

OTM avoids multiple copies of inverted index. Instead of separately outsourcing inverted index for each authorized subscriber, single copy of inverted index is outsourced, encrypted with Proxy Re-Encryption i.e., $\hat{\mathcal{I}}_{kw_{0...n}}^{\omega_o}$. For each authorized subscriber cloud server utilizes appropriate transformation key to transform encrypted keyword accordingly i.e., $\mathcal{T}_P(\hat{\mathcal{I}}_{kw_{0...n}}^{\omega_o}, \omega_o \rightarrow u_i) \rightarrow \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_{u_i}}$. Search queries are submitted in encrypted format $(\alpha_{0...l}^{\sigma_{sk}})$, and evaluated by using private matching protocol i.e., $(P(y_{0...n} \in \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_{u_i}}) = \Delta_{y_{0...n}})$. Since, search queries are encrypted and result of query evaluation is oblivious to cloud server, it cannot learn any information about the keywords that are concealed in search query.

In order to compromise privacy of the outsourced data cloud server need access to secret key of a subscriber i.e., ω_{u_i} , that conceals keywords used in encrypted search query. Once, cloud server has access to the secret key it can effortlessly decipher the keywords that comprises the inverted index. However, only authorized subscribers have access to their secret key as it is encrypted with their respective public key $(\omega_{u_i}^{k_{pub}})$. Thus, for a cloud server computational complexity to compromise privacy of the outsourced is equivalent to asymmetric encryption. However, even if it manages to gain access to secret key it can only decipher encrypted keywords associated with the outsourced data - confidentiality of the outsourced data is preserved as it is encrypted with symmetric encryption key, which is disseminated to authorized subscribers. Since, OTM deals with the encrypted data search authorized data access is beyond its scope.

6.6.2 Malicious Subscriber

OTM not only realizes oblivious data search in untrusted domain it also tackles the problem of unauthorized data search. It ensures that unauthorized subscribers are not able to deduce any information about the encrypted outsourced data by simply learning the presence or absence of keywords. OTM does provide protection against conspired attacked by unauthorized subscribers and untrusted cloud server. As discussed earlier in the descriptive detail of OTM, proxy re-encryption conceals the keywords in inverted index outsourced to a cloud server. With proxy re-encryption, we are able to maintain single inverted index for all authorized subscribers; this also ensures that only search queries from authorized subscribers can be evaluated successfully.

To search encrypted data search criteria ($\mathcal{C}_{kw_{0...n}}$) is concealed with secret key i.e., $\mathcal{E}_P(\hat{\mathcal{C}}_{kw_{0...l}}, \omega_u) = \hat{\mathcal{C}}_{kw_{0...l}}^{\omega_u}$. Once concealed it is then transmitted to third party to model oblivious search query. Since, oblivious query is evaluated against encrypted index, cloud server transforms the encrypted index by using subscriber's transformation key i.e., $\mathcal{T}_P(\hat{\mathcal{I}}_{kw_{0...n}}^{\omega_o}, \omega_{o \rightarrow u_i}) \rightarrow \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_{u_i}}$. For each authorized subscriber transformation key and secret key pair is $(\omega_{o \rightarrow u_i}, \omega_{u_i})$ is generated by the data owner. Data owner transmits the transformation key to the cloud server and hands over the secret key to authorized subscriber.

Since, only authorized subscribers have their secret keys, search queries from unauthorized subscribers cannot be evaluated successfully as concealed search criteria is only comparable with encrypted index transformed with valid transformation key - transformation and secret key must be compatible with each other i.e., $\omega_{o \rightarrow u_i}, \omega_{u_i}$. Even if unauthorized subscribers collude with cloud server, execution of unauthorized search queries cannot assist them to learn any useful information. Search criteria encrypted with arbitrary secret key is not compatible with concealed inverted index i.e., $\hat{\mathcal{C}}_{kw_{0...l}}^{\omega_o} \notin \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_{u_i}}$. Thus, for unauthorized subscribers it is computationally infeasible to deduce any information that can lead to potential loss of data privacy - proxy re-encryption is an asymmetric encryption based on bilinear groups.

6.6.3 Malicious Third Party

In OTM, third party assists authorized subscribers to search encrypted data. Its core purpose is to model oblivious search queries and to sort the result of oblivious query evaluation according

to keyword frequencies. Authorized subscribers can themselves model oblivious queries and sort search result; however third party is merely included to delegate computational intensive task to an entity having reasonable pool of computational resources. Nevertheless, if query evaluation results are not needed to be in specific order third party can be completely avoided.

OTM consider third party as a trustable entity. However, OTM does not leverage third party with any information that can be used to compromise privacy of the outsourced data. Security analysis discussed previously is also applicable for third party as well. To learn presence or absence of an arbitrary keyword, secret key must be known to third party, also cloud server must possess the respective transformation key; $\omega_{o \rightarrow u_i}, \omega_{u_i}$ must be a valid key pair. Thus to compromise privacy of the outsourced data, secret key is required such that concealed search criteria is comparable with the encrypted index persisted by the cloud server i.e., $\hat{\mathcal{C}}_{kw_{0...l}}^{\omega_{u_i}} \in \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_{u_i}}$. Since, authorized subscribers have access to their secret key only they can conceal arbitrary search criteria i.e., $(\mathcal{E}_P(\hat{\mathcal{C}}_{kw_{0...l}}, \omega_u) = \hat{\mathcal{C}}_{kw_{0...l}}^{\omega_u})$ such that private matching protocol can be evaluated successful.

Since, OTM consider third party as a trusted entity, it has access to keyword frequency $\mathcal{I}_{f_{0...n}}$. By using keyword frequency, third party can identified the group of subscribers that are searching for similar keywords. Nevertheless, authorized subscribers conceal their search criteria by using secret key to restrain third party from learning actual keywords that corresponds to deciphered keyword frequencies. In case of oblivious query evaluation, third party can only learn presence or absence of particular concealed keywords i.e., $\hat{\mathcal{C}}_{kw_{0...l}}^{\omega_{u_i}} \in \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_{u_i}}$; however it cannot deduce any information more than keyword frequency count.

6.7 Implementation

Application scenario discussed in Section 6.3 is realized for Google's cloud ecosystem, data search, key management, and intermediate services are implemented as standard Java web services. Data search and key management services are deployed on Google App Engine. Intermediate service is deployed on secure local server. Google Docs hosts the outsourced documents. Google Datastore is utilized to store encrypted inverted index associated with the documents stored in Google Docs. Data search service is responsible for executing search queries over the encrypted inverted index. For each user, data owner outsources user's proxy re-encryption key to key man-

agement service; whereas, each user generates its own RSA key pair. Public key is persisted by the key management service; and the user securely stores its private key. Intermediate service is utilized to model oblivious query and process the response of data search service.

To create inverted index Apache Lucene [95] is utilized - a high-performance, full-featured text search engine library. With Apache Lucene arbitrary number of keywords are associated with the outsourced documents. Although, there is no restriction on the length of inverted index, however Lucene is restricted from indexing keywords that are smaller than a four characters. Keywords can be manually added or removed from the inverted index. SHA-512 hashing algorithm is utilized to hash keywords in inverted index. Hashed value of individual keyword is encoded as a BigInteger of arbitrary length. To achieve oblivious query evaluation, Pascal Paillier cryptosystem is utilized.

Owner and client applications are deployed as standard Java SE 7.0 desktop applications. The owner application is responsible for generating inverted index, encrypting it with proxy re-encryption and outsourcing it to Google Datastore. The client application is utilized to encode search criteria and encrypting it with proxy re-encryption.

6.8 Evaluation

We evaluate our proposed privacy-aware search for cloud storage on Google's cloud ecosystem (i.e., Google App Engine [96] and Google Datastore [97]). Data search and Key management services are individually deployed on Google App Engine by using F4 frontend instance class having 2.40 GHz processor and 512 MB main memory [98]. The performance analysis of owner and client applications are carried out on 32-bit Windows 7 machine having 2.60 GHz Dual Core processor with 2 GB main memory. We test execution overhead of intermediate service on 64-bit Windows 7 machine having 3.30 GHz Core i5 processor with 4 GB main memory.

For evaluation, initially we analyse owner, and client applications by measuring the execution time required to generate encrypted inverted index, encrypt search criteria, and decrypt response of intermediate service. We then present execution overhead to model and generate oblivious query, and time required to learn result of oblivious query evaluation. Finally, we present the execution time and cost analysis of oblivious query evaluation on Google App Engine. The core purpose of this evaluation is to measure the execution overhead of enabling oblivious query evaluation on

cloud storage and modelling of privacy-aware search query. We intentionally neglect the details of key exchange between the key management service and authorized subscribers.

6.8.1 Inverted Index and Search Criteria Generation and Processing

We utilize Apache Lucene to generate inverted index. Indexed keywords are hashed by using SHA-512 hashing algorithm. Individual hashed value is then encoded as a BigInteger of arbitrary length. Finally, the encoded values are encrypted with proxy re-encryption by using 1248 bit key and frequencies of indexed keywords are encrypted with AES by using 256 bit key. Figure 6.2 shows the time required to index file of varied sizes ranging from 1 to 30 MB. It delineates the execution time to encrypt indexed terms within inverted index. Figure 6.3 presents the time exerted by client application in generating encrypted search criteria comprising of 2 to 14 keywords and decrypting the response of third party.

The evaluation of owner application reveals that the time required to generate inverted index by using Apache Lucene is linear to file size. Besides this, execution time to conceal indexed terms with proxy re-encryption by using 1248 bit key also shows linear behaviour. Although, inverted index can be generated and encrypted in linear time by the owner application; however, size of inverted index (i.e., number of indexed terms) effects the computational time and cost required to evaluate obviously queries on Google App Engine. Data owner should only select those indexed terms that are relevant to a document. For client application, we utilized 2 to 14 different keywords to define search criteria. Our evaluation results highlight the fact that client application can conceal search criteria with proxy re-encryption within fairly response time, considering the level of secrecy achieved with 1248 bit key. Besides this, the decryption of intermediate service's response shows linear behaviour with respect to the number of keywords that comprises the search criteria.

6.8.2 Oblivious Query Modelling and Response Extraction

To model oblivious query a polynomial is defined by the third party, such that the concealed keywords constituting the search criteria are roots of that polynomial. We call this process as query modelling. After that, the third party initialize a 1248 bit key pair of Pascal Paillier crypto

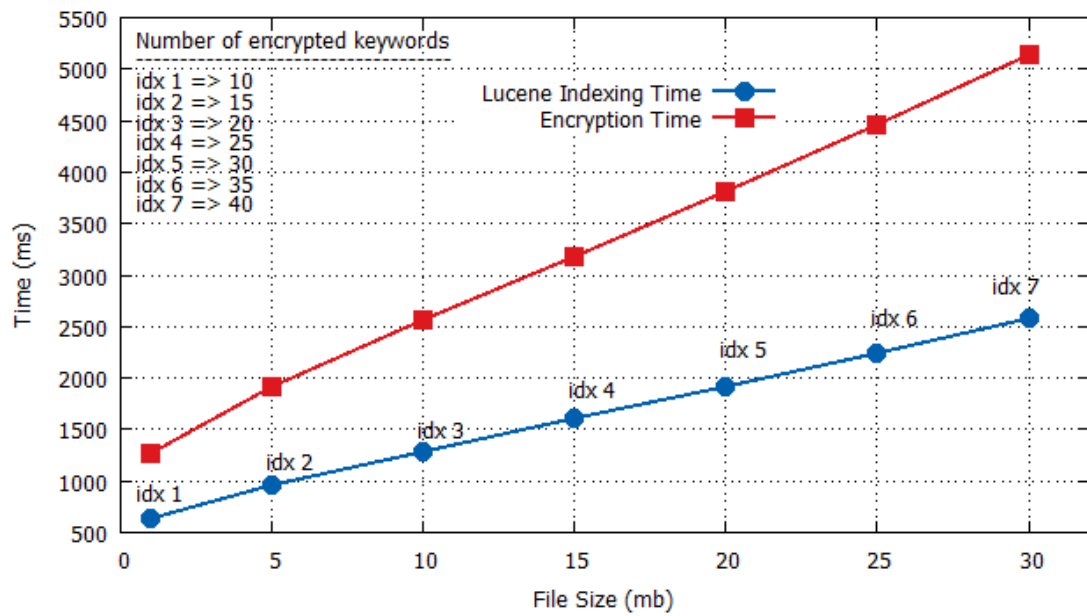


Figure 6.2: Inverted index generation and indexed term encryption time.

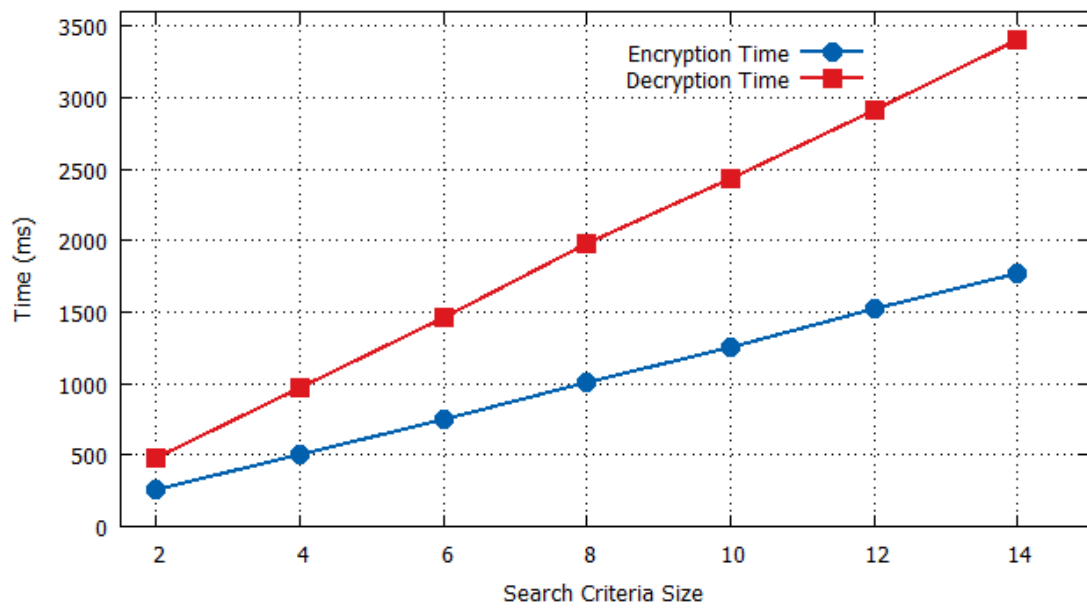


Figure 6.3: Search criteria encryption and decryption time.

system and encrypt each coefficient of the defined polynomial. We refer this process as query encryption. The encrypted query is then transmitted to the cloud storage along with the public key of Pascal Paillier crypto system.

Data search service obviously searches the cloud storage, and responds back the query evaluation result. Third party then learns the keywords that match with the encrypted inverted index by deciphering the query evaluation result. Matched keywords are sorted by the third party and sent back to the user. We refer the process of learning matched keywords as response extraction. Figure 6.4 presents the execution time of oblivious query modelling, query encryption, and response extraction.

Search query comprises of higher number of keywords can be effortlessly modelled by the user. However, execution time of query encryption linearly increases with the increase in size of search query. It applies for the response extraction, as well. Although number of terms effect the encryption and response extraction time, still it remains fairly amicable and never increases form 2901 and 4627 milliseconds respectively for query having 14 distinct keywords comprising the search criteria.

6.8.3 Oblivious Query Evaluation on Google App Engine

We measure the execution time and cost of oblivious query evaluation on Google App Engine. For the performance analysis we consider the execution time of a billable CPU (CPU Time), time taken to complete the request (response time) and estimated CPU usage cost for 1000 identical requests (cpm_usd). To obviously evaluate the search queries, data search service takes encrypted query from the third party. It then execute the oblivious search query for each value in the encrypted inverted index. The result of oblivious query evaluation is then sent back of the third party. Figure 6.5 shows computational and cost analysis of oblivious query evaluation on Google App Engine for a single encrypted inverted index entry.

It is clear that execution time and cost greatly depends on number of keywords comprising the selection criteria. However, the performance analysis has shown a linear relation among the size of query, CPU Time and its estimated cost. Through the predictive cost analysis, we have shown that the estimated execution cost of oblivious query evaluation having 2 to 14 keywords

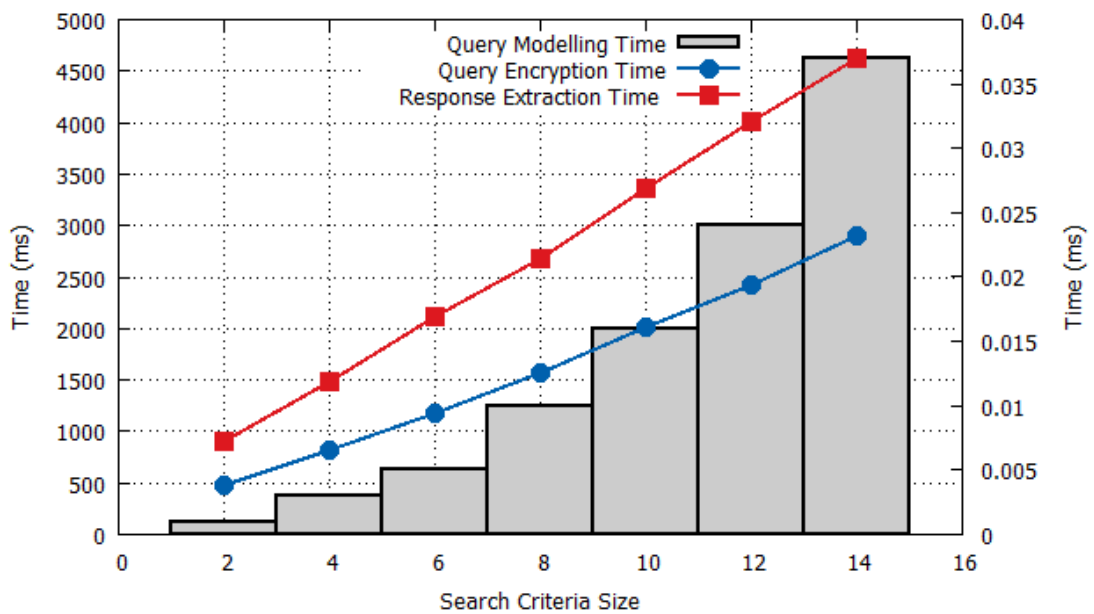


Figure 6.4: Query modelling, oblivious query generation encryption and response extraction time.

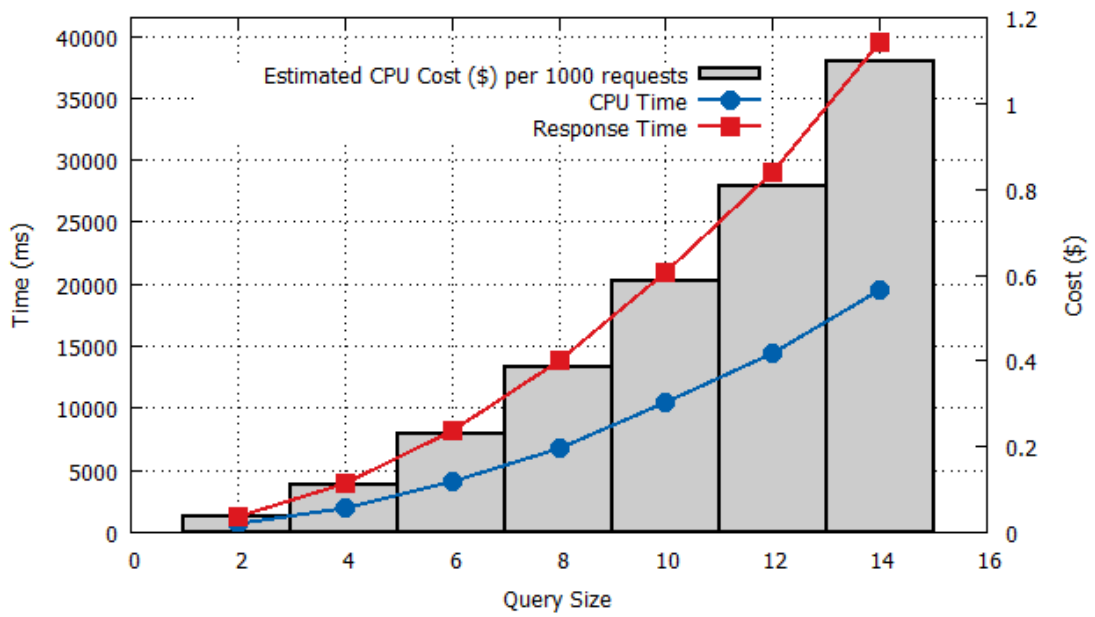


Figure 6.5: Oblivious query evaluation time, cloud server response time and estimated execution cost for 1000 requests.

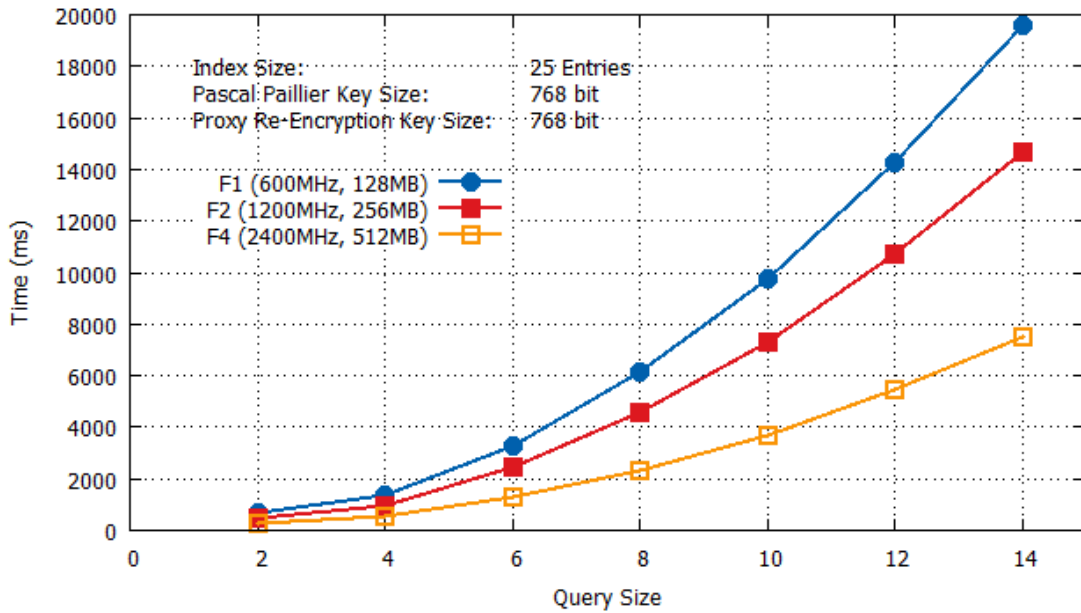


Figure 6.6: Computation time (ms) required to evaluate oblivious term matching on F1, F2 and F4 Frontend instances of Google App Engine.

remains between 0.035 to 1.09 dollars for 1000 requests of similar computational requirements. We have highlighted the fact that oblivious data search can be realized by a cloud storage system with amicable computational load and fairly reasonable cost, without compromising privacy of the outsourced data and search queries as well.

Google App Engine provides three different configurations of computer machines, called frontend classes. Frontend class instances have compute power of 600, 1200, and 2400 MHz, equipped with 128, 256 and 512 MB main memory. We evaluated oblivious term matching on these Frontend classes. 25 index entries were used to evaluated different query sizes ranging from 2 to 14 keywords. Index entries were encrypted with 768 bit proxy re-encryption key size, whereas oblivious queries were encrypted with 768 bit key size Pascal Paillier encryption. Figure 6.6 and 6.7, show the computation time and response time of oblivious term matching on App Engine. Figure 6.8 shows the CPU cycles estimated by App Engine to evaluate oblivious term matching. CPU cycles in all three Frontend instances (F1, F2, and F4) remain same.

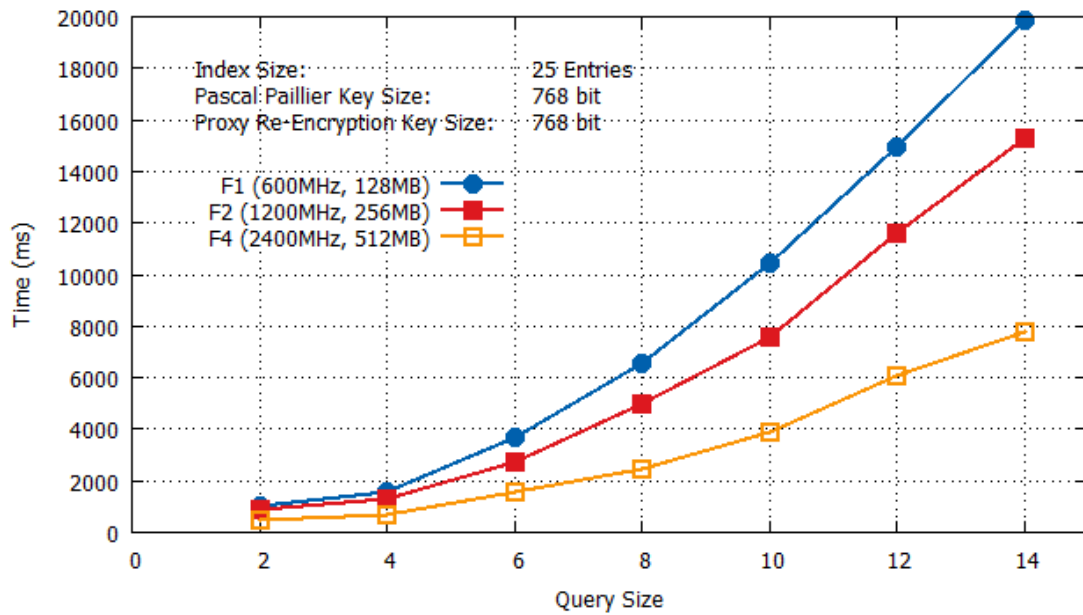


Figure 6.7: Response time (ms) of oblivious term matching on F1, F2 and F4 Frontend instances of Google App Engine.

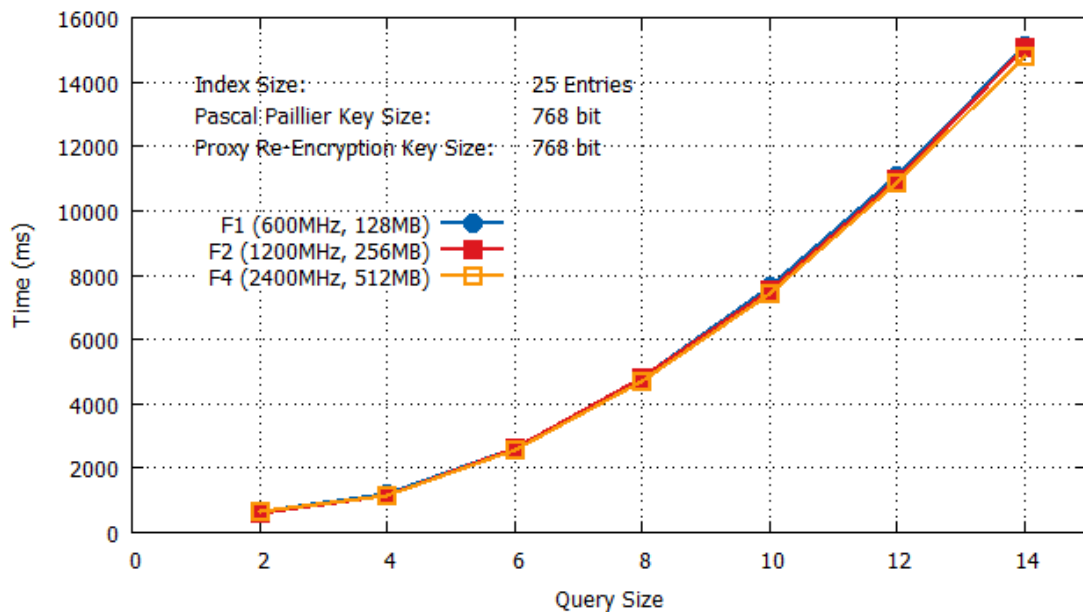


Figure 6.8: CPU Time (ms) of oblivious term matching on F1, F2 and F4 Frontend instances of Google App Engine.

6.9 Enhancing Encrypted Search Data with Post-Processing

Search over encrypted data enables data searching capability within untrusted domain. It realizes keyword matching without the need to decipher the encrypted data. William Harrower et. al., describes number of searching techniques that can be applied to plain text to enhance the search capability of a user [99].

- *word sub-match*: search of substring within a document - a document containing word “cryptography” should be returned when word “crypto” is searched
- *case insensitivity*: search queries that are not case sensitive - search for a word “HASH” should successfully return all the documents containing “Hash”, “hash” or “HASH”
- *regular expressions*: search queries that comply with pattern matching - regular expression can be translated to word
- *proximity based queries*: allow search for a particular word “X” that is close to “Y”
- *natural language search*: search for words that are semantically close to each other

However, the intrinsic nature of cipher-text and higher degree of security requirements (i.e., computational indistinguishability, randomness, permutation) limit these searching capabilities to exact matched queries. [99] discusses number of improvements that can be made to extend the capabilities of exact matched queries to search relevant encrypted documents. As suggested, keywords can be indexed twice to enable case insensitivity search queries i.e., once as it appears in the plain text and second in lowercase. Similarly, compound words can be indexed twice i.e., “dark-room” can be indexed as it appears “darkroom” and also by splitting it “dark” and “room”. These improvements can only be applied to encrypted data search algorithms that perform search over indexed data structures i.e., bloom filters and secure indexes. Nevertheless, the actual searching methodology is based on exact matching.

The proposed methodology of encrypted data search (oblivious data search, OTM) is based on secure inverted index instead of simple indexed data structure. Since, OTM engages third party to post-process the oblivious search results extra functionalities beyond sorting, can be incorporated

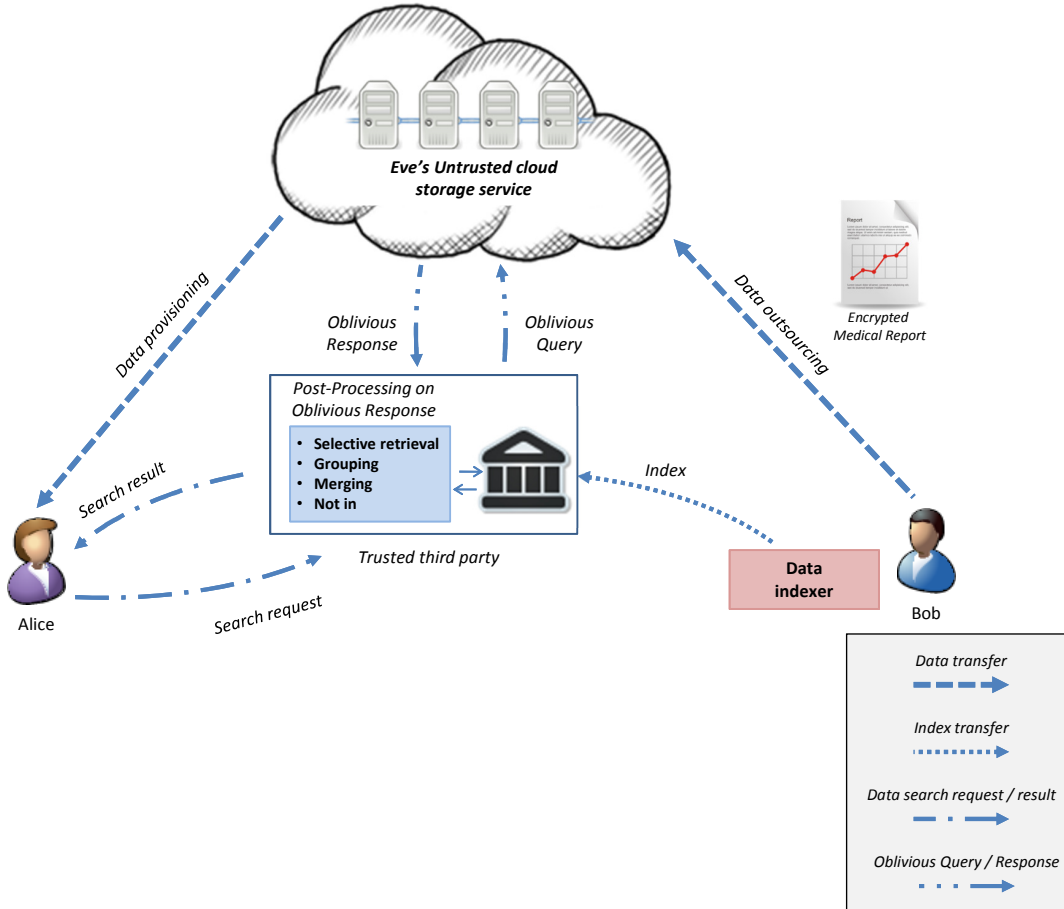


Figure 6.9: Extended functionalities - oblivious term matching conceptual model.

to extend the searching experience. However, the basic searching methodology still be based on exact matching, as it relies on malleability property of homomorphic encryption. Even though, [100, 101] have tried to realize range queries by incorporating order preserving encryption [102]. However, these methodologies can only process numeric data by obfuscating them in range values.

In the following, we delineate the functionalities that can be adopted by post-processing the oblivious query evaluation results at third party². Figure 6.9 illustrates the conceptual model of oblivious term matching with extended functionalities. Authorized subscribers can also post-process the results by their own as OTM merely engages third party to avoid computational intensive tasks at subscribers' end.

²The analogy of proposed functionalities is borrowed from standard SQL. However, these functionalities are proposed within the context of encrypted data search and thus are unlike to standard SQL clauses.

- *selective retrieval*: oblivious query evaluation results can be post-processed to retrieve selective amount of documents which match the encrypted search criteria specified by the authorized subscriber. Third party can be configured to only process specific number or percentage of results. It follows analogy of Top clause of SQL.
- *grouping*: third party can group the oblivious query evaluation results according to presence and absence of encrypted search criteria. Although it cannot learn actual keywords specified by the encrypted criteria; however, as encrypted search criteria models the polynomial as root values it can learn presence and absence of encrypted search criteria. Once, third party deciphers the oblivious response from cloud server it can group the results based on search criteria. Furthermore, groups can be sorted according to keyword frequencies.
- *merging*: if outsourced file is indexed separately or encrypted indexes are distributed among multiple cloud service providers then post-processing of oblivious query evaluation can be utilized to merge results. The analogy of merging is same as SQL union clause; however, it can span on multiple encrypted indexes maintained separately by cloud service providers.
- *not in*: it can be realized by retrieving all those documents that do not contain encrypted search criteria. Since, oblivious search queries are evaluated on entire document space, third party can effortlessly identify documents that do not meet the encrypted search criteria and lesion information to the subscriber.

The aforementioned extended functionalities can be realized by post-processing the oblivious response from cloud server. These extended functionalities conform to the security analysis discussed in Section 6.6. Since, third party is merely post-processing the encrypted keyword frequencies it cannot learn or deduce any confidential information that can be used to compromise privacy of the outsourced data.

6.10 Discussion

Data search is an integral part of cloud storage service. Searching methodologies ensure that subscriber of a cloud storage can access relevant data, without generating needless data access

requests. However, when confidential data is outsourced to these services in encrypted format, subscribers can no longer use standard search queries to look for a particular file or data content. Mainly because, comparison operators cannot be evaluated for the encrypted data and search criteria specified in search query. Besides this, standard search queries do not ensure privacy of the outsourced as malicious or curious cloud service provider can use them to learn confidential information about the outsourced data.

Numerous efforts have been made in the form of cryptographic primitives and enterprise search products to achieve searching capabilities over the encrypted data. These systems mainly utilize cryptographic trapdoors or index data structures to execute search queries. However, these approaches lose their efficacy in the area of cloud storage due to their intrinsic properties of trapdoor distribution, and in-house index management. Moreover, these systems either do not enforce access control policies or rely on trusted third party to achieve privacy-aware data search in an untrusted domain of CSP.

In order to leverage subscribers of cloud storage with searching capabilities we have proposed Oblivious Term Matching (OTM). It is privacy-aware data search that can be regarded as a value added service for existing cloud storages. OTM ensures that search queries are obliviously evaluated by a CSP, without learning any information about the outsourced data. Since, OTM is an indexed based data search, unlike trapdoor-based approaches authorized subscribers are not confined to a limited number of trapdoors defined by a data owner. To restrain CSP from compromising privacy of the outsource data; indexes are encrypted before they can be outsourced to a CSP. OTM is independent of data encryption that ensures confidentiality of the outsource data, thus it can be integrated with any cloud storage system to realize a privacy-aware data search.

OTM utilizes homomorphic and proxy re-encryption to ensure that a cloud server obliviously evaluates search queries and only authorized subscribers can search cloud storage. Since, we utilize private matching, cloud server cannot learn any information about the search criteria (i.e., keywords), as coefficients of a search query are homomorphically encrypted by using Pascal Pailier crypto system. Private matching ensures that cloud server cannot even learn the selection criteria that match with the values in encrypted inverted index. Thus, for query evaluation OTM provides two levels of secrecy. First, cloud server cannot learn the search criteria. Second, it

cannot learn the keywords that are common between the search criteria and encrypted inverted index.

To realize a privacy-aware data search service we utilize proxy re-encryption. It encrypts the inverted index generated by the data owner and search criteria defined by an authorized subscriber. Proxy re-encryption ensures that cloud server only has to persist single copy of encrypted inverted index and yet it is able to evaluate oblivious queries generated by authorized subscribers. For each authorized subscribers cloud server has a valid transformation key. Whenever a subscriber initiates a data search request cloud server transforms the encrypted inverted index by using an appropriate transformation key. Since, only authorized users have their respective transformation key with the cloud server, search queries of an unauthorized user cannot be evaluated successfully. Even if cloud server behaves maliciously and teams up with an unauthorized subscriber, privacy of the outsourced data cannot be compromised this because cloud server does not have valid transformation key and unauthorized user does not have its proxy re-encryption secret key.

OTM utilizes trusted third party to model oblivious queries, process the response of cloud server, and sorting the result of oblivious query evaluation according to the frequencies of matched keywords. Utilizing trusted third party to sort result reveals frequency of individual encrypted keyword. Since, these keywords are concealed with proxy re-encryption no oblivious information is leaked to trusted third party. However, if sorting of results is not required, trusted third party can be seamlessly avoided without losing efficacy of OTM.

The practicality of proposed privacy-aware cloud search is demonstrated by realizing it for Google's cloud ecosystem. We deploy search service on Google App Engine, and encrypted indexed are stored on Google's Datastore. To generate inverted index we opt for Apache Lucene. However, OTM is not confined to any specific indexing framework. Data owner can even manually associate keywords and their respective frequencies with the outsourced data. OTM is mainly based on two asymmetric crypto systems i.e., Pascal Paillier and proxy re-encryption. For our implementation we opt for 1248 bit long keys (by default). As recommended by ECRYPT II 2011, 1248 bit long key based on Discrete Logarithm Group provides long-term protection against small organizations, and very short-term protection against agencies [103], [104]. However, key length of both Pascal Paillier and proxy re-encryption are configurable by the data owner according to

the level resilience required against a determined attacker.

With OTM, we have realized a data search service which ensures that only authorized subscribers can search the outsourced data obliviously. Search queries are obliviously executed by a cloud server that does not learn any information about the outsourced data, not even about the result of query execution. OTM ensures that cloud server can not relate search queries of two different users even if the queries are modelled with similar search criteria. Through our implementation we have highlighted the fact the OTM is independent of underlying cloud storage system. It can seamlessly be integrated with other cloud storage services to leverage subscribers with privacy-aware searching capabilities.

6.11 Summary

This chapter presented Oblivious Term Matching (OTM) - a methodology to search encrypted data outsourced to an untrusted domain. It enables data owner to associate index file with the outsourced data containing arbitrary number of keywords in encrypted form. Search queries are obliviously evaluated for the encrypted index file. OTM is based on private matching, which ensures that query evaluation process appear oblivious to a query evaluator. OTM prevents unauthorized users to deduce confidential information about the data or data owner by simply learning result of query evaluation.

Instead of relying on trapdoors defined for specific keywords, OTM utilizes index-based search. Since, search queries are not bound to trapdoors, authorized users can define their own search queries. OTM greatly increases efficacy of search over encrypted data, as search queries are not limited to predefined trapdoors. Arbitrary number of keywords can be associated with index file, without the need to exchange any information with authorized users. Modeling and post-processing of search queries at trusted third party takes off computational load from individual users. Since, encrypted search criteria is utilized to model oblivious search queries, privacy of outsourced data remains intact, not even trusted third party can deduce confidential information about the outsourced data.

In this chapter, OTM is presented in the context of cloud-based data sharing service. It provides data searching capabilities to authorized users, within the untrusted domain of cloud service

provider. Encrypted data along with concealed index file is outsourced to cloud server. OTM ensures that cloud server obliviously evaluates search queries submitted by authorized users. For unauthorized users, OTM generates randomized response that does not compromise privacy of the outsourced data. OTM can be regarded as a value added service, the enable existing cloud storage services to leverage their subscribers with privacy-aware data searching capabilities, consequently enabling them to access relevant data contents without losing data privacy.

This chapter concludes the research carried out in this dissertation. The subsequent sections summarize the contributions made in this dissertation to the area of cloud-based storage services. In the end, we conclude this study with potential future directions that can be explored to extend the research carried in this dissertation.

7.1 Conclusion

Cloud-based storage services provide a cost effective solution to deal with the problem of on-demand data accessibility. These services enable their subscribers to share, collaborate, archive and synchronize data across different devices and domain, without the concerns of data provisioning and availability. Cloud infrastructure associated with these services is owned, managed and operated by an untrusted entity called cloud service provider. Since, an untrusted is in-charge of processing, persisting and provisioning of outsourced data there is a great deal of privacy concerns when confidential data is outsourced to such services.

To ensure data privacy and confidentiality often cryptographic methodologies are employed (i.e., encryption algorithms, one-way hash functions, key exchange protocols etc.); however, these methodologies are not enough to achieve fine-grained data governance. Apart from this, applying cryptographic methodologies greatly affects the efficacy of a cloud service provider to process outsourced data. This results in underutilized cloud resources - as data owner would have to either setup auxiliary compute resources or rely on trusted third party services to process and provision outsourced data.

7.1.1 Access Control Enforcement in Cloud Storage Services

Access control policies ensure fine-grained access control. However, conventional methodologies were designed to restrain illicit data access in a trusted domain in which only user accessing data could behave maliciously. Contrary to that, cloud storage services were provisioned from public domain by an untrusted entity. Thus, conventional access control policy could be exploited by a cloud service provider to compromise privacy of the outsourced data. Apart from that access control policies and access attributes of an authorized subscribers could reveal confidential information about the outsourced data and data owner as well.

In this dissertation to address the problem of access control enforcement within untrusted domain, we proposed oblivious access control enforcement O-ACE. It enabled data owner to utilize cloud infrastructure for access control policy evaluation without revealing any confidential information about the outsourced data and data owner. The result of access control policy evaluation appeared oblivious to cloud service provider; whereas, only authorized subscriber managed to gain access to the encrypted outsourced data by learning valid data encryption key. O-ACE was implemented in Java and evaluated on Google app-engine. Our performance evaluation results revealed that O-ACE's CPU usage cost was only $0.01 \sim 0.30$ \$ for every 1000 requests for access control policies comprised of two to twelve distinct parameters.

7.1.2 Encrypted Data Search for Cloud Storage Services

Search over encrypted data realizes privacy-aware data searching capabilities within untrusted domain. Encrypted data can be searched either by using trapdoor-based encryption or by persisting indexed data at a secure location. Trapdoor-based searching methodologies relied on untrusted entity to search encrypted data, with privacy considerations and without the need to decipher encrypted data. Indexed based data search provided a richer data searching experience as compared to trapdoor-based data search. However, it utilized extra compute resources or relied on trusted third party services to execute search queries.

To cater the problem of encrypted data search within the untrusted domain, in this dissertation we proposed oblivious term matching OTM. It enabled data owner to realize privacy-aware search methodology without relying on any trusted third party for the execution of search queries. It

engaged cloud service provider to execute search queries, whereas authorized subscribers could define their own search queries without need to get trapdoors from the data owner. We evaluated Java based implementation of OTM on Google app-engine. The evaluation results showed that OTM computation cost for 1000 search queries comprised of two to fourteen keywords was 0.03 \sim 1.09 \$ dollars, thus showed amicable computational load on cloud resources.

7.2 Future Directions

In this dissertation, we contributed to the area of cloud-based storage services. Two methodologies were proposed that utilized oblivious computation within untrusted domain to realize access control policies and to search encrypted data - with privacy considerations. These methodologies ensured that cloud resources were utilized efficiently resulting in maximized utilization of cloud infrastructure with amicable computational load.

- Both O-ACE and OTM utilize oblivious computation (i.e., polynomial evaluation) with homomorphic encryption; they have a tendency to be parallelized for faster computation of encrypted information. Specifically for OTM, map-reduce programming paradigm can be employed to process huge encrypted index scattered within cloud storage service. One of potential direction that can be explored is Hadoop [105] based oblivious data search.
- Another area that can be investigated is oblivious computation with conditional range operators. It enables data owner to define access control policies that can cater range values. Garbled circuits [106] can be exploited to equip O-ACE with diverse number of conditional operators.
- This dissertation was more focused on enabling oblivious computation within untrusted domain devoid of relying on any trusted party / services. Current implementation of oblivious computation do not consider any optimization scheme for polynomial evaluation. For more efficient implementation of O-ACE and OTM, Horner's rules [107] can be explored for efficient polynomial evaluation by transforming monomial form into a computationally efficient form.

Bibliography

- [1] L. Press, “Personal computing: the post-pc era,” *Communications of the ACM*, vol. 42, no. 10, pp. 21–24, 1999.
- [2] T. Chen, “30th anniversary of the pc and the post-pc era [editor’s note],” *Network, IEEE*, vol. 25, no. 5, pp. 2–3, 2011.
- [3] M. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, “Cloud computing: Distributed internet computing for it and scientific research,” *Internet Computing, IEEE*, vol. 13, no. 5, pp. 10–13, 2009.
- [4] T. Velte, A. Velte, and R. Elsenpeter, *Cloud Computing, A Practical Approach*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2010.
- [5] B. T OGRAPH and Y. MORGENS, “Cloud computing,” *Communications of the ACM*, vol. 51, no. 7, 2008.
- [6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.future.2008.12.001>
- [7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Commun. ACM*, vol. 53, pp. 50–58, April 2010. [Online]. Available: <http://doi.acm.org/10.1145/1721654.1721672>

- [8] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud computing: An overview," *Cloud Computing*, pp. 626–631, 2009.
- [9] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, "Cost-benefit analysis of cloud computing versus desktop grids," in *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–12. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1586640.1587662>
- [10] M. D. Assunção, A. Costanzo, and R. Buyya, "A cost-benefit analysis of using cloud computing to extend the capacity of clusters," *Cluster Computing*, vol. 13, pp. 335–347, September 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10586-010-0131-x>
- [11] J. Staten, *Hollow Out The MOOSE: Reducing Cost With Strategic Rightsourcing*, March 2009.
- [12] *Gartner Identifies the Top 10 Strategic Technologies for 2011*, <http://www.gartner.com/it/page.jsp?id=1454221>.
- [13] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," *NIST special publication*, vol. 800, p. 145, 2011.
- [14] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s13174-010-0007-6>
- [15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
- [16] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

- [17] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, nov. 2008, pp. 1–10.
- [18] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in *Grid Computing Environments Workshop, 2008. GCE '08*, nov. 2008, pp. 1–10.
- [19] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*. IEEE, 2009, pp. 44–51.
- [20] Gartner - top trends for 2010, by brian prentice. [Online]. Available: <http://www.gartner.com>
- [21] C. Pettey and L. Goasduff. Gartner says that consumers will store more than a third of their digital content in the cloud by 2016. [Online]. Available: <http://www.gartner.com/it/page.jsp?id=2060215>
- [22] R. Villars and M. Shirer. Demand from public cloud service providers and private cloud adopters will drive strong growth for full range of storage solutions, according to idc. [Online]. Available: <http://www.idc.com/getdoc.jsp?containerId=prUS23097611>
- [23] J. Rebello. Subscriptions to cloud storage services to reach half-billion level this year. [Online]. Available: <http://www.isuppli.com/mobile-and-wireless-communications/news/pages/subscriptions-to-cloud-storage-services-to-reach-half-billion-level-this-year.aspx>
- [24] File sync; online backup - access and file sharing from any device - sugarsync. [Online]. Available: <http://www.sugarsync.com/>
- [25] Online backup, data backup and remote backup solutions — mozy. [Online]. Available: <http://mozy.com/>
- [26] Cloud storage from just cloud. free online storage. [Online]. Available: <http://www.justcloud.com/>

- [27] Zero-knowledge data backup, sync, access, storage and share from any device. [Online]. Available: <https://spideroak.com/>
- [28] Google drive. [Online]. Available: drive.google.com/
- [29] Skydrive live - windows live. [Online]. Available: <https://skydrive.live.com/>
- [30] Dropbox - simplify your life. [Online]. Available: <https://www.dropbox.com/>
- [31] Cloud storage reviews and prices - 2012. [Online]. Available: http://www.nextadvisor.com/cloud_storage/compare.php
- [32] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ser. ICIS '09. New York, NY, USA: ACM, 2009, pp. 1044–1048. [Online]. Available: <http://doi.acm.org/10.1145/1655925.1656114>
- [33] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.
- [34] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.
- [35] W. Jansen and T. Grance, "Guidelines on security and privacy in public cloud computing," *NIST Special Publication*, pp. 800–144, 2011.
- [36] Cloud security alliance: Security guidelines for critical areas of focus in cloud computing. [Online]. Available: <https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf>
- [37] R. Krutz and R. Vines, *Cloud security: A comprehensive guide to secure cloud computing*. Wiley, 2010.
- [38] P. Wooley, "Identifying cloud computing security risks," 2011.

- [39] L. Kaufman, "Data security in the world of cloud computing," *Security & Privacy, IEEE*, vol. 7, no. 4, pp. 61–64, 2009.
- [40] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra, "Executing sql over encrypted data in the database-service-provider model," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. ACM, 2002, pp. 216–227.
- [41] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. O'Reilly Media, Incorporated, 2009.
- [42] W. Stallings and L. Brown, *Computer security*. Prentice-Hall, 2008.
- [43] K. Khan and Q. Malluhi, "Establishing trust in cloud computing," *IT professional*, vol. 12, no. 5, pp. 20–27, 2010.
- [44] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 693–702.
- [45] R. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. Lee, "Trustcloud: A framework for accountability and trust in cloud computing," in *Services (SERVICES), 2011 IEEE World Congress on*. IEEE, 2011, pp. 584–588.
- [46] D. Catteddu and G. Hogben, "Cloud computing risk assessment," *European Network and Information Security Agency (ENISA)*, 2009.
- [47] (2011) World health organization: World health statistics 2011. [Online]. Available: <http://www.idc.com/getdoc.jsp?containerId=prUS23097611>
- [48] (2012) Hipaa compliant data centers. [Online]. Available: <http://www.onlinetech.com/hipaa-compliant-data-centers-full-access>
- [49] (2012) Breaches affecting 500 or more individuals. [Online]. Available: <http://www.hhs.gov/ocr/privacy/hipaa/administrative/breachnotificationrule/breachtool.html>

- [50] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 85–90.
- [51] S. Pearson, "Taking account of privacy when designing cloud computing services," in *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, ser. CLOUD '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 44–52. [Online]. Available: <http://dx.doi.org/10.1109/CLOUD.2009.5071532>
- [52] S. Pearson and A. Charlesworth, "Accountability as a way forward for privacy protection in the cloud," *Cloud Computing*, pp. 131–144, 2009.
- [53] M. Davino, "Assessing privacy risk in outsourcing," *American Health Information Management Association*, vol. 75, pp. 42–46, March 2004.
- [54] N. Leavitt, "Is cloud computing really ready for prime time," *Growth*, vol. 27, no. 5, 2009.
- [55] D. Svantesson and R. Clarke, "Privacy and consumer risks in cloud computing," *Computer Law & Security Review*, vol. 26, no. 4, pp. 391–397, 2010.
- [56] L. Popa, M. Yu, S. Ko, S. Ratnasamy, and I. Stoica, "Cloudpolice: taking access control out of the network," in *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010, p. 7.
- [57] M. Harbach, S. Fahl, M. Brenner, T. Muders, and M. Smith, "Towards privacy-preserving access control with hidden policies, hidden credentials and hidden decisions," in *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*. IEEE, 2012, pp. 17–24.
- [58] J. DeCew, "Privacy," in *The Stanford Encyclopedia of Philosophy*, fall 2012 ed., E. N. Zalta, Ed., 2012.
- [59] A. Levin and P. Abril, "Two notions of privacy online," *Vand. J. Ent. & Tech. L.*, vol. 11, p. 1001, 2008.

- [60] R. Parker, "Definition of privacy, a," *Rutgers L. Rev.*, vol. 27, p. 275, 1973.
- [61] A. Miller, *The assault on privacy: Computers, data banks, and dossiers*. University of Michigan Press, 1971.
- [62] R. Clarke. (2006) Introduction to Dataveillance and Information Privacy, and Definitions of Terms. [Online]. Available: <https://www.anu.edu.au/people/Roger.Clarke/DV/Intro.html>
- [63] P. Samarati and S. de Vimercati, "Access Control: Policies, Models, and Mechanisms," in *Foundations of Security Analysis and Design*, ser. Lecture Notes in Computer Science, R. Focardi and R. Gorrieri, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, Oct. 2001, vol. 2171, ch. 3, pp. 137–196. [Online]. Available: http://dx.doi.org/10.1007/3-540-45608-2_3
- [64] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "Fade: Secure overlay cloud storage with file assured deletion," in *SecureComm*, 2010, pp. 380–397.
- [65] J. Yao, S. Chen, S. Nepal, D. Levy, and J. Zic, "Truststore: Making amazon s3 trustworthy with services composition," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, may 2010, pp. 600–605.
- [66] E. Geron and A. Wool, "Crust: Cryptographic remote untrusted storage without public keys," in *Security in Storage Workshop, 2007. SISW '07. Fourth International IEEE*, sept. 2007, pp. 3–14.
- [67] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 89–98. [Online]. Available: <http://doi.acm.org/10.1145/1180405.1180418>
- [68] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proceedings of the 14th international conference on Financial cryptography and data security*, ser. FC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 136–149. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1894863.1894876>

- [69] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '10. New York, NY, USA: ACM, 2010, pp. 261–270. [Online]. Available: <http://doi.acm.org/10.1145/1755688.1755720>
- [70] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, ser. CCSW '09. New York, NY, USA: ACM, 2009, pp. 55–66. [Online]. Available: <http://doi.acm.org/10.1145/1655008.1655016>
- [71] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2003, pp. 29–42. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1090694.1090698>
- [72] E.-j. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in *Proc. Network and Distributed Systems Security (NDSS) Symposium 2003*, 2003, pp. 131–145. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.6458>
- [73] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on*, 2000, pp. 44–55.
- [74] Y. cheng Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *In Proc. of 3rd Applied Cryptography and Network Security Conference (ACNS, 2005*, pp. 442–455.
- [75] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," 2006.
- [76] Z. Yang, S. Zhong, and R. N. Wright, "Privacy-preserving queries on encrypted data," in *In Proc. of 11th European Symposium On Research In Computer Security (Esorics)*, 2006, pp. 479–495.

- [77] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *EUROCRYPT*, 2004, pp. 506–522.
- [78] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, june 2011, pp. 383–392.
- [79] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, june 2010, pp. 253–262.
- [80] S. Kamara, C. Papamanthou, and T. Roeder, "Cs2: A searchable cryptographic cloud storage system," Microsoft Research, TechReport MSR-TR-2011-58, May 2011.
- [81] Google search appliance. [Online]. Available: <http://www.google.co.uk/enterprise/search/gsa.html>
- [82] Enterprise search server solutions. [Online]. Available: <http://sharepoint.microsoft.com/en-us/product/capabilities/search/Pages/Search-Server.aspx>
- [83] A. Singh, M. Srivatsa, and L. Liu, "Search-as-a-service: Outsourced search over outsourced storage," *ACM Trans. Web*, vol. 3, pp. 13:1–13:33, September 2009.
- [84] P. Paillier, "Trapdooring discrete logarithms on elliptic curves over rings," in *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ser. ASIACRYPT '00. London, UK: Springer-Verlag, 2000, pp. 573–584. [Online]. Available: <http://dl.acm.org/citation.cfm?id=647096.716885>
- [85] —, "Public key cryptosystems based on composite degree residuosity classes," in *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, ser. EUROCRYPT'99. Berlin, Heidelberg: Springer-Verlag, 1999, pp. 223–238. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1756123.1756146>

- [86] M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection." Springer-Verlag, 2004, pp. 1–19.
- [87] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, pp. 1–30, February 2006.
- [88] Sabrina, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of Access Control Evolution on Outsourced Data," in *VLDB*, 2007, pp. 123–134.
- [89] R. Housley, W. Polk, W. Ford, and D. Solo, *Internet X.509 Public Key Infrastructure*, The Internet Engineering Task Force Std., April 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3280.txt>
- [90] O. Goldreich, R. Israel, and T. Dana, "Foundations of cryptography," 1995.
- [91] The legion of the bouncy castle. [Online]. Available: <http://www.bouncycastle.org/>
- [92] S. Cantor, J. Kemp, R. Philpott, and E. Maler, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Std., 03 2005. [Online]. Available: <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf>
- [93] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 89–98. [Online]. Available: <http://doi.acm.org/10.1145/1180405.1180418>
- [94] J. Sun, X. Zhu, and Y. Fang, "A privacy-preserving scheme for online social networks with efficient revocation," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.
- [95] Apache lucene core. [Online]. Available: <http://lucene.apache.org/core/>
- [96] Google app engine - run your web applications on google's infrastructure. [Online]. Available: <http://code.google.com/appengine/>

- [97] Google app engine - using the datastore. [Online]. Available: <http://code.google.com/appengine/docs/java/gettingstarted/usingdatastore.html>
- [98] Google app engine - adjusting application performance. [Online]. Available: <https://developers.google.com/appengine/docs/adminconsole/performance/settings>
- [99] W. Harrower, "Searching encrypted data," *Department of Computing, Imperial College London, Tech. Rep.*, 2009.
- [100] E. Shmueli, R. Waisenberg, Y. Elovici, and E. Gudes, "Designing secure indexes for encrypted databases," *Data and Applications Security XIX*, pp. 925–925, 2005.
- [101] W. Lu, A. Swaminathan, A. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," *SPIE/IS&T Media Forensics and Security*, pp. 7254–18, 2009.
- [102] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 563–574.
- [103] Ecrypt ii yearly report on algorithms and key sizes. [Online]. Available: <http://www.ecrypt.eu.org/documents/D.SPA.17.pdf>
- [104] New european schemes for signatures, integrity, and encryption. [Online]. Available: <https://www.cosic.esat.kuleuven.be/nessie/>
- [105] Apache hadoop. [Online]. Available: <http://hadoop.apache.org>
- [106] A. C.-C. Yao, "How to generate and exchange secrets," in *Foundations of Computer Science, 1986., 27th Annual Symposium on*, oct. 1986, pp. 162–167.
- [107] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2001.

International Journal Papers:

- [1] **Zeeshan Pervez**, Ammar Ahmad Awan, Asad Masood Khattak, Sungyoung Lee, Eui-Nam Huh, “Privacy-aware Searching with Oblivious Term Matching for Cloud Storage”, In Press Journal of Supercomputing (IF 0.534), 2012.
- [2] **Zeeshan Pervez**, Asad Masood Khattak, Sungyoung Lee, Young-Koo Lee, and Eui-Nam Huh, “Oblivious Access Control Policies for Cloud Based Data Sharing Systems”, In Press Computing, Springer (IF: 1.087), 2012.
- [3] **Zeeshan Pervez**, Asad Masood Khattak, Sungyoung Lee and Young-Koo Lee, “SAPDS: Self-Healing Attribute-Based Privacy Aware Data Sharing in Cloud”, Journal of Supercomputing (IF 0.534), pp.11581-11604, ISSN: 0920-8542, January 7, 2011.
- [4] **Zeeshan Pervez**, Asad Masood Khattak, Sungyoung Lee, Young-Koo Lee, “Achieving Dynamic and Distributed Session Management with Chord for Software as a Service Cloud”, Journal of Software (JSW, ISSN 1796-217X), Academy Publisher, 2011.
- [5] Asad Masood Khattak, **Zeeshan Pervez**, Khalid Latif, and Sungyoung Lee, “Time Efficient Reconciliation of Mappings in Dynamic Web Ontologies”, Journal of Knowledge-based Systems, (IF: 1.574), 2012, (In Press).
- [6] Asad Masood Khattak, Phan Tran Ho Truc, Le Xuan Hung, La The Vinh, Viet-hung Dang, Donghai Guan, **Zeeshan Pervez**, Manhyung Han, Sungyoung Lee and Young-koo Lee, “To-

wards Smart Homes Using Low Level Sensory Data”, Journal of Sensors (IF 1.77), ISSN: 1424-8220, 2011.

- [7] Asad Masood Khattak, **Zeeshan Pervez**, Sungyoung Lee and Young-Koo Lee, “Intelligent Healthcare Service Provisioning using Ontology with Low Level Sensory Data”, Transaction on Internet Information Systems (TIIS) (IF: 0.164) ISSN: 1976-7277, Vol.5, No 11, pp. 2016-2034, 2011.

Book Chapter:

- [8] “Key Management Schemes of Wireless Sensor Networks”: A Survey. In Security of Self-Organizing Networks: MANET, WSN, WMN, VANE, Auerbach Publications, CRC Press, Taylor & Francis Group, USA

International Conference Papers:

- [9] **Zeeshan Pervez**, Asad Masood Khattak, Sungyoung Lee and Young-Koo Lee, “CSMC: Chord based Session Management Framework for Software as a Service Cloud”. Published in 5th IEEE International Conference on Ubiquitous Information Management and Communication, ICUIMC 2011, Seoul, Korea.
- [10] **Zeeshan Pervez**, Sungyoung Lee, Sung Jin Hur, Young-Koo Lee, “Multi-Tenant, Secure, Load Disseminated SaaS Architecture”. Published in 12th IEEE Conference on Advance Communication Technology, ICACT 2010, Phoenix Park, Korea.
- [11] **Zeeshan Pervez**, Asad Masood Khattak, Sungyoung Lee and Young-Koo Lee, “Dual Validation Framework for Multi-Tenant SaaS Architecture”. Published in 5th IEEE International Conference on Future Information Technology, FutureTech 2010, Busan, Korea.
- [12] Asad Masood Khattak, **Zeeshan Pervez**, Manhyoung Han, Sungyoung Lee and Chris Nugent, “DDSS: Dynamic Decision Support System for Elderly”, The 25th IEEE International

Symposium on Computer-Based Medical Systems (CBMS 2012), Rome, Italy, June 20-22, 2012

- [13] Asad Masood Khattak, **Zeeshan Pervez**, Wajahat Ali Khan, Sungyoung Lee, and Young-Koo Lee, "A Self Evolutionary Rule-base", The 4th International Conference on u - and e -service, Science and Technology (UNESST'11), Jeju, Korea, December 8 10, 2011.
- [14] Asad Masood Khattak, **Zeeshan Pervez**, Khalid Latif, A. M. Jehad Sarkar, Sungyoung Lee and Young-Koo Lee, "Reconciliation of Ontology Mappings to Support Robust Service Interoperability", The 8th IEEE International Conference on Services Computing (IEEE SCC 2011), Washington DC, July 4-9, 2011.
- [15] Asad Masood Khattak, **Zeeshan Pervez**, Sungyoung Lee and Young-Koo Lee, "Activity Manipulation using Ontological Data for u-Healthcare", The 8th International Conference on Wearable Micro and Nano Technologies for Personalized Health (pHealth 2011), Lyon, France, June 29 - July 1, 2011.
- [16] Asad Masood Khattak, Khalid Latif, **Zeeshan Pervez**, Iram Fatima, Sungyoung Lee and Young-Koo Lee, "Change Tracer: A Protg Plug-in for Ontology Recovery and Visualization", The 13th Asia-Pacific Web Conference (APWeb2011), (LNCS Conference), Beijing, China, April 18-20, 2011.
- [17] Asad Masood Khattak, **Zeeshan Pervez**, Iram Fatima, Sungyoung Lee, Young-Koo Lee, "Towards Efficient Analysis of Activities in Chronic Disease Patients", The 7th International Conference on Ubiquitous Healthcare, Jeju, Korea, October 2010.
- [18] Asad Masood Khattak, **Zeeshan Pervez**, Koo Kyo Ho, Sungyoung Lee, Young-Koo Lee, "Intelligent Manipulation of Human Activities using Cloud Computing for u-Life Care", The 10th Annual International Symposium on Applications and the Internet (SAINT 2010)", Seoul, Korea, July 2010.
- [19] Asad Masood Khattak, **Zeeshan Pervez**, Jehad Sarkar, Young-Koo Lee, "Service Level Semantic Interoperability", International Workshop on Computing Technologies and Business Strategies for u-Healthcare (CBuH 2010), Seoul, Korea, July 2010.

- [20] Asad Masood Khattak, La The Vinh, Dang Viet Hung, Phan Tran Ho Truc, Le Xuan Hung, D. Guan, **Zeeshan Pervez**, Manhyung Han, Sungyoung Lee, Young-Koo Lee, “Context-aware Human Activity Recognition and Decision Making”, “12th International Conference on e-Health Networking, Application Services”, Lyon, France, July, 2010.
- [21] Asad Masood Khattak, **Zeeshan Pervez**, Sung-young Lee, Young-Koo Lee, “After Effects of Ontology Evolution”, The 5th International Conference on Future Information Technology (FutureTech10), Busan Korea, May, 2010.

Domestic Conference Papers:

- [22] **Zeeshan Pervez**, Sung-young Lee, Young-Koo Lee, “Dictionary Based Software Watermarking Technique”, Proceedings of the 32th Korea Information Processing Society Fall Conference announced Volume 16, Issue 2 (November 2009).